

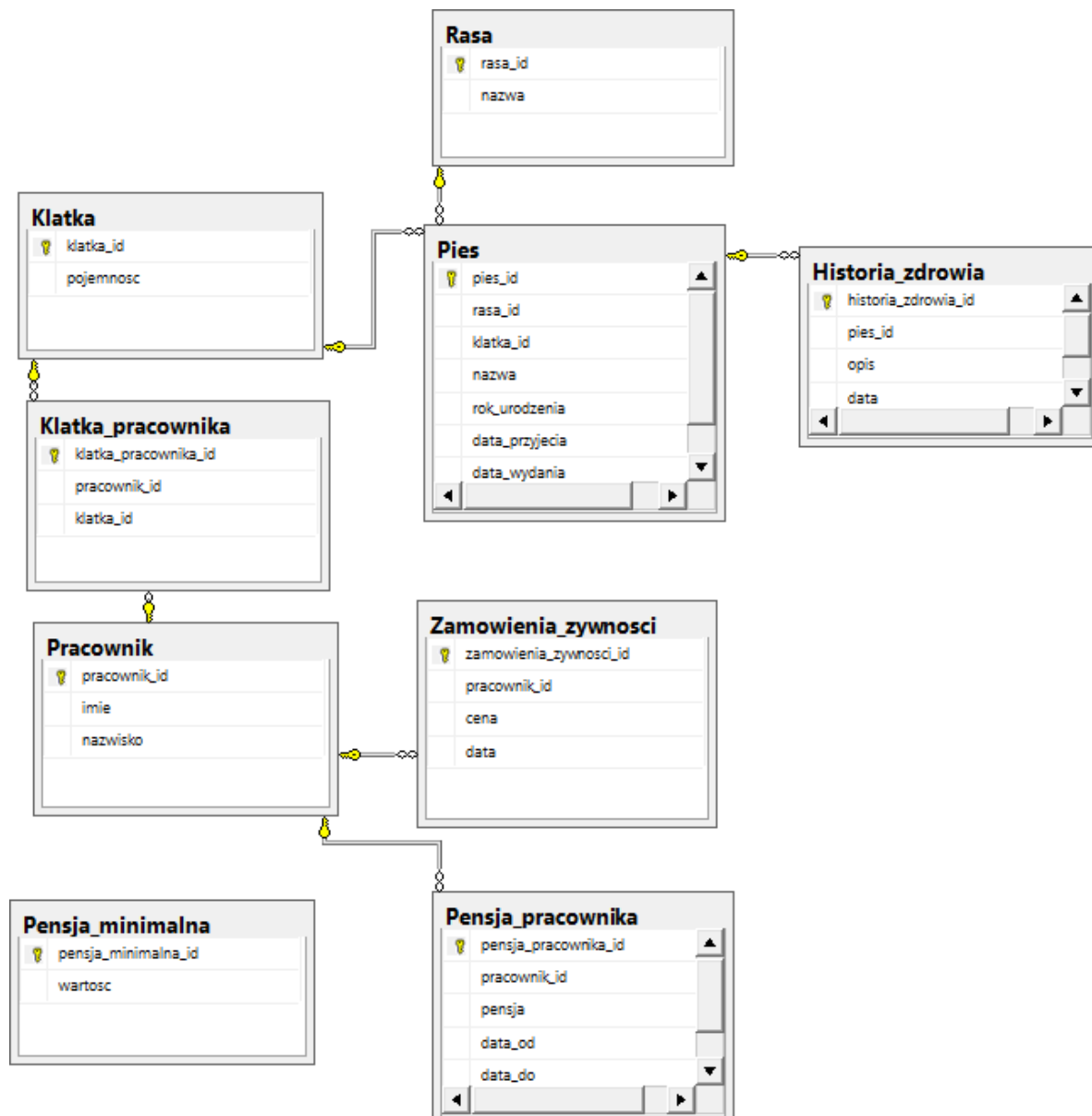
Schronisko dla psów

Zaawansowane Systemy Baz danych

FTIMS 2016/2017

Łukasz Kuta - 195637
Rafał Koper - 195624
Środa godz. 12.00

SCHEMAT



TABELE

```
create table Rasa
(
    rasa_id int identity(1,1) constraint rasa_id_nn not null,
    nazwa varchar(25) constraint rasa_nazwa_nn not null
)

alter table Rasa add constraint rasa_id_pk primary key (rasa_id)

create table Klatka
(
    klatka_id int identity(1,1) constraint klatka_id_nn not null,
    pojemnosc int constraint pojemnosc_nn not null
)

alter table Klatka add constraint klatka_id_pk primary key (klatka_id)

create table Pies
(
    pies_id int identity(1,1) constraint pies_id_nn not null,
    rasa_id int constraint rasa_id_nn not null,
    klatka_id int,
    nazwa varchar(25),
    rok_urodzenia smallint,
    data_przyjecia date constraint data_przyjecia_nn not null,
    data_wydania date
)

alter table Pies add constraint pies_id_pk primary key (pies_id)
alter table Pies add constraint rasa_id_fk foreign key (rasa_id) references
Rasa(rasa_id)
alter table Pies add constraint klatka_id_fk foreign key (klatka_id) references
Klatka(klatka_id)

CREATE TABLE Pracownik
(
    pracownik_id int identity(1,1) constraint pracownik_id_nn NOT NULL,
    imie varchar(25),
    nazwisko varchar(25),
)

ALTER TABLE Pracownik ADD CONSTRAINT pracownik_id_pk PRIMARY KEY (pracownik_id)

CREATE TABLE Pensja_pracownika
(
    pensja_pracownika_id int identity(1,1) constraint pracownik_id_nn NOT NULL,
    pracownik_id int constraint pracownik_id_nn NOT NULL,
    pensja smallmoney constraint pensja_nn NOT NULL,
    data_od date constraint data_od_nn NOT NULL,
    data_do date
)
```

```
ALTER TABLE Pensja_pracownika ADD CONSTRAINT pensja_pracownika_id_pk PRIMARY KEY
(pensja_pracownika_id)
ALTER TABLE Pensja_pracownika ADD CONSTRAINT pracownik_id_fk2 FOREIGN KEY
(pracownik_id) REFERENCES Pracownik(pracownik_id)
```

```
CREATE TABLE Zamowienia_zywnosci
(
    zamowienia_zywnosci_id int identity(1,1) constraint zamowienia_zywnosci_id_nn
NOT NULL,
    pracownik_id int constraint pracownik_id_nn NOT NULL,
    cena float,
    data datetime
)
```

```
ALTER TABLE Zamowienia_zywnosci ADD CONSTRAINT zamowienia_zywnosci_id_pk PRIMARY KEY
(zamowienia_zywnosci_id)
ALTER TABLE Zamowienia_zywnosci ADD CONSTRAINT zamowienia_zywnosci_id_fk FOREIGN KEY
(pracownik_id) REFERENCES Pracownik(pracownik_id)
```

```
Create TABLE Klatka_pracownika
(
    klatka_pracownika_id int identity(1,1) constraint klatka_pracownika_id_nn NOT
NULL,
    pracownik_id int constraint pracownik_id_nn NOT NULL,
    klatka_id int constraint klatka_id_nn NOT NULL
)
```

```
ALTER TABLE Klatka_pracownika ADD CONSTRAINT klatka_pracownika_id_pk PRIMARY KEY
(klatka_pracownika_id)
ALTER TABLE Klatka_pracownika ADD CONSTRAINT pracownik_id_fk FOREIGN KEY
(pracownik_id) REFERENCES Pracownik(pracownik_id)
ALTER TABLE Klatka_pracownika ADD CONSTRAINT klatka_id_fk2 FOREIGN KEY (klatka_id)
REFERENCES Klatka(klatka_id)
```

```
Create TABLE Historia_zdrowia
(
    historia_zdrowia_id int identity(1,1) constraint historia_zdrowia_id_nn NOT
NULL,
    pies_id int constraint pies_id_nn NOT NULL,
    opis varchar(50) constraint opis_nn NOT NULL,
    data date constraint data_nn NOT NULL
)
```

```
ALTER TABLE Historia_zdrowia ADD CONSTRAINT historia_zdrowia_id_pk PRIMARY KEY
(historia_zdrowia_id)
ALTER TABLE Historia_zdrowia ADD CONSTRAINT pies_id_fk FOREIGN KEY (pies_id)
REFERENCES Pies(pies_id)
```

```
Create TABLE Pensja_minimalna
(
    pensja_minimalna_id int constraint pensja_minimalna_id_nn NOT NULL,
    wartosc smallmoney constraint wartosc_nn NOT NULL
)
```

```
ALTER TABLE Pensja_minimalna ADD CONSTRAINT pensja_minimalna_id_pk PRIMARY KEY
(pensja_minimalna_id)
```

KWERENDY

```
--1. Wszystkie psy rasy Akita Inu które zostały wydane
select p.pies_id,p.nazwa,p.data_przyjecia from pies as p
join Rasa as r on r.rasa_id=p.rasa_id
where r.nazwa='Akita Inu' and p.data_wydania is not null
```

	pies_id	nazwa	data_przyjecia
1	8	NULL	2011-01-15
2	16	NULL	2006-01-15

```
--2. Wszystkie psy które były szczepione w ciągu 4 lat od przyjecia
select p.pies_id,p.nazwa,p.data_przyjecia,h.data as data_zabiegu from pies as p
join Historia_zdrowia as h on h.pies_id=p.pies_id
where h.opis='Szczepienie' and (year(h.data)-year(p.data_przyjecia) <4)
```

	pies_id	nazwa	data_przyjecia	data_zabiegu
1	3	NULL	2008-01-15	2010-03-23
2	5	NULL	2008-01-15	2011-02-19
3	6	Biały kiel	2011-01-15	2013-02-11
4	8	NULL	2011-01-15	2011-05-12

```
--3. Różnica między najstarszym a najmłodszym psem będącym jeszcze w schronisku
select MAX(p.rok_urodzenia) - MIN(p.rok_urodzenia) as roznica
from Pies as p
where p.data_wydania is null
```

	roznica
1	11

```
--4. Wolne miejsca w poszczególnych klatkach
select k.klatka_id,(k.pojemnosc - (select count(klatka_id) from pies where
k.klatka_id=klatka_id)) as wolne_miejsca from klatka as k
group by k.klatka_id,k.pojemnosc
```

	klatka_id	wolne_miejsca
1	1	0
2	2	0
3	3	0
4	4	2
5	5	0
6	6	1
7	7	1

--5. Ilosc wolnych miejsc ogółem

```
select (sum(k.pojemnosc) - (select count(p.klatka_id) from pies as p)) as
wolne_miejsca from klatka as k
```

	wolne_miejsca
1	4

--6. Pracownicy ktorych pensja nie zmienila sie od poczatku

```
select concat(p.imie, ' ', p.nazwisko) as Nazwa from Pracownik as p
where ((select count(pracownik_id) from Pensja_pracownika where
pracownik_id=p.pracownik_id) = 1 )
```

	Nazwa
1	Brayan Nowak
2	Adam Adamski
3	Piotr Piotrkowski

--7. suma rocznych wydatków na pensje pracowników

```
select (sum(p.pensja)*12) as roczne_wydatki from Pensja_pracownika as p
where p.data_do is null
```

	roczne_wydatki
1	103080,00

--8. Dane pracowników zajmujących się więcej niż 2 klatkami

```
select p.imie, p.nazwisko
from Pracownik as p
where p.pracownik_id
IN ( select kp.pracownik_id
      from Klatka_pracownika as kp
      group by kp.pracownik_id
      having COUNT(kp.pracownik_id) > 2)
```

	imie	nazwisko
1	Agata	Xsinska
2	Jan	Kowalski
3	Jan	Nowak

--9. Cena i data ostatniego zamowienia zywnosci wraz z danymi pracownika, ktory je wykonal

```
select p.imie, p.nazwisko, z.cena, z.data
from Pracownik as p
right join
(
select top 1 zz.pracownik_id, zz.cena, zz.data
from Zamowienia_zywnosci as zz
order by zz.data desc
) as z on z.pracownik_id = p.pracownik_id
```

	imie	nazwisko	cena	data
1	Agata	Xsinska	561	2016-05-25 00:00:00.000

--10. Wszystkie psy przyjęte w roku, w którym ilość przyjęć była największa

```
select p.pies_id, p.nazwa, j.data_przyjecia
from Pies as p
join
(
select p.pies_id, p.data_przyjecia
from Pies as p
group by p.pies_id, p.data_przyjecia
having YEAR(p.data_przyjecia) = (Select top 1 YEAR(p.data_przyjecia)
From Pies as p
group by YEAR(p.data_przyjecia)
order by COUNT(YEAR(p.data_przyjecia)) desc)
) as j on j.pies_id = p.pies_id
```

	pies_id	nazwa	data_przyjecia
1	6	Biały kiel	2011-01-15
2	7	NULL	2011-01-21
3	8	NULL	2011-01-15
4	9	Brutus	2011-01-15
5	10	NULL	2011-12-15
6	11	Warka	2011-12-15

--11. Ilosc przyjetych psow w poszczegolnych latach

```
select YEAR(p.data_przyjecia) as rok, COUNT(*) as ilosc
from Pies as p
group by YEAR(p.data_przyjecia)
```

	rok	ilosc
1	2002	1
2	2003	1
3	2004	1
4	2006	1
5	2007	1
6	2008	4
7	2009	2
8	2010	2
9	2011	6
10	2015	1
11	2017	1

```
--12. Ilosc wydanych psow w poszczegolnych latach
select YEAR(p.data_wydania) as rok, COUNT(*) as ilosc
from Pies as p
where p.data_wydania is not null
group by YEAR(p.data_wydania)
```

	rok	ilosc
1	2004	1
2	2011	1
3	2012	1
4	2013	1
5	2014	1
6	2015	2
7	2017	1

```
--13. Rasa psa, którą każdy z pracowników ma najwięcej pod opieką
select pr.pracownik_id, max(klatki.nazwa) as nazwa
from Pracownik as pr
join
(
    select kp.pracownik_id, kp.klatka_id, psy.nazwa
    from Klatka_pracownika as kp
    join
    (
        select p.rasa_id, p.klatka_id, r.nazwa
        from Pies as p, Rasa as r
        where r.rasa_id = p.rasa_id
    ) as psy on psy.klatka_id = kp.klatka_id
)
as klatki on klatki.pracownik_id = pr.pracownik_id
group by pr.pracownik_id
```

	pracownik_id	nazwa
1	1	Rottweiler
2	2	Rottweiler
3	3	Owczarek niemiecki
4	4	Terier
5	6	Rottweiler

```
--14. Suma zarobkow kazdego z pracowników
```

```
Select pp.pracownik_id, p.imie, p.nazwisko, SUM(DATEDIFF(month, pp.data_od, CAST(
CASE
WHEN pp.data_do is null
THEN GETDATE()
ELSE pp.data_do
END AS date))) * pp.pensja) as zarobki_ogółem
from Pensja_pracownika as pp, Pracownik as p
where p.pracownik_id = pp.pracownik_id
group by pp.pracownik_id, p.imie, p.nazwisko
```


	pracownik_id	imie	nazwisko	zarobki_ogółem
1	1	Agata	Xsinska	62460,00
2	2	Jan	Kowalski	86280,00
3	3	Brayan	Nowak	81310,00
4	4	Jan	Nowak	114760,00
5	6	Adam	Adamski	0,00
6	7	Piotr	Piotrkowski	0,00

--15 Psy z imieniem zawierającym 'a'

```
Select *
From Pies
where nazwa like '%a%'
```

	pies_id	rasa_id	klatka_id	nazwa	rok_urodzenia	data_przyjecia	data_wydania
1	2	2	1	Maja	2004	2004-03-19	NULL
2	6	4	2	Biały kiel	2011	2011-01-15	NULL
3	11	9	NULL	Warka	2007	2011-12-15	2017-01-10

FUNKCJE

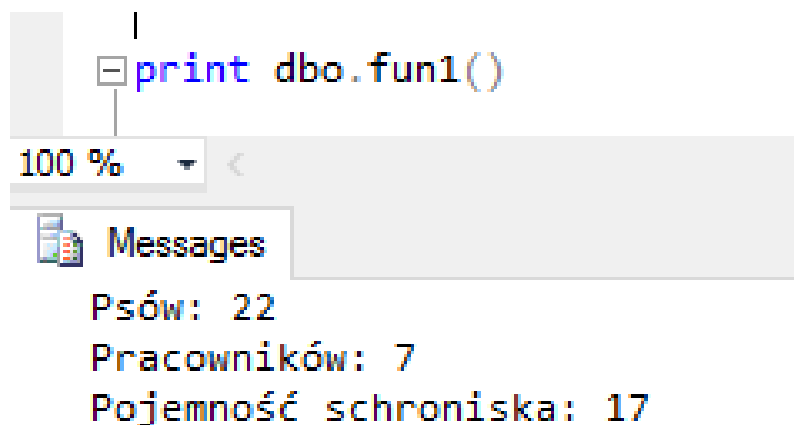
```
IF OBJECT_ID('fun1') IS NOT NULL
DROP FUNCTION fun1
GO
```

--1. Funkcja wyświetlająca podsumowanie dla schroniska

```
CREATE FUNCTION fun1()
RETURNS VARCHAR(100)
AS
BEGIN
    DECLARE @stats varchar(100)
    DECLARE @new_line AS CHAR(2) = CHAR(13) + CHAR(10)

    SET @stats = (SELECT TOP 1 'Psów: ' + CAST(COUNT(p.pies_id) as varchar(10)) +
    @new_line
                FROM schronisko..Pies as p)
    SET @stats += (SELECT TOP 1 'Pracowników: ' + CAST(COUNT(pr.pracownik_id) as
    varchar(10)) + @new_line
                FROM schronisko..Pracownik as pr)
    SET @stats += (SELECT TOP 1 + 'Pojemność schroniska: ' + CAST(SUM(k.pojemnosc)
    as varchar(10))
                FROM schronisko..Klatka as k)

    RETURN @stats;
END
GO
```



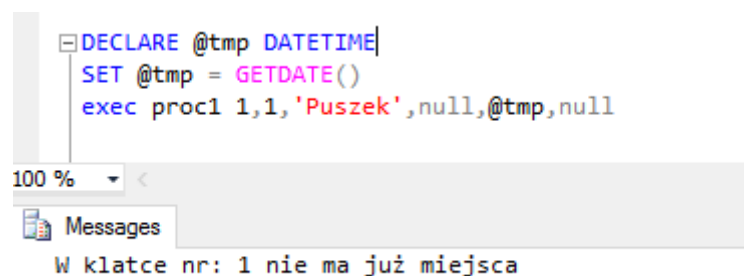
PROCEDURY

--1. Dodawanie nowego psa do schroniska

```
if exists (select 1 from sys.objects where type = 'P' and name = 'proc1')
drop procedure proc1
go

create procedure proc1
@rasa_id int = null,
@klatka_id int = null,
@nazwa varchar(25) = null,
@rok_urodzenia smallint = null,
@data_przyjecia date = null,
@data_wydania date = null
as
begin
    if @rasa_id is null or @data_przyjecia is null
    begin
        print 'Wystąpił błąd. Podano zbyt mało danych!'
    end
    else
    begin
        if @klatka_id in (select k.klatka_id
                        from schronisko..klatka as k
                        group by k.klatka_id,k.pojemnosc
                        having (k.pojemnosc - (select count(klatka_id)
                                           from schronisko..pies
                                           where k.klatka_id=klatka_id)) > 0)
        begin
            insert into schronisko..Pies values
            (@rasa_id, @klatka_id, @nazwa, @rok_urodzenia, @data_przyjecia, @data_wydania)
            print 'Pomyślnie dodano psa do schroniska.'
        end
        else
        begin
            print 'W klatce nr: ' + CAST(@klatka_id AS
            varchar(5)) + ' nie ma już miejsca'
        end
    end
end
go
```

Efekt dodania psa do klatki, która nie ma już miejsc



Efekt dodania psa do klatki z wolnym miejscem

```
DECLARE @tmp DATETIME
SET @tmp = GETDATE()
exec proc1 1,4,'Pluto',2011,@tmp
```

100 %

Messages

(1 row(s) affected)
Pomyślnie dodano psa do schroniska.

Nowy pies został wpisany do bazy i został mu nadany identyfikator o nr 38

	pies_id	rasa_id	klatka_id	nazwa	rok_urodzenia	data_przyjecia	data_wydania
1	1	1	1	NULL	2008	2010-07-15	NULL
2	2	2	1	Maja	2004	2004-03-19	NULL
3	3	3	NULL	NULL	2006	2008-01-15	2004-01-01
4	4	3	1	NULL	2006	2008-01-15	NULL
5	5	3	2	NULL	2006	2008-01-15	NULL
6	6	4	2	Biały kiel	2011	2011-01-15	NULL
7	7	4	2	NULL	2010	2011-01-21	NULL
8	8	5	NULL	NULL	2004	2011-01-15	2015-01-05
9	9	5	3	Brutus	2011	2011-01-15	NULL
10	10	6	3	NULL	2007	2011-12-15	NULL
11	11	9	NULL	Warka	2007	2011-12-15	2017-01-10
12	12	8	5	Puszek	2004	2009-12-15	NULL
13	13	7	6	NULL	2000	2002-01-19	NULL
14	14	7	7	NULL	2001	2003-10-21	NULL
15	15	6	7	NULL	2011	2015-01-05	NULL
16	16	5	NULL	NULL	2004	2006-01-15	2011-06-20
17	17	6	NULL	NULL	2005	2007-02-16	2012-07-21
18	18	7	NULL	NULL	2006	2008-03-17	2013-08-22
19	19	8	NULL	NULL	2007	2009-04-18	2014-09-23
20	20	9	NULL	NULL	2008	2010-05-19	2015-10-24
21	21	1	1	Puszek	NULL	2017-01-03	NULL
22	38	1	4	Pluto	2011	2017-01-10	NULL

```

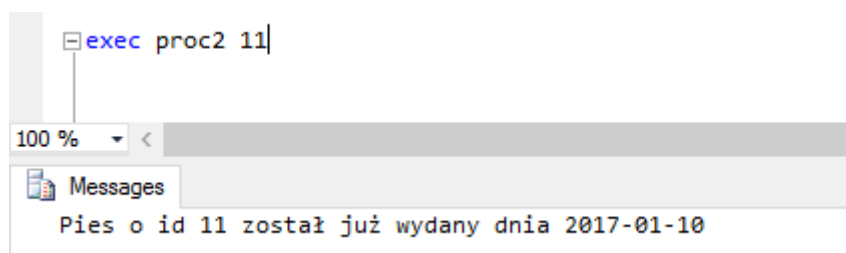
--2. Wydanie psa ze chroniska
if exists (select 1 from sys.objects where type = 'P' and name = 'proc2')
drop procedure proc2
go

create procedure proc2
@id_psa int = null
as
begin
    if @id_psa is null or @id_psa not in (select p.pies_id
                                         from schronisko..Pies as p)
    begin
        print 'Błędne id psa!'
    end
    else
    begin
        DECLARE @data_wydania date
        set @data_wydania = ( select p.data_wydania
                             from schronisko..Pies as p
                             where p.pies_id = @id_psa )
        if @data_wydania is not null
        begin
            print 'Pies o id ' + CAST(@id_psa AS varchar(5)) +
                  ' został już wydany dnia ' + CAST(@data_wydania as
            varchar(10))
        end
        else
        begin
            update schronisko..pies
            set data_wydania=GETDATE()
            where pies_id = @id_psa

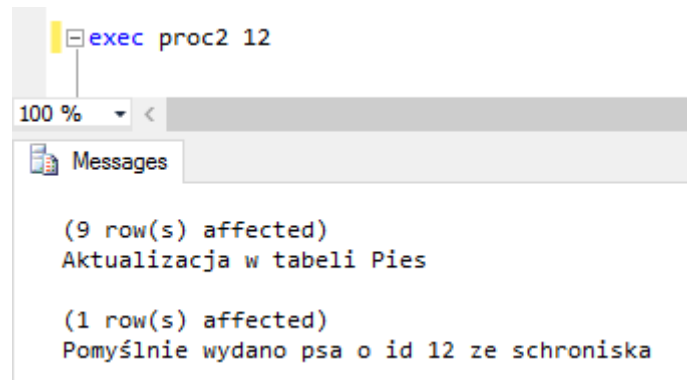
            print 'Pomyślnie wydano psa o id ' + CAST(@id_psa
            as varchar(5)) + ' ze schroniska'
        end
    end
end
go

```

Próba wydania już wydanego psa



Wydanie psa znajdującego się w bazie



Wpis psa o id 12 w tabeli Pies po operacji wydania

	pies_id	rasa_id	klatka_id	nazwa	rok_urodzenia	data_przyjcia	data_wydania
1	12	8	NULL	Puszek	2004	2009-12-15	2017-01-11

--3. Dodanie nowego pracownika wraz z przypisaniem mu klatki, którą opiekuje się najmniej pracowników

```
if exists (select 1 from sys.objects where type = 'P' and name = 'proc3')
drop procedure proc3
go

create procedure proc3
@imie varchar(25) = null,
@nazwisko varchar(25) = null,
@pensja smallmoney = null
as
begin
    if @imie = null or @nazwisko = null
    begin
        print 'Podano błędne imie/nazwisko'
    end
    else
    begin
        insert into schronisko..Pracownik (imie, nazwisko)
        values (@imie, @nazwisko)
        print 'Dodano nowego pracownika: ' + CAST(@imie as varchar(25)) +
        ' ' + CAST (@nazwisko as varchar(25))
        DECLARE @id_pracownika int, @id_klatki int
        SET @id_pracownika = (select top 1 pracownik_id
                             from schronisko..Pracownik
                             order by pracownik_id desc)
        SET @id_klatki = (select top 1 klatka_id
                         from schronisko..Klatka_pracownika
                         group by klatka_id
                         order by COUNT(pracownik_id))
```

```

insert into schronisko..Klatka_pracownika (pracownik_id,
klatka_id) values
(@id_pracownika, @id_klatki)
print 'Dodano klatke o id: ' + CAST(@id_klatki as varchar(5)) + '
dla nowego pracownika'

if @pensja is not null
begin
    update schronisko..Pensja_pracownika
    set pensja = @pensja
    where pracownik_id = @id_pracownika
    print 'Zmieniono pensje dla nowego pracownika.
    Teraz wynosi ona ' + CAST (@pensja as varchar(5))
end
end
go

```

```
exec proc3 'Łukasz', 'Szakuł'
```

100 % <



Messages

(1 row(s) affected)

Dodano domyślna płace dla nowego pracownika

(1 row(s) affected)

Dodano nowego pracownika: Łukasz Szakuł

(1 row(s) affected)

Dodano klatke o id: 5 dla nowego pracownika

WYZWALACZE

--1. zmiana klatki na null kiedy wprowadzimy date wydania psa

```
create trigger trig1
on pies
after update
as
    update pies
    set klatka_id=null
    where data_wydania is not null
    print 'Aktualizacja w tabeli Pies'
go
```

--2. Dodanie nowego pracownika przypisuje mu domyślną pensję minimalną

```
create trigger trig2
on Pracownik
after insert
as
    DECLARE @id_pracownika int, @pensja_minimalna smallmoney
    SET @id_pracownika = (select top 1 pracownik_id from Pracownik order by
pracownik_id desc)
    SET @pensja_minimalna = (select top 1 wartosc from pensja_minimalna)

    insert into schronisko..Pensja_pracownika (pracownik_id, pensja, data_od)
values
    (@id_pracownika, @pensja_minimalna, GETDATE());

    print 'Dodano domyślną płacę dla nowego pracownika'
go
```

Efekty działania powyższych wyzwalaczy możemy zaobserwować w poprzednich przykładach.

Wyzwalacz trig1 przy procedurze nr. 2 (udane wydanie psa ze schroniska).

Wyzwalacz trig2 przy procedurze nr. 3 (udane zatrudnienie nowego pracownika).