

# Informatyka, studia dzienne, I st. Semestr IV

## Zaawansowane systemy baz danych 2017/2018

---

Prowadzący: dr hab. inż. Przemysław Ignaciuk Środa 10:15

*Autorzy:*

*Łukasz Darul*

*Jędrzej Skoczylas*

### Sprawozdanie - Baza danych, zadanie projektowe - Szpital

---

#### **Spis treści:**

- 1.Opis projektu.
- 2.Kwerendy.
- 3.Procedury.
- 4.Funkcja.
- 5.Wyzwalacze.

# 1. Opis projektu

---

Zaimplementowana przez nas baza danych stanowi symulację bazy danych szpitala przy następujących założeniach:

- 1.Szpital ma wiele pacjentów, przy czym nie każdy pacjent ma zakładaną kartę pacjenta, wynika to z chęci urealnienia bazy tzn. nie każda osoba, która przyjdzie do szpitala otrzyma pomoc, wynika to z tego, że nie każdy posiada ubezpieczenie będące konieczne do realizacji świadczeń medycznych.
- 2.W karcie pacjenta figurują wszystkie pobyty pacjenta.
- 3.W czasie pobytu pacjent może odbywać wiele różnych wizyt.
- 4.Wizyta może skutkować skierowaniem na badania.
- 5.Pracownik może posiadać specjalizacje.
- 6.Zwolnienia i zatrudnienia pracowników są zapisywane w zdarzeniach.

W celach zwiększenia funkcjonalności bazy niektóre operacje zostały zautomatyzowane przy użyciu wyzwalaczy jak np. w momencie aktualizacji płacy pracownika dodaje się wpis do dziennika zdarzeń.Dodatkowo zaimplementowano metodę on delete cascade, która umożliwia usunięcie wszelkich śladów po obecności obiektu w każdej z tabel posiadających korelacje ze sobą.

Skrót algorytmu działania:

Na początku pacjent rejestrowany jest w tabeli pacjent będącej zbiorem osób, które przyszły do szpitala.Następnie część pacjentów została zarejestrowana(w.w. powód dotyczący ubezpieczenia) lub nie bo możliwa jest sytuacja, że owy pacjent istnieje już w bazie pacjentów.Po utworzeniu karty pacjent otrzymuje wpis dotyczący pobytu w szpitalu. W trakcie pobytu pacjent odbywa jedną lub kilka wizyt rozpoznawczych podczas, których lekarz prowadzący decyduje o tym czy istnieje potrzeba skierowania pacjenta na badania specjalistyczne.Wizyty rozpoznawcze mogą być odbywane przez różnych lekarzy z różnymi specjalnościami.Tabela zdarzenia służy nam do rejestrowania zmian w pensjach pracowników oraz zmian w kadrze.

Nowa karta

### karta\_pacjenta

id\_karty  
id\_pacjenta  
data\_rejestracji

### pobyt

id\_pobytu  
id\_karty  
pocz\_pobytu  
koniec\_pobytu

### pacjent

id\_pacjenta  
imie  
nazwisko  
data\_ur

### zdarzenia

id\_zdarzenia  
opis\_zdarzenia

### specjalnosci

id\_specjalnosci  
nazwa\_specjalnosci

### wizyta

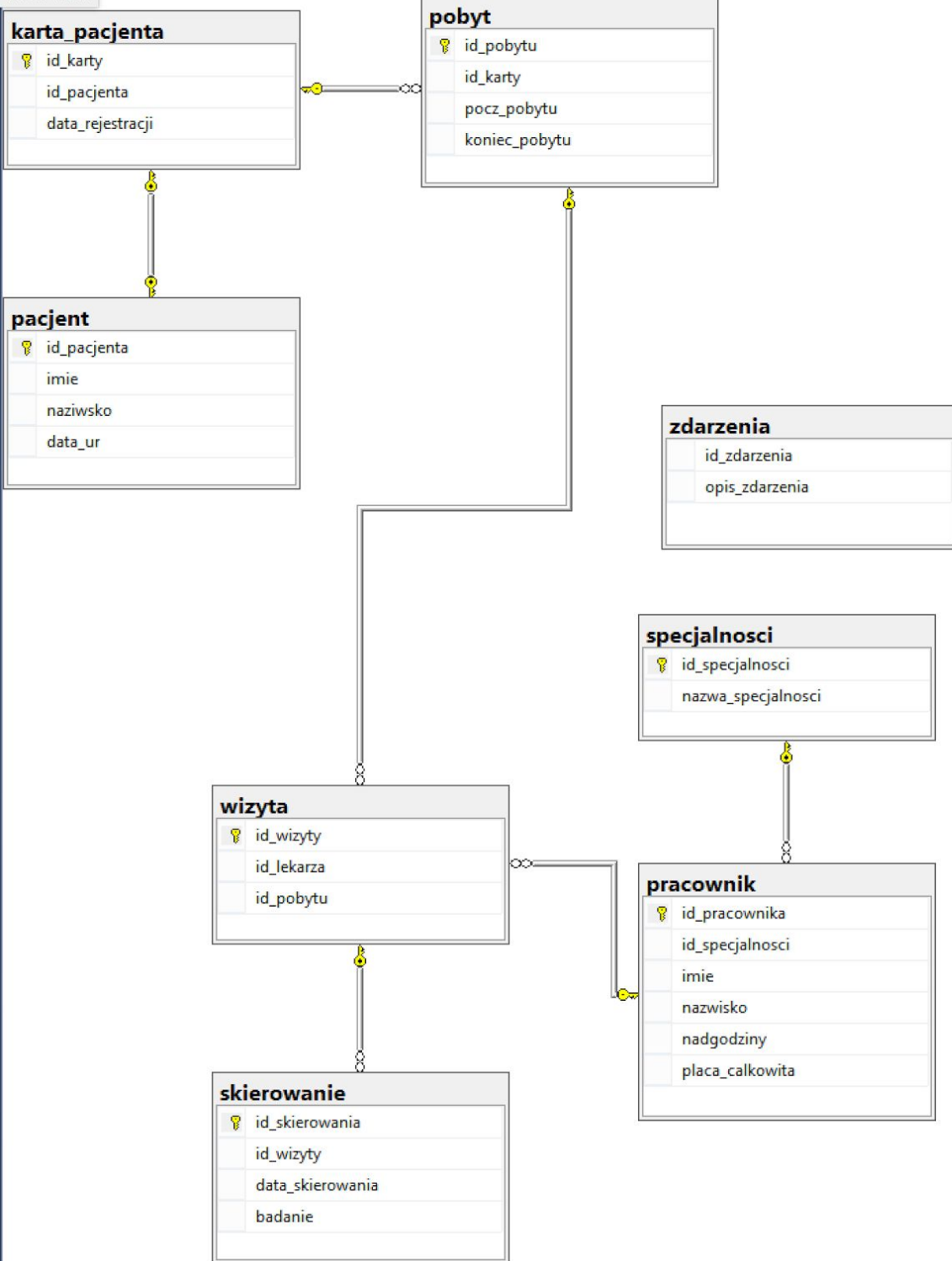
id\_wizyty  
id\_lekarza  
id\_pobytu

### pracownik

id\_pracownika  
id\_specjalnosci  
imie  
nazwisko  
nadgodziny  
placa\_calkowita

### skierowanie

id\_skierowania  
id\_wizyty  
data\_skierowania  
badanie



## 2.Kwerendy

---

### 1.Pacjenci o nazwisku zaczynającym się na D

```
select imie,nazwisko from pacjent where nazwisko like 'D%'
```

	imie	nazwisko
1	Waldemar	Drekst
2	Witold	Drekst

### 2.Pacjenci urodzeni po roku 1994

```
select nazwisko from pacjent where datepart(yy,data_ur)<1994
```

	nazwisko
1	Drekst
2	Romaniuk
3	Rajewska
4	Piotrowska
5	Kowalski
6	Drekst
7	Romaniuk
8	Ziemowicz
9	Tuper

### 3.Podaj najmłodszego pacjenta

```
select top 1 imie,nazwisko,data_ur from pacjent  
order by data_ur desc
```

	imie	nazwisko	data_ur
1	Agnieszka	Wellzer	2003-12-30 00:00:00.000

### 4.Podaj pracownika zarabiającego mniej niż sredni zarobek w szpitalu

```
select id_pracownika,imie,nazwisko,placa_calkowita from pracownik  
where placa_calkowita<(select avg(placa_calkowita) from pracownik)
```

	id_pracownika	imie	nazwisko	placa_calkowita
1	1	Andrzej	Lavoide	4500
2	3	Adrian	Kreatywny	2200
3	4	Krzysztof	Staff	6000
4	5	Mirosław	Stary	3500

**5.Podaj id pacjentów, którzy nie otrzymali skierowania na badania po odbyciu wizyty**

```
select distinct pacjent.id_pacjenta from pacjent
left join karta_pacjenta on karta_pacjenta.id_pacjenta = pacjent.id_pacjenta
left join pobyt on pobyt.id_karty=karta_pacjenta.id_karty
left join wizyta on wizyta.id_pobytu=pobyt.id_pobytu
left join skierowanie on skierowanie.id_wizyty=wizyta.id_wizyty
where skierowanie.id_skierowania is null
```

	id_pacjenta
1	2
2	4
3	5
4	7
5	8
6	10
7	11

**6.Podaj pracowników ktorzy odbyli przynajmniej jedna wizyte**

```
select imie,nazwisko from pracownik
left join wizyta on wizyta.id_lekarza=pracownik.id_pracownika
where wizyta.id_wizyty is not null
```

	imie	nazwisko
1	Andrzej	Lavoide
2	Mirosław	Stary
3	Adrian	Kreatywny
4	Krzysztof	Staff
5	Mirosław	Stary
6	Jedrzej	Dropicki
7	Lukasz	Walewski
8	Marcin	Lavoide
9	Lukasz	Walewski
10	Andrzej	Lavoide
11	Krzysztof	Staff
12	Jedrzej	Dropicki

### 7.Podaj lekarzy o tej samej specjalności

```
select p1.imie,p1.nazwisko ,p1.id_specjalnosci from pracownik p1,pracownik p2
where p1.id_specjalnosci=p2.id_specjalnosci and p1.id_pracownika!=p2.id_pracownika
```

	imie	nazwisko	id_specjalnosci
1	Michał	Dandrewicz	2
2	Marcin	Lavoide	2

### 8.Podaj procentowy zarobek specjalności względem całego szpitala

```
select specjalnosci.nazwa_specjalnosci,
cast(round((cast(sum(placa_calkowita) as float)/cast((select sum(placa_calkowita) from
pracownik) as float))*100,2) as varchar)+'%' as udzial
from pracownik
left join specjalnosci on specjalnosci.id_specjalnosci=pracownik.id_specjalnosci
group by specjalnosci.nazwa_specjalnosci
```

	nazwa_specjalnosci	udzial
1	Ginekolog	7.11%
2	Kardiolog	18.29%
3	Laryngolog	9.15%
4	Okulista	13.62%
5	Ortopeda	4.47%
6	Pediatra	35.16%
7	Pulmonolog	12.2%

### 9.Zawody gdzie zarobek wynosi +5000zł

```
select specjalnosci.nazwa_specjalnosci,avg(placa_calkowita) as srednia from pracownik
left join specjalnosci on specjalnosci.id_specjalnosci=pracownik.id_specjalnosci
group by specjalnosci.nazwa_specjalnosci
having avg(placa_calkowita)>5000
```

	nazwa_specjalnosci	srednia
1	Kardiolog	9000
2	Okulista	6700
3	Pediatra	8650
4	Pulmonolog	6000

### 10. Wypisz 3 najbardziej zapracowanych lekarzy

```
select top 3 id_pracownika, imie, nazwisko, count(wizyta.id_wizyty) as ilosc_wizyt from  
pracownik
```

```
left join wizyta on wizyta.id_lekarza = pracownik.id_pracownika
```

```
group by imie, nazwisko, id_pracownika
```

```
order by count(wizyta.id_wizyty) desc
```

	id_pracownika	imie	nazwisko	ilosc_wizyt
1	1	Andrzej	Lavoide	4
2	7	Lukasz	Walewski	4
3	4	Krzysztof	Staff	3

### 11. Ile skierowan otrzymał każdy pacjent?

```
select p.imie, p.nazwisko, p.id_pacjenta, count(skierowanie.id_skierowania) as wizyty from  
pacjent p
```

```
left join karta_pacjenta on karta_pacjenta.id_pacjenta = p.id_pacjenta
```

```
left join pobyt on pobyt.id_karty = karta_pacjenta.id_karty
```

```
left join wizyta on wizyta.id_pobytu = pobyt.id_pobytu
```

```
left join skierowanie on skierowanie.id_wizyty = wizyta.id_wizyty
```

```
group by p.imie, p.nazwisko, p.id_pacjenta
```

	imie	nazwisko	id_pacjenta	skierowania
1	Maciej	Kowalski	1	2
2	Waldemar	Drekst	2	3
3	Piotr	Romaniuk	3	2
4	Katarzyna	Rajewska	4	2
5	Agnieszka	Wellzer	5	0
6	Adrianna	Piotrowska	6	1
7	Adam	Kowalski	7	1
8	Witold	Drekst	8	1
9	Piotr	Romaniuk	9	3
10	Maria	Ziemowicz	10	0
11	Zofia	Tuper	11	0

### 12. Podaj zawody nie wykonywane w szpitalu

```
select nazwa_specjalnosci from specjalnosci
```

```
where not exists(select id_specjalnosci from pracownik where  
specjalnosci.id_specjalnosci = pracownik.id_specjalnosci)
```

	nazwa_specjalnosci
1	Fizjolog
2	Reumatolog

### 13.Podaj najbardziej chorowitego pacjenta

`select top 1 pacjent.imie, pacjent.nazwisko, count(pobyt.id_pobytu) as ilosc_wizyt from`  
`pacjent`

`left join karta_pacjenta on karta_pacjenta.id_pacjenta=pacjent.id_pacjenta`  
`left join pobyt on pobyt.id_karty=karta_pacjenta.id_karty`  
`group by pacjent.imie, pacjent.nazwisko`  
`order by count(pobyt.id_pobytu) desc`

	imie	nazwisko	ilosc_wizyt
1	Katarzyna	Rajewska	4

### 14.Jaki procent lekarzy stanowią pediatrzy

`select cast(count(case when nazwa_specjalnosci='Pediatra' then 1 else null end)as`  
`float)/cast(count(*)as float)*100 as procent from specjalnosci`  
`left join pracownik on pracownik.id_specjalnosci=specjalnosci.id_specjalnosci`

	procent
1	20

### 15.Podaj ile lat miał każdy pacjent w momencie pierwszej wizyty

`select distinct p.imie, p.nazwisko,`  
`datediff(year, p.data_ur,`  
`(`  
`select top 1 pocz_pobytu from pobyt`  
`left join karta_pacjenta on karta_pacjenta.id_karty=pobyt.id_karty`  
`where p.id_pacjenta=karta_pacjenta.id_pacjenta`  
`)) from pacjent p where exists(select id_pacjenta from karta_pacjenta where`  
`id_pacjenta=p.id_pacjenta)`

	imie	nazwisko	(No column name)
1	Adam	Kowalski	38
2	Adrianna	Piotrowska	3
3	Agnieszka	Wellzer	5
4	Katarzyna	Rajewska	48
5	Maciej	Kowalski	7
6	Piotr	Romaniuk	21
7	Piotr	Romaniuk	88
8	Waldemar	Drekst	35
9	Witold	Drekst	39



# 3.Procedury

## 3.1 Tworzy pacjenta i generuje id inne niż zawarte w bazie

```
go
create procedure TWORZ_PACJENT (@imie varchar(10),@nazwisko varchar(10),@data_ur datetime
as
begin
    declare @identyfikator int
    set @identyfikator = (select top 1 id_pacjenta from pacjent order by id_pacjenta desc) + 1

    insert into pacjent values(@identyfikator,@imie,@nazwisko,@data_ur)
end
```

PRZED

	id_pacjenta	imie	nazwisko	data_ur
1	1	Maciej	Kowalski	1997-06-23 00:00:00.000
2	2	Waldemar	Drekst	1970-01-01 00:00:00.000
3	3	Piotr	Romaniuk	1924-08-12 00:00:00.000
4	4	Katarzyna	Rajewska	1965-11-23 00:00:00.000
5	5	Agnieszka	Wellzer	2003-12-30 00:00:00.000
6	6	Adrianna	Piotrowska	1988-05-06 00:00:00.000
7	7	Adam	Kowalski	1970-01-12 00:00:00.000
8	8	Witold	Drekst	1972-05-09 00:00:00.000
9	9	Piotr	Romaniuk	1974-01-22 00:00:00.000
10	10	Maria	Ziemowicz	1986-03-15 00:00:00.000
11	11	Zofia	Tuper	1989-01-01 00:00:00.000

PO

	id_pacjenta	imie	nazwisko	data_ur
1	1	Maciej	Kowalski	1997-06-23 00:00:00.000
2	2	Waldemar	Drekst	1970-01-01 00:00:00.000
3	3	Piotr	Romaniuk	1924-08-12 00:00:00.000
4	4	Katarzyna	Rajewska	1965-11-23 00:00:00.000
5	5	Agnieszka	Wellzer	2003-12-30 00:00:00.000
6	6	Adrianna	Piotrowska	1988-05-06 00:00:00.000
7	7	Adam	Kowalski	1970-01-12 00:00:00.000
8	8	Witold	Drekst	1972-05-09 00:00:00.000
9	9	Piotr	Romaniuk	1974-01-22 00:00:00.000
10	10	Maria	Ziemowicz	1986-03-15 00:00:00.000
11	11	Zofia	Tuper	1989-01-01 00:00:00.000
12	12	Tester	Tester	2018-02-16 02:04:20.697

### 3.2. Podwyższa przychód specjalnościom o konkretny procent

```
go
create procedure AKTUALIZUJ_PRACOWNIK @id_pracownika int,@procent float
as
    begin
        declare @payment int,@npayment float
        set @payment = (select placa_calkowita from pracownik where
id_pracownika=@id_pracownika)
        set @npayment =(select round(((@payment + @payment*(@procent/100)),2))

        update pracownik
        set placa_calkowita = @npayment
        where id_pracownika = @id_pracownika
    end
```

PRZED

	id_pracownika	id_specjalnosci	imie	nazwisko	placa_calkowita
1	1	1	Andrzej	Lavoide	4500
2	2	2	Michal	Dandrewicz	7300
3	3	3	Adrian	Kreatywny	2200
4	4	4	Krzysztof	Staff	6000
5	5	5	Mirosław	Stary	3500
6	6	6	Jedrzej	Dropicki	6700
7	7	7	Lukasz	Walewski	9000
8	8	2	Marcin	Lavoide	10000

PO(Zmiana dla 1 pracownika o 20% )

$4500 \cdot 1.2=5400$

	id_pracownika	id_specjalnosci	imie	nazwisko	placa_calkowita
1	1	1	Andrzej	Lavoide	5400
2	2	2	Michal	Dandrewicz	7300
3	3	3	Adrian	Kreatywny	2200
4	4	4	Krzysztof	Staff	6000
5	5	5	Mirosław	Stary	3500
6	6	6	Jedrzej	Dropicki	6700
7	7	7	Lukasz	Walewski	9000
8	8	2	Marcin	Lavoide	10000

### 3.3 Usuwanie rejestracji utworzonych pomiędzy datami oraz wszelkich danych

Na potrzeby tej funkcji istotne było ustawienie w procesie tworzenia tabel opcji on cascade delete celem “dynamicznego” usunięcia wszystkich danych związanych z kartą pacjenta.

```
go
create procedure USUN_KARTE @start smalldatetime,@koniec smalldatetime
as
begin
    delete from karta_pacjenta where data_rejestracji between @start and @koniec
end
```

PRZED (Dla uproszczenia dodano tylko tabele karta\_pacjenta i wizyty będące tabelami skrajnymi na trasie pobytu pacjenta w szpitalu)

	id_karty	id_pacjenta	data_rejestracji
1	K001	1	2003-12-04 00:00:00
2	K002	2	2001-11-08 00:00:00
3	K003	3	2011-10-12 00:00:00
4	K004	4	2013-09-16 00:00:00
5	K005	5	2004-08-20 00:00:00
6	K006	6	1988-07-21 00:00:00
7	K007	7	2006-06-23 00:00:00
8	K008	8	2010-05-25 00:00:00
9	K009	9	1990-04-27 00:00:00

Tabela z kartami pacjentów

	id_wizyty	id_lekarza	id_pobytu
1	W01	1	1
2	W02	5	2
3	W03	3	3
4	W04	4	3
5	W05	5	4
6	W06	6	5
7	W07	7	5
8	W08	8	6
9	W09	7	7
10	W10	1	8
11	W11	4	8
12	W12	6	9
13	W13	8	10
14	W14	7	11
15	W15	1	11
16	W16	3	12
17	W17	6	13
18	W18	8	14
19	W19	4	15
20	W20	5	16
21	W21	7	16
22	W22	2	17
23	W23	1	18

Tabela z wizytami pacjentów

PO

	id_karty	id_pacjenta	data_rejestracji
1	K002	2	2001-11-08 00:00:00
2	K003	3	2011-10-12 00:00:00
3	K004	4	2013-09-16 00:00:00
4	K006	6	1988-07-21 00:00:00
5	K007	7	2006-06-23 00:00:00
6	K008	8	2010-05-25 00:00:00
7	K009	9	1990-04-27 00:00:00

	id_wizyty	id_lekarza	id_pobytu
1	W03	3	3
2	W04	4	3
3	W05	5	4
4	W06	6	5
5	W07	7	5
6	W08	8	6
7	W09	7	7
8	W10	1	8
9	W11	4	8
10	W12	6	9
11	W13	8	10
12	W14	7	11
13	W15	1	11
14	W17	6	13
15	W18	8	14
16	W19	4	15
17	W20	5	16
18	W21	7	16
19	W22	2	17
20	W23	1	18

# 4.Funkcja

## 4.1.Funkcja średni zarobek dla każdej specjalności

```
go
create function POKAZ_PRACOWNIKOW (@nazwa_sp varchar(14))
returns varchar(15)
as
begin
declare @placa int

select @placa = (select avg(placa_calkowita) from pracownik,specjalnosci
where
specjalnosci.id_specjalnosci=pracownik.id_specjalnosci and nazwa_specjalnosci=@nazwa_sp)

return cast(@placa as varchar(15))
end
```

	nazwa_specjalnosci	zarobek
1	Laryngolog	4500
2	Pediatra	8650
3	Ortopeda	2200
4	Pulmonolog	6000
5	Ginekolog	3500
6	Okulista	6700
7	Kardiolog	9000
8	Fizjolog	NULL
9	Reumatolog	NULL

# 5.Wyzwalacze

## 5.1. Dodaje lekarza , jeżeli nie ma żadnego o takim imieniu i nazwisku [INSERT]

```
create trigger dod_lekarz
on Szpital.pracownik
instead of insert
as
    if not exists(
        select id_pracownika
        from Szpital.pracownik
        where imie in(select imie from inserted)
            and nazwisko in(select nazwisko from inserted))
    begin
        insert into Szpital.pracownik (id_pracownika ,id_specjalnosci,imie
,nazwisko,placa_calkowita)
        select id_pracownika ,id_specjalnosci,imie ,nazwisko,placa_calkowita from inserted
    end
    else
    begin
        print 'Lekarz o tych danych juz istnieje'
    end
```

Przed operacją.

	id_pracownika	id_specjalnosci	imie	nazwisko	placa_calkowita
1	1	1	Andrzej	Lavoide	4500
2	2	2	Michal	Dandrewicz	7300
3	3	3	Adrian	Kreatywny	2200
4	4	4	Krzysztof	Staff	6000
5	5	5	Mirosław	Stary	3500
6	6	6	Jedrzej	Dropicki	6700
7	7	7	Lukasz	Walewski	9000
8	8	2	Marcin	Lavoide	10000



Po operacji.

	id_pracownika	id_specjalnosci	imie	nazwisko	placa_calkowita
1	1	1	Andrzej	Lavoide	4500
2	2	2	Michal	Dandrewicz	7300
3	3	3	Adrian	Kreatywny	2200
4	4	4	Krzysztof	Staff	6000
5	5	5	Mirosław	Stary	3500
6	6	6	Jedrzej	Dropicki	6700
7	7	7	Lukasz	Walewski	9000
8	8	2	Marcin	Lavoide	10000
9	9	2	Tester	Testowy	13000

## 5.2.W przypadku usunięcia specjalności z szpitala dodaje wpis[DELETE]

go

create trigger archiwizacja

on specjalnosci

after delete

as

begin

declare @specjalnosc varchar(14) = (select nazwa\_specjalnosci from deleted)

insert into zdarzenia values('Usunieto specjalnosc : '+@specjalnosc+' w dniu :

' + cast(GETDATE() as varchar(20)))

end

Przed operacją.

	id_specjalnosci	nazwa_specjalnosci
1	1	Laryngolog
2	2	Pediatra
3	3	Ortopeda
4	4	Pulmonolog
5	5	Ginekolog
6	6	Okulista
7	7	Kardiolog
8	8	Fizjolog
9	9	Reumatolog

Po operacji.

	id_zdarzenia	opis_zdarzenia
1	1	Usunieto specjalnosc : Kardiolog w dniu : May 27...

	id_specjalnosci	nazwa_specjalnosci
1	1	Laryngolog
2	2	Pediatra
3	3	Ortopeda
4	4	Pulmonolog
5	5	Ginekolog
6	6	Okulista
7	8	Fizjolog
8	9	Reumatolog

### 5.3. W momencie aktualizacji płacy pracownika dodaje wpis do dziennika zdarzeń

go

create trigger aktualizacja

on pracownik

after update

as

begin

declare @pracownik int = (select id\_pracownika from deleted)

declare @placa\_old int = (select placa\_calkowita from deleted)

declare @placa\_new int = (select placa\_calkowita from inserted)

declare @diff int = @placa\_new - @placa\_old

insert into zdarzenia values('Pracownik ' + cast(@pracownik as varchar(3)) + 'otrzymał

podwyżkę w wysokości ' + cast(@diff as varchar(9)))

end

Przed operacją.

	id_pracownika	imie	nazwisko	placa_calkowita
1	1	Andrzej	Lavoide	7000

Po operacji.

	id_zdarzenia	opis_zdarzenia
1	3	Pracownik 1otrzymał podwyżkę w wysokosci 1000

	id_pracownika	imie	nazwisko	placa_calkowita
1	1	Andrzej	Lavoide	8000