Schronisko dla psów

Zaawansowane Systemy Baz danych

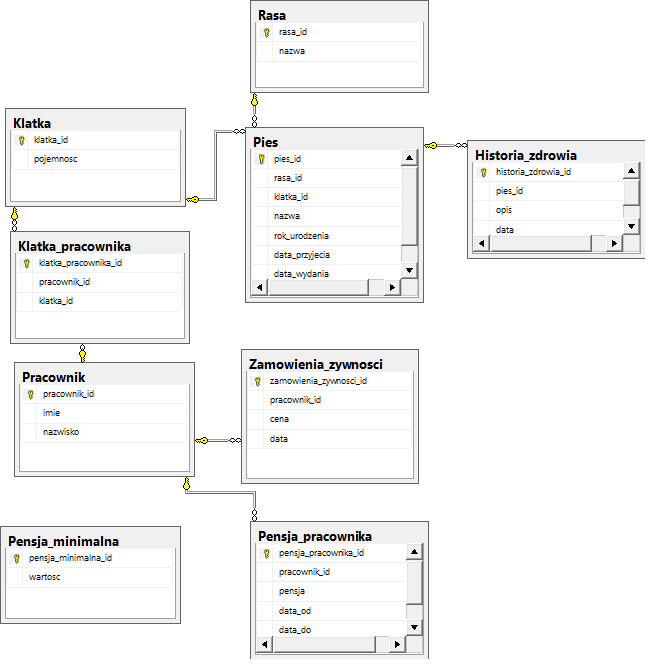
FTIMS 2016/2017

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

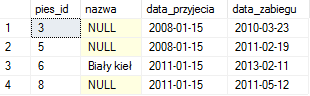


--2. Wszystkie psy które były szczepione w ciągu 4 lat od przyjecia

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)



--3. Różnica miedzy najstarszym a najmlodszym psem będącym jeszcze w schronisku

select MAX(p.rok\_urodzenia) - MIN(p.rok\_urodzenia) as roznica

from Pies as p

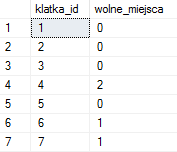
where p.data\_wydania is null



--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



--5. Ilosc wolnych miejsc ogółem

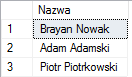
select (sum(k.pojemnosc) - (select count(p.klatka\_id) from pies as p)) as wolne\_miejsca from klatka as k



--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 )



--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



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

select p.imie, p.nazwisko

from Pracownik as p

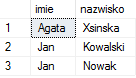
where p.pracownik\_id

IN ( select kp.pracownik\_id

from Klatka\_pracownika as kp

group by kp.pracownik\_id

having COUNT(kp.pracownik\_id) > 2)



--9. Cena i data ostatniego zamownienia 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



--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

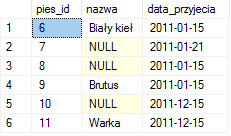
having YEAR(p.data\_przyjecia) = (Select top 1 YEAR(p.data\_przyjecia)

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

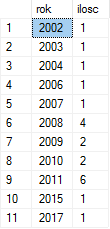


--11. Ilosc przyjetych psow w poszczegolnych latach

select YEAR(p.data\_przyjecia) as rok, COUNT(\*) as ilosc

from Pies as p

group by YEAR(p.data\_przyjecia)



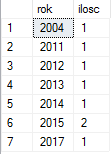
--12. Ilosc wydanych psow w poszczegolnych latach

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)



--13. Rasa psa, którą każdy z pracowników ma najwiecej 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



--14. Suma zarobkow kazdego z pracownikow

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



--15 Psy z imiemiem zawierającym 'a'

Select \*

From Pies

where nazwa like '%a%'



# 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)

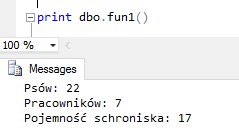
SET @stats += (SELECT TOP 1 + 'Pojemność schroniska: ' + CAST(SUM(k.pojemnosc) as varchar(10))

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'

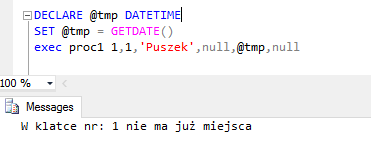
end

end

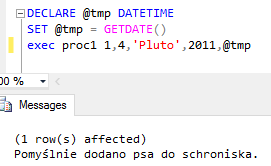
end

go

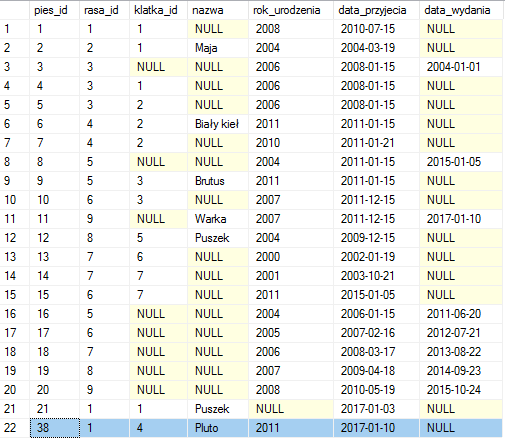
Efekt dodania psa do klatki, która nie ma już miejsc



Efekt dodania psa do klatki z wolnym miejscem



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



--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'

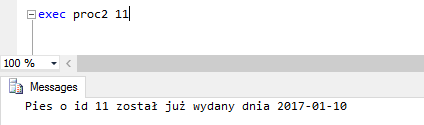
end

end

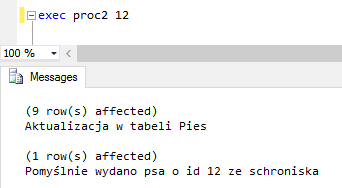
end

go

Próba wydania już wydanego psa



Wydanie psa znajdującego się w bazie



Wpis psa o id 12 w tabeli Pies po operacji wydania



--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

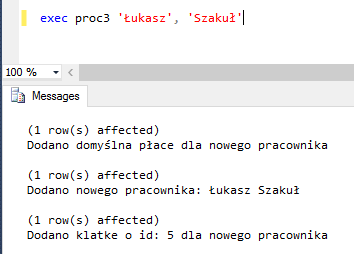
print 'Zmieniono pensje dla nowego pracownika. Teraz wynosi ona ' + CAST (@pensja as varchar(5))

end

end

end

go



# 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 pensje 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 domyślna płace 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).