-----------------------------------------------------------------

----------------------------- zad.1 -----------------------------

-----------------------------------------------------------------

-- Utwórz tabele:

------ Pacjent(id, imie, nazwisko, data urodzenia,

---------------priorytet, ulica, numer, miasto),

------ Wizyta (id, data umowienia, data wizyty,

---------------cena, id lekarza, id pacjenta)

-- Dodaj klucze główne i jeden klucz obcy

-- Dodaj dwa wiersze do tabeli pacjent

-- Dodaj kolumnę płeć w tabeli pacjent

-- Usuń pacjentów mających więcej niż 100 lat

create table pacjent(

id integer not null,

imie varchar(30) not null,

nazwisko varchar(30) not null,

data\_urodzenia datetime not null,

priorytet bit,

ulica varchar(30),

numer varchar(10),

miasto varchar(30)

)

create table wizyta(

id integer not null,

data\_umowienia datetime,

data\_wizyty datetime not null,

cena money,

id\_lekarza integer not null,

id\_pacjenta integer not null

)

alter table pacjent

add constraint pk\_pacjent\_id

primary key (id)

alter table wizyta

add constraint pk\_wizyta\_id

primary key (id)

alter table wizyta

add constraint fk\_wizyta\_pacjent

foreign key (id\_pacjenta) references pacjent (id)

insert into pacjent

values (1,'Jan','Kowalski','2000-01-01',0,'Mala','56a','Warszawa')

insert into pacjent

values (2,'Helena','Trojanska','1900-01-11',0,'Konwaliowa','5','Krakow')

insert into pacjent

values (3,'Katarzyna','Buk','1889-11-09',1,'Starodawna','22','Zakopane'),

(4,'Jan','Brzechwa','1991-12-19',1,'Tuwimowa','2c','Gdynia'),

(5,'Ksawery','Zaspa','1980-07-04',0,'Pulawska','57g','Warszawa')

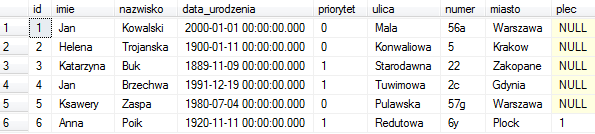
alter table pacjent

add plec bit

insert into pacjent

values (6,'Anna','Poik','1920-11-11',1,'Redutowa','6y','Plock',1)

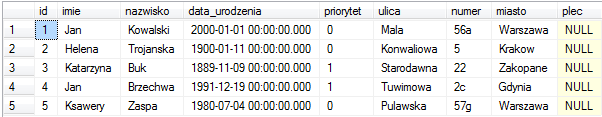
select \* from pacjent



delete from pacjent

where id=6

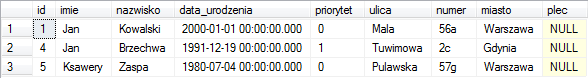
select \* from pacjent



delete from pacjent

where datediff(yy,data\_urodzenia, getdate())>100

select \* from pacjent



-----------------------------------------------------------------

----------------------------- zad.2 -----------------------------

-----------------------------------------------------------------

-- Zaproponuj indeksy dla nowo utworzonych tabel wraz z ich wlasciwosciami

-- (klastrowy/nieklastrowy, unikalny itp.). Napisz kod tworzacy dwa wybrane

-- indeksy dwoch roznych typow.

-- Kiedy dajemy indeksy?

---- zbior ma duzo wierszy

---- chcemy potem po tej kolumnie wyszukiwac

---- duza roznorodnosc wartosci

-- Przymykamy oko na to, ze nasz zbior danych nie jest duzy...

-- pacjent:

---- id -> klastrowy, unikalny

---- imie, nazwisko -> nieklastrowy, nieunikalny

---- data\_urodzenia -> nieklastrowy, nieunikalny

-- wizyta:

---- id -> klastrowy, unikalny

---- data\_wizyty -> nieklastrowy, nieunikalny

---- id\_lekarza -> nieklastrowy, nieunikalny

---- id\_pacjenta -> nieklastrowy, nieunikalny

-- na kluczu glownym tworza sie automatycznie, wiec tam nam sql nie pozwoli juz dodac

create unique clustered index id on pacjent (id)

create index data\_ur on pacjent (data\_urodzenia)

-----------------------------------------------------------------

----------------------------- zad.3 -----------------------------

-----------------------------------------------------------------

-- Northwind:

-- Dodaj kolumne bonus typu money w tabeli employees.

-- Napisz procedure skladowa, ktora dla kazdego pracownika wyznaczy wartosc premii

-- w zaleznosci od liczby zamowien, ktore nadzorowal:

---- 0-50 zamowienia -> 100

---- 51-100 zamowienia -> 150

---- wiecej niz 100 zamowienia -> 250

alter table employees

add bonus int

create procedure bonusy

as

begin

declare @pracow int -- to beda zmienne

declare @liczbazam int

declare pracownik cursor local for select employeeid from employees -- a to iterator

open pracownik

fetch next from pracownik into @pracow -- zmienna pracownik staje sie kolejnym (pierwszym)

-- pracownikiem

while @@fetch\_status=0

begin

set @liczbazam=(select count(\*) from orders where employeeid=@pracow)

if @liczbazam <= 50 and @liczbazam >= 0

begin

update employees

set bonus = 100

where employeeid=@pracow

end

else if @liczbazam <= 100

begin

update employees

set bonus = 150

where employeeid=@pracow

end

else

begin

update employees

set bonus = 250

where employeeid=@pracow

end

fetch next from pracownik into @pracow

end

close pracownik

deallocate pracownik

end

execute bonusy

select employeeid, lastname, firstname, bonus

from employees

