Projekt

Przychodnia

Zaawansowane Systemy Baz Danych

Dawid Michałowski (203942)
Robert Radczyc (203976)

Zajęcia: **środa, 8:30**
Rok akademicki **2016/2017**

Spis treści

1. Opis systemu.

Zaprojektowana została baza danych przykładowej przychodni. W danej przychodni mamy pracowników, niektórzy z nich są lekarzami. Pracownicy nie będący lekarzami dbają o porządek pokoju (osoby zarządzające). Każdy lekarz ma swój pokój. Lekarz posiadają również specjalizacje. Jeden lekarz otrzymuje premię do pensji w zależności od posiadanych specjalizacji. Do przychodni przychodzą pacjenci na wizyty. Wizyta odbywa się w pokoju lekarza prowadzącego wizytę. Podczas wizyty badany jest pacjent w celu ustalenia choroby. Podczas wizyty diagnozowane jest występowanie choroby. Każda choroba może być leczona lekarstwami.

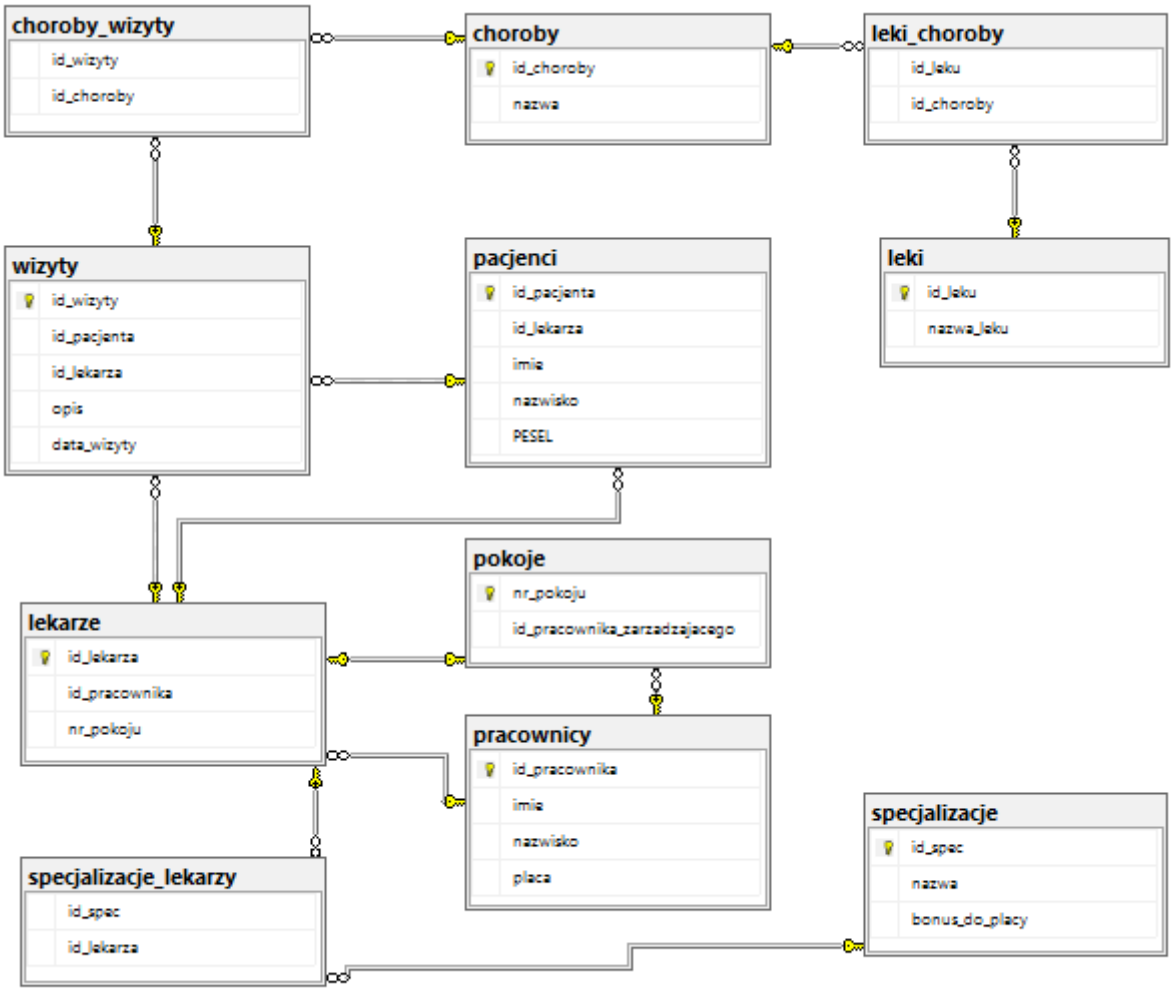
2. Cel utworzenia bazy.

Zwiększenie wydajności przychodni poprzez organizację pracy na serwerze, śledzenie historii przychodni, dzięki, której szybciej mogą zostać zdiagnozowane choroby danych pacjentów biorąc pod uwagę poprzednie choroby (historia wizyt).

3. Założenia.

- Lekarz posiada wiele specjalizacji, tak samo jedną specjalizację może mieć wiele lekarzy. Problem został rozwiązany za pomocą tabeli specjalizacje_lekarzy łączącej specjalizacje z lekarzami;
- Lekarz bez specjalizacji ma pensję równą 2000zł (bez żadnej specjalizacji).
- Choroby mogę być leczone wieloma lekarstwami, jedno lekarstwo może pomagać w leczeniu wielu chorób (tabela leki_choroby);
- Każdy lekarz posiada swój osobisty pokój, który jest zarządzany przez pracownika nie-lekarza.
- Wizyta zawsze odbywa się w pokoju danego lekarza. Wizyta nie może odbywać się w pokoju innego lekarza, niż prowadzącego wizytę;
- Każdy pacjent posiada osobistego lekarza, co nie oznacza, że nie może odbywać wizyty u innego lekarza.

4. Schemat bazy danych.



Rysunek 1. Schemat przedstawiający połączenia pomiędzy tabelami w bazie danych. Każdy kluczyk to Foreign Key, zaś każda żarówka to Primary Key.

5. Struktura.

Tabela 1. Pracownicy.

```
CREATE TABLE przychodnia..pracownicy (
id_pracownika INT IDENTITY(1,1) CONSTRAINT pracownik_PK PRIMARY KEY,
imie VARCHAR(20) NOT NULL,
nazwisko VARCHAR(20) NOT NULL,
placa INT NOT NULL
);
```

Tabela 2. Specjalizacje.

```
CREATE TABLE przychodnia..specjalizacje (
id_spec INT IDENTITY(1,1) CONSTRAINT spec_PK PRIMARY KEY,
nazwa VARCHAR(20) NOT NULL UNIQUE,
bonus_do_placy INT NOT NULL
);
```

Tabela 3. Pokoje.

```
CREATE TABLE przychodnia..pokoje (
nr_pokoju INT CONSTRAINT pokoj_PK PRIMARY KEY,
id_pracownika_zarzadzajacego INT NOT NULL,
CONSTRAINT id_pracownika_zarzadzajacego_FK FOREIGN KEY(id_pracownika_zarzadzajacego)
REFERENCES przychodnia..pracownicy(id_pracownika)
);
```

Tabela 4. Lekarze.

```
CREATE TABLE przychodnia..lekarze (
id_lekarza INT IDENTITY(1,1) CONSTRAINT lekarz_PK PRIMARY KEY,
id_pracownika INT NOT NULL,
nr_pokoju INT NOT NULL UNIQUE,
CONSTRAINT pracownik_lekarza_FK FOREIGN KEY(id_pracownika) REFERENCES
przychodnia..pracownicy(id_pracownika),
CONSTRAINT nr_pokoju_lekarza_FK FOREIGN KEY(nr_pokoju) REFERENCES
przychodnia..pokoje(nr_pokoju)
);
```

Tabela 5. Specjalizacje lekarzy.

```
CREATE TABLE przychodnia..specjalizacje_lekarzy (
id_spec INT NOT NULL,
id_lekarza INT NOT NULL,
CONSTRAINT lek_spec_FK FOREIGN KEY(id_lekarza) REFERENCES przychodnia..lekarze(id_lekarza),
CONSTRAINT spec_lek_FK FOREIGN KEY(id_spec) REFERENCES przychodnia..specjalizacje(id_spec)
);
```

Tabela 6. Pacjenci.

```
CREATE TABLE przychodnia..pacjenci (
id_pacjenta INT IDENTITY(1,1) CONSTRAINT pacjent_PK PRIMARY KEY,
id_lekarza INT NOT NULL,
imie VARCHAR(20) NOT NULL,
nazwisko VARCHAR(20) NOT NULL,
PESEL VARCHAR(11),
CONSTRAINT PESEL_pacjenta_FK CHECK (PESEL LIKE '%[0-9]%'),
CONSTRAINT lekarz_pacjenta_FK FOREIGN KEY(id_lekarza) REFERENCES
przychodnia..lekarze(id_lekarza)
);
```

Tabela 7. Wizyty.

```
CREATE TABLE przychodnia..wizyty (
id_wizyty INT IDENTITY(1,1) CONSTRAINT wizyta_PK PRIMARY KEY,
id_pacjenta INT NOT NULL,
id_lekarza INT NOT NULL,
opis VARCHAR(500),
data_wizyty DATETIME NOT NULL,
CONSTRAINT lek_wizyty_FK FOREIGN KEY(id_lekarza) REFERENCES
przychodnia..lekarze(id_lekarza),
CONSTRAINT pacjent_wizyty_FK FOREIGN KEY(id_pacjenta) REFERENCES
przychodnia..pacjenci(id_pacjenta)
);
```

Tabela 8. Leki.

```
CREATE TABLE przychodnia..leki (
id_leku INT IDENTITY(1,1) CONSTRAINT lek_PK PRIMARY KEY,
nazwa_leku VARCHAR(32)
);
```

Tabela 9. Choroby.

```
CREATE TABLE przychodnia..choroby (
id_choroby INT IDENTITY(1,1) CONSTRAINT choroba_PK PRIMARY KEY,
nazwa VARCHAR(32)
);
```

Tabela 10. Leki choroby.

```
CREATE TABLE przychodnia..leki_choroby (
id_leku INT NOT NULL,
id_choroby INT NOT NULL,
CONSTRAINT lek_choroby_FK FOREIGN KEY(id_leku) REFERENCES przychodnia..leki(id_leku),
CONSTRAINT choroba_leku_FK FOREIGN KEY(id_choroby) REFERENCES
przychodnia..choroby(id_choroby)
);
```

Tabela 11. Choroby wizyty.

```
CREATE TABLE przychodnia..choroby_wizyty (
id_wizyty INT NOT NULL,
id_choroby INT NOT NULL,
CONSTRAINT wizyta_choroby_FK FOREIGN KEY(id_wizyty) REFERENCES
przychodnia..wizyty(id_wizyty),
CONSTRAINT choroba_wiizyty_FK FOREIGN KEY(id_choroby) REFERENCES
przychodnia..choroby(id_choroby)
);
```

6. Przykładowe dane.

```
INSERT INTO pracownicy(imie,nazwisko,placa) VALUES
('Dawid','Rakieta',1400),--normalny
('Martinez','Lopez',1500),--normalny
('Karol','Balon',1200),--normalny
('Barbara','Wafel',3200),--alergolog
('Paweł','Tergog',4600),--laryngolog,alergolog
('Michał','Lambada',2000),--lekarz
('Karolina','Rodnik',7200)--kazda specjalizacja

INSERT INTO specjalizacje(nazwa,bonus_do_placy) VALUES
('Laryngolog',1400),
('Kardiolog',1500),
('Pediatra',1100),
('Alergolog',1200)

INSERT INTO pokoje(nr_pokoju,id_pracownika_zarzadzajacego) VALUES
(100,1),
(101,1),
(102,2),
(103,1),
(104,3),
(105,2),
(106,2),
(107,3),
(108,3)

INSERT INTO lekarze(id_pracownika,nr_pokoju) VALUES
(4,100),
(5,102),
(6,104),
(7,105)

INSERT INTO specjalizacje_lekarzy(id_lekarza,id_spec) VALUES
(1,4),
(2,1),
(2,4),
(4,1),
(4,2),
(4,3),
(4,4)

INSERT INTO pacjenci(id_lekarza,imie,nazwisko,PESEL) VALUES
(1,'Danuta','Benger','55080803397'),
(2,'Kamila','Koń','58040107122'),
(3,'Bartosz','Keke','20041813638'),
(3,'Ireneusz','Polipeus','80100809185'),
(1,'Eliasz','Nielodom','78022314045'),
(1,'Konrad','Bezele','01051711534')

INSERT INTO wizyty(id_pacjenta,id_lekarza,opis,data_wizyty) VALUES
(1,1,'Wykryto uczulenia na kilka zwierząt','2016/05/21'),
(1,1,'Powstanie dodatkowego uczulenia','2016/06/20'),
(2,1,'Brak jakichkolwiek uczuleń','2016/07/01'),
(2,2,'Nasilające się bóle ucha','2016/05/15'),
(3,2,'Wykryto kilka alergii na rośliny i zwierzęta','2016/08/07'),
(4,3,'Wysoka gorączka, dreszcze, kaszel, bóle głowy','2016/09/29'),
(5,4,'Bóle serca, nasilające się','2016/05/29'),
(6,4,'Ostry ból gardła','2016/12/12'),
(4,4,'Wysoka gorączka, bóle głowy','2017/01/04'),
(1,1,'Pacjent symulował','2017/01/05')
```

```sql
INSERT INTO leki(nazwa_leku) VALUES
('Orefor'),
('SerceFixer'),
('AlergMinimizer'),
('AlergMoc'),
('Zwierzalerg'),
('Rivenal'),
('Augumantin'),
('Bakatar'),
('Nitroglicernix')

INSERT INTO choroby(nazwa) VALUES
('Niedoczynność przestawki komory'),
('Alergia: koty'),
('Alergia: chomiki'),
('Alergia: króliki'),
('Alergia: zboża'),
('Alergia: róże'),
('Zapalenie gardła'),
('Zapalenie ucha środkowego'),
('Zapalenie płuc')

INSERT INTO leki_choroby(id_leku,id_choroby) VALUES
(1,7),
(2,1),
(3,2),
(3,3),
(4,3),
(3,4),
(3,5),
(3,6),
(4,2),
(4,3),
(4,4),
(4,5),
(5,2),
(5,3),
(5,4),
(6,8),
(7,7),
(7,8),
(7,9),
(8,7),
(9,1)

INSERT INTO choroby_wizyty(id_wizyty,id_choroby) VALUES
(1,2),
(1,3),
(2,4),
(4,8),
(5,2),
(5,3),
(5,5),
(5,6),
(6,9),
(7,1),
(8,7),
(9,9)
```

7. Zapytania.

Zapytanie 1. Wybierz nazwiska pracowników - lekarzy, którzy przepisali Augumantin.

```
SELECT DISTINCT nazwisko AS 'Nazwisko - Augumantin' FROM pracownicy p
INNER JOIN lekarze l on l.id_pracownika = p.id_pracownika
INNER JOIN wizyty w on w.id_lekarza = l.id_lekarza
INNER JOIN choroby_wizyty cw on w.id_wizyty = cw.id_wizyty
INNER JOIN choroby c on c.id_choroby = cw.id_choroby
INNER JOIN leki_choroby lc on lc.id_choroby = c.id_choroby
INNER JOIN leki lek on lek.id_leku = lc.id_leku
WHERE lek.nazwa_leku = 'Augumantin'
```

Rezultat:

	Nazwisko - Augumantin
1	Lambada
2	Rodnik
3	Tergog

Zapytanie 2. Ilość chorób zdiagnozowanych w roku 2016.

```
SELECT COUNT(c.id_choroby) AS 'Liczba chorób 2016' FROM choroby c, choroby_wizyty cw, wizyty
w WHERE
c.id_choroby = cw.id_choroby AND cw.id_wizyty = w.id_wizyty AND
FORMAT(w.data_wizyty, 'yyyy') = '2016'
```

Rezultat:

	Liczba chorób 2016
1	11

Zapytanie 3. Podstawowa pensja każdego lekarza (bez bonusu specjalizacji), nazwisko, pensja.

```
--3. podstawowa pensja kazdego lekarza (bez bonusu specjalizacji), nazwisko, pensja
SELECT p.nazwisko AS 'Nazwisko', p.placa-SUM(ISNULL(s.bonus_do_placy,0)) AS 'Placa bez
stanowiska' FROM
pracownicy p
INNER JOIN lekarze l ON l.id_pracownika = p.id_pracownika
LEFT JOIN specjalizacje_lekarzy sl ON l.id_lekarza = sl.id_lekarza
LEFT JOIN specjalizacje s ON s.id_spec = sl.id_spec
GROUP BY p.nazwisko, p.placa
```

Rezultat:

	Nazwisko	Placa bez stanowiska
1	Lambada	2000
2	Wafel	2000
3	Tergog	2000
4	Rodnik	2000

Zapytanie 4. Imię i nazwisko wszystkich pacjentów urodzonych w kwietniu.

```
SELECT CONCAT(p.imie,' ', p.nazwisko) AS 'Imie Nazwisko'
FROM pacjenci p WHERE
SUBSTRING(p.PESEL,3,2) = '04'
```

Rezultat:

	Imie Nazwisko
1	Kamila Koń
2	Bartosz Keke

Zapytanie 5. Nazwisko pacjenta u którego stwierdzono ta sama chorobę w przeciągu 6 miesięcy.

```
SELECT DISTINCT p.nazwisko AS 'Nazwisko'
FROM pacjenci p, wizyty w1, wizyty w2, choroby ch, choroby_wizyty cw, choroby ch2,
choroby_wizyty cw2
WHERE p.id_pacjenta = w1.id_pacjenta
AND p.id_pacjenta = w2.id_pacjenta
AND w1.id_wizyty != w2.id_wizyty
AND w1.id_wizyty = cw.id_wizyty
AND cw.id_choroby = ch.id_choroby
AND w2.id_wizyty = cw2.id_wizyty
AND cw2.id_choroby = ch2.id_choroby
AND ch.id_choroby = ch2.id_choroby
AND DATEDIFF(m,w1.data_wizyty,w2.data_wizyty) <= 6 AND
DATEDIFF(m,w1.data_wizyty,w2.data_wizyty) >= -6
Ad. Dwa datediffy, jedna data jest o 6 miesięcy starsza a druga o 6 miesięcy młodsza,
dlatego przedział jest od -6 do 6
```

Rezultat:

	Nazwisko
1	Polipeus

Zapytanie 6. Opis wizyty i nazwisko pacjenta, któremu nie zdiagnozowano choroby podczas wizyty.

```
SELECT p.nazwisko AS 'Nazwisko', w.opis AS 'Opis wizyty'
FROM pacjenci p, wizyty w
WHERE w.id_pacjenta = p.id_pacjenta
AND w.id_wizyty NOT IN(SELECT id_wizyty FROM choroby_wizyty)
ORDER BY p.nazwisko
```

Rezultat:

	Nazwisko	Opis wizyty
1	Benger	Pacjent symulował
2	Koń	Brak jakichkolwiek uczuleń

Zapytanie 7. Imię, nazwisko i data urodzenia (z PESELu) pacjenta.

```
SELECT p.imie AS 'Imie', p.nazwisko AS 'Nazwisko', CAST(CONCAT(
CASE
       WHEN SUBSTRING(p.PESEL,1,2) <= SUBSTRING(CAST(DATEPART(YY,GETDATE()) AS
VARCHAR(4)),3,2)
              THEN '20'
       ELSE '19'
END
,SUBSTRING(p.PESEL,1,2),'/',SUBSTRING(p.PESEL,3,2),'/',SUBSTRING(p.PESEL,5,2))AS DATE) AS
'Data urodzenia'
FROM pacjenci p
```

Rezultat:

	Imie	Nazwisko	Data urodzenia
1	Danuta	Benger	1955-08-08
2	Kamila	Koń	1958-04-01
3	Bartosz	Keke	1920-04-18
4	Irene...	Polipeus	1980-10-08
5	Eliasz	Nielodom	1978-02-23
6	Konrad	Bezele	2001-05-17

Zapytanie 8. Alergie zdiagnozowane przez danego lekarza.

```
SELECT p.nazwisko AS 'Nazwisko', STUFF(
                    (SELECT
                         ', ' + SUBSTRING(ch.nazwa,10,LEN(ch.nazwa)-9)
                         FROM  lekarze l, wizyty w, choroby_wizyty cw, choroby ch
                                WHERE
                                       p.id_pracownika = l.id_pracownika AND
                                       l.id_lekarza = w.id_lekarza AND
                                       w.id_wizyty = cw.id_wizyty AND
                                       ch.id_choroby = cw.id_choroby AND
                                       ch.nazwa LIKE 'Alergia%'
                                ORDER BY ch.nazwa
                         FOR XML PATH(''), TYPE
                    ).value('.','varchar(max)')
                    ,1,2, ''
                ) AS 'Alergie'
FROM pracownicy p, lekarze l2, wizyty w2, choroby_wizyty cw2, choroby ch2
WHERE
       p.id_pracownika = l2.id_pracownika AND
       l2.id_lekarza = w2.id_lekarza AND
       w2.id_wizyty = cw2.id_wizyty AND
       ch2.id_choroby = cw2.id_choroby AND
       ch2.nazwa LIKE 'Alergia%'
GROUP BY p.nazwisko, p.id_pracownika
```

Rezultat:

	Nazwisko	Alergie
1	Wafel	chomiki, koty, króliki
2	Tergog	chomiki, koty, róże, zboża

Zapytanie 9. Wizyty, które odbyły się w miesiącu z liczba dni 30.

```
SELECT p.nazwisko AS 'Lekarz', pa.nazwisko AS 'Pacjent', w.opis AS 'Opis'
FROM wizyty w, lekarze l, pracownicy p, pacjenci pa
WHERE w.id_lekarza = l.id_lekarza AND p.id_pracownika = l.id_pracownika AND w.id_pacjenta =
pa.id_pacjenta
AND DAY(DATEADD(DD,-1,DATEADD(MM,DATEDIFF(MM,-1,w.data_wizyty),0))) = 30
```

Rezultat:

	Lekarz	Pacjent	Opis
1	Wafel	Benger	Powstanie dodatkowego uczulenia
2	Lambada	Polipeus	Wysoka gorączka, dreszcze, kaszel, bóle głowy

Zapytanie 10. Płace i nazwiska pracowników (nie-lekarzy), którzy zarabiają więcej niż 1/3 średniej płacy lekarzy.

```
SELECT p.nazwisko AS 'Nazwisko', p.placa AS 'Placa'
FROM pracownicy p
LEFT JOIN lekarze lek ON lek.id_pracownika = p.id_pracownika
WHERE p.placa > (SELECT AVG(p2.placa)/3 FROM  pracownicy p2, lekarze l WHERE
p2.id_pracownika = l.id_pracownika)
AND lek.id_lekarza IS NULL
```

Rezultat:

	Nazwisko	Placa
1	Lopez	1500

Zapytanie 11. Pacjenci urodzeni w roku przestępnym.

```
SELECT p.nazwisko AS 'Nazwisko'
FROM pacjenci p
WHERE CAST(CONCAT(
CASE
	WHEN SUBSTRING(p.PESEL,1,2) <= SUBSTRING(CAST(DATEPART(YY,GETDATE()) AS
VARCHAR(4)),3,2)
		THEN '20'
	ELSE '19'
END
,SUBSTRING(p.PESEL,1,2)) AS FLOAT) / CAST(4 AS FLOAT) -
CAST(CONCAT(
CASE
	WHEN SUBSTRING(p.PESEL,1,2) <= SUBSTRING(CAST(DATEPART(YY,GETDATE()) AS
VARCHAR(4)),3,2)
		THEN '20'
	ELSE '19'
END
,SUBSTRING(p.PESEL,1,2)) AS INT) / 4 = 0
```

Rezultat:

	Nazwisko
1	Keke
2	Polipeus

Zapytanie 12. Ile w bazie jest emerytów (65+) oraz dzieci (-18).

```
SELECT COUNT(p1.id_pacjenta) AS 'Emeryci',
	(SELECT COUNT(p2.id_pacjenta) FROM pacjenci p2 WHERE
		DATEDIFF(YY,CAST(CONCAT(
		CASE
			WHEN SUBSTRING(p2.PESEL,1,2) <= SUBSTRING(CAST(DATEPART(YY,GETDATE())
AS VARCHAR(4)),3,2)
				THEN '20'
			ELSE '19'
		END
		,SUBSTRING(p2.PESEL,1,2),'-',SUBSTRING(p2.PESEL,3,2),'-',SUBSTRING(p2.PESEL,5,
2))AS DATE), CONVERT(datetimeoffset, GETDATE())) < 18
) AS 'Dzieci'
FROM pacjenci p1
WHERE DATEDIFF(YY,CAST(CONCAT(
CASE
	WHEN SUBSTRING(p1.PESEL,1,2) <= SUBSTRING(CAST(DATEPART(YY,GETDATE()) AS
VARCHAR(4)),3,2)
		THEN '20'
	ELSE '19'
END
,SUBSTRING(p1.PESEL,1,2),'-',SUBSTRING(p1.PESEL,3,2),'-',SUBSTRING(p1.PESEL,5,2))AS DATE),
CONVERT(datetimeoffset, GETDATE())) > 65
```

Rezultat:

	Emeryci	Dzieci
1	1	1

Zapytanie 13. Ile pacjentów odwiedziło lekarza, w ciągu ostatniego roku, który nie ma żadnej specjalizacji.

```
SELECT DISTINCT COUNT(p.id_pacjenta) AS 'Pacjenci lekarza bez spec. w ciagu ostatniego roku'
FROM pacjenci p
INNER JOIN wizyty w ON w.id_pacjenta = p.id_pacjenta
INNER JOIN lekarze l ON w.id_lekarza = l.id_lekarza
LEFT JOIN specjalizacje_lekarzy sl ON sl.id_lekarza = l.id_lekarza
WHERE sl.id_spec IS NULL AND DATEDIFF(d,w.data_wizyty ,GETDATE())/365.25 < 1
```

Rezultat:

	Pacjenci lekarza bez spec. w ciagu ostatniego roku
1	1

Zapytanie 14. Lekarze zarabiający więcej niż lekarz, który ma najwięcej przyjęć.

```
SELECT p.nazwisko AS 'Nazwisko'
FROM pracownicy p, lekarze l
WHERE
p.id_pracownika = l.id_pracownika AND
p.placa > (SELECT TOP 1 best.placa FROM (
SELECT TOP 1 p2.id_pracownika AS idprac, COUNT(w2.id_wizyty) AS wiz, p2.placa AS placa FROM
pracownicy p2, lekarze l2, wizyty w2 WHERE
p2.id_pracownika = l2.id_pracownika AND w2.id_lekarza = l2.id_lekarza GROUP BY
p2.id_pracownika, placa ORDER BY wiz DESC) best)
```

Rezultat:

	Nazwisko
1	Tergog
2	Rodnik

Zapytanie 15. Pokoje, w których stwierdzono Zapalenie ucha środkowego (przez lekarza specjalistę (taki który ma specjalizacje)) lub Zapalenie płuc (stwierdzone przez lekarza bez specjalizacji).

```
SELECT pok.nr_pokoju AS 'Nr pokoju'
FROM pokoje pok, lekarze l, wizyty w, choroby_wizyty cw, choroby c WHERE
l.nr_pokoju = pok.nr_pokoju AND w.id_lekarza = l.id_lekarza AND
w.id_wizyty = cw.id_wizyty AND cw.id_choroby = c.id_choroby AND
((c.nazwa = 'Zapalenie ucha środkowego' AND l.id_lekarza IN (SELECT sc.id_lekarza FROM
specjalizacje_lekarzy sc))
OR (c.nazwa = 'Zapalenie płuc' AND l.id_lekarza NOT IN(SELECT sc.id_lekarza FROM
specjalizacje_lekarzy sc)))
```

Rezultat:

	Nr pokoju
1	102
2	104

8. Funkcje.

Funkcja 1. Funkcja wyliczająca datę z PESELu.

```
IF object_id(N'zwroc_date_urodzenia_z_PESELu', N'FN') IS NOT NULL
    DROP FUNCTION dbo.zwroc_date_urodzenia_z_PESELu
GO
CREATE FUNCTION dbo.zwroc_date_urodzenia_z_PESELu (@PESEL VARCHAR(20))
RETURNS DATE
AS
BEGIN
declare @dataurodzenia as DATE

  set @dataurodzenia = (CONVERT(DATE,CONCAT(
CASE
       WHEN SUBSTRING(@PESEL,1,2) <= SUBSTRING(CAST(DATEPART(YY,GETDATE()) AS
VARCHAR(4)),3,2)
              THEN '20'
       ELSE '19'
END
,SUBSTRING(@PESEL,1,2),'-',SUBSTRING(@PESEL,3,2),'-',SUBSTRING(@PESEL,5,2))))
return @dataurodzenia
END
--TEST
GO
SELECT dbo.zwroc_date_urodzenia_z_PESELu(p.PESEL) AS 'Data ur.', p.PESEL FROM pacjenci p;
```

Rezultat:

	Data ur.	PESEL
1	1955-08-08	55080803397
2	1958-04-01	58040107122
3	1920-04-18	20041813638
4	1980-10-08	80100809185
5	1978-02-23	78022314045
6	2001-05-17	01051711534

9. Procedury.

Procedura 1. Procedura dodająca lekarza (pracownika).

```
CREATE OR ALTER PROCEDURE dodaj_lekarza
	@imie VARCHAR(20),
	@nazwisko VARCHAR(20),
	@nrpokoju INT
AS
	DECLARE @ID INT;
	INSERT INTO pracownicy(imie,nazwisko,placa) VALUES (@imie,@nazwisko,2000);
	SET @ID = (SELECT TOP 1 p.id_pracownika FROM pracownicy p ORDER BY p.id_pracownika
DESC);
	IF (EXISTS(SELECT TOP 1 * FROM pokoje p WHERE p.nr_pokoju = @nrpokoju) AND
		NOT EXISTS(SELECT TOP 1 * FROM lekarze l WHERE l.nr_pokoju = @nrpokoju))
		INSERT INTO lekarze(nr_pokoju,id_pracownika) VALUES (@nrpokoju,@ID);
	ELSE
		DELETE FROM pracownicy WHERE id_pracownika = @ID;
	SELECT * FROM pracownicy p
	INNER JOIN lekarze l ON l.id_pracownika = p.id_pracownika
GO
--TEST
EXEC dodaj_lekarza @imie = 'Dadeko', @nazwisko = 'Koreo', @nrpokoju = 101;
```

Rezultat:

	id_pracownika	imie	nazwisko	placa	id_lekarza	id_pracownika	nr_pokoju
1	4	Barbara	Wafel	3200	1	4	100
2	5	Paweł	Tergog	4600	2	5	102
3	6	Michał	Lambada	2000	3	6	104
4	7	Karolina	Rodnik	7200	4	7	105
5	8	Dadeko	Koreo	2000	5	8	101

Procedura 2. Dodanie specjalizacji lekarzowi (z wzięciem pod uwagę bonusu do płacy).

```
CREATE OR ALTER PROCEDURE dodaj_spec_lekarzowi_i_zwieksz_pensje
	@idLekarza INT,
	@nazwaSpec VARCHAR(20)
AS
	SELECT * FROM lekarze l
	INNER JOIN pracownicy p ON l.id_pracownika = p.id_pracownika
	INNER JOIN specjalizacje_lekarzy sl ON l.id_lekarza = sl.id_lekarza
	INNER JOIN specjalizacje s ON sl.id_spec = s.id_spec
	WHERE l.id_lekarza = @idLekarza;
	DECLARE @IDSPEC INT;
	DECLARE @idPracownika INT;
	DECLARE @bonus INT;
	SET @bonus = (SELECT TOP 1 bonus_do_placy FROM specjalizacje WHERE nazwa =
@nazwaSpec);
	SET @IDSPEC = (SELECT TOP 1 id_spec FROM specjalizacje WHERE nazwa = @nazwaSpec);
	SET @idPracownika = (SELECT TOP 1 id_pracownika FROM lekarze WHERE id_lekarza =
@idLekarza);
	IF (EXISTS(SELECT TOP 1 * FROM specjalizacje WHERE nazwa = @nazwaSpec) AND
EXISTS(SELECT TOP 1 * FROM lekarze WHERE id_lekarza = @idLekarza))
	BEGIN
		INSERT INTO specjalizacje_lekarzy(id_lekarza, id_spec) VALUES (@idLekarza,
@IDSPEC);
		UPDATE pracownicy SET placa = placa + @bonus WHERE id_pracownika =
@idPracownika;
	END
	SELECT * FROM lekarze l
	INNER JOIN pracownicy p ON l.id_pracownika = p.id_pracownika
	INNER JOIN specjalizacje_lekarzy sl ON l.id_lekarza = sl.id_lekarza
```

```
        INNER JOIN specjalizacje s ON sl.id_spec = s.id_spec
        WHERE l.id_lekarza = @idLekarza;
GO
--TEST
EXEC dodaj_spec_lekarzowi_i_zwieksz_pensje @idLekarza = 1, @nazwaSpec = 'Laryngolog'
```

Rezultat:

	id_lekarza	id_pracownika	nr_pokoju	id_pracownika	imie	nazwisko	placa	id_spec	id_lekarza	id_spec	nazwa	bonus_do_placy
1	1	4	100	4	Barbara	Wafel	3200	4	1	4	Alergolog	1200

	id_lekarza	id_pracownika	nr_pokoju	id_pracownika	imie	nazwisko	placa	id_spec	id_lekarza	id_spec	nazwa	bonus_do_placy
1	1	4	100	4	Barbara	Wafel	4600	4	1	4	Alergolog	1200
2	1	4	100	4	Barbara	Wafel	4600	1	1	1	Laryngolog	1400

Procedura 3a. Usunięcie lekarza z bazy poprzez podanie jego id.

```
CREATE OR ALTER PROCEDURE usun_lekarza_pracownika_id
        @idLekarza INT
AS
        DECLARE @idPracownika INT;
        SET @idPracownika = (SELECT TOP 1 l.id_pracownika FROM lekarze l
                WHERE l.id_lekarza = @idLekarza)
        ALTER TABLE lekarze NOCHECK CONSTRAINT pracownik_lekarza_FK;
        ALTER TABLE specjalizacje_lekarzy NOCHECK CONSTRAINT ALL;
        ALTER TABLE wizyty NOCHECK CONSTRAINT ALL;
        ALTER TABLE pacjenci NOCHECK CONSTRAINT ALL;
        DELETE FROM lekarze WHERE id_lekarza = @idLekarza;
        DELETE FROM pracownicy WHERE id_pracownika = @idPracownika;
        DELETE FROM specjalizacje_lekarzy WHERE id_lekarza = @idLekarza;
                UPDATE pacjenci SET id_lekarza = (SELECT TOP 1 licznik.IDLEK FROM
                                (SELECT COUNT(p.id_lekarza) AS value, p.id_lekarza AS IDLEK FROM
pacjenci p GROUP BY p.id_lekarza) licznik ORDER BY value)
        WHERE id_lekarza = @idLekarza;
        ALTER TABLE pacjenci CHECK CONSTRAINT ALL;
        ALTER TABLE pacjenci CHECK CONSTRAINT ALL;
        ALTER TABLE specjalizacje_lekarzy CHECK CONSTRAINT ALL;
        ALTER TABLE lekarze CHECK CONSTRAINT pracownik_lekarza_FK;
GO
```

Procedura 3b. Usunięcie lekarza z bazy poprzez podanie jego imienia i nazwiska.

```
CREATE OR ALTER PROCEDURE usun_lekarza_pracownika_dane
        @imie VARCHAR(20),
        @nazwisko VARCHAR(20)
AS
        DECLARE @idLekarza2 INT;

        IF ((SELECT COUNT(*) FROM pracownicy WHERE imie = @imie AND nazwisko = @nazwisko) =
1)
        BEGIN
                SET @idLekarza2 = (SELECT TOP 1 l.id_lekarza FROM lekarze l, pracownicy p
                WHERE l.id_pracownika = p.id_pracownika AND p.imie = @imie AND p.nazwisko =
@nazwisko);
                EXEC usun_lekarza_pracownika_id @idLekarza = @idLekarza2;
        END
        ELSE
                PRINT 'Istnieje co najmniej dwóch lekarzy o takim samym imieniu i nazwisku,
usuń poprzez usun_lekarza_pracownika_id';
        SELECT * FROM pracownicy;
GO

EXEC usun_lekarza_pracownika_dane @imie ='Barbara', @nazwisko = 'Wafel';
```

Rezultat:

	id_pracownika	imie	nazwisko	placa
1	1	Dawid	Rakieta	1400
2	2	Martinez	Lopez	1500
3	3	Karol	Balon	1200
4	5	Paweł	Tergog	4600
5	6	Michał	Lambada	2000
6	7	Karolina	Rodnik	7200
7	8	Dadeko	Koreo	2000