*21.01.2013*

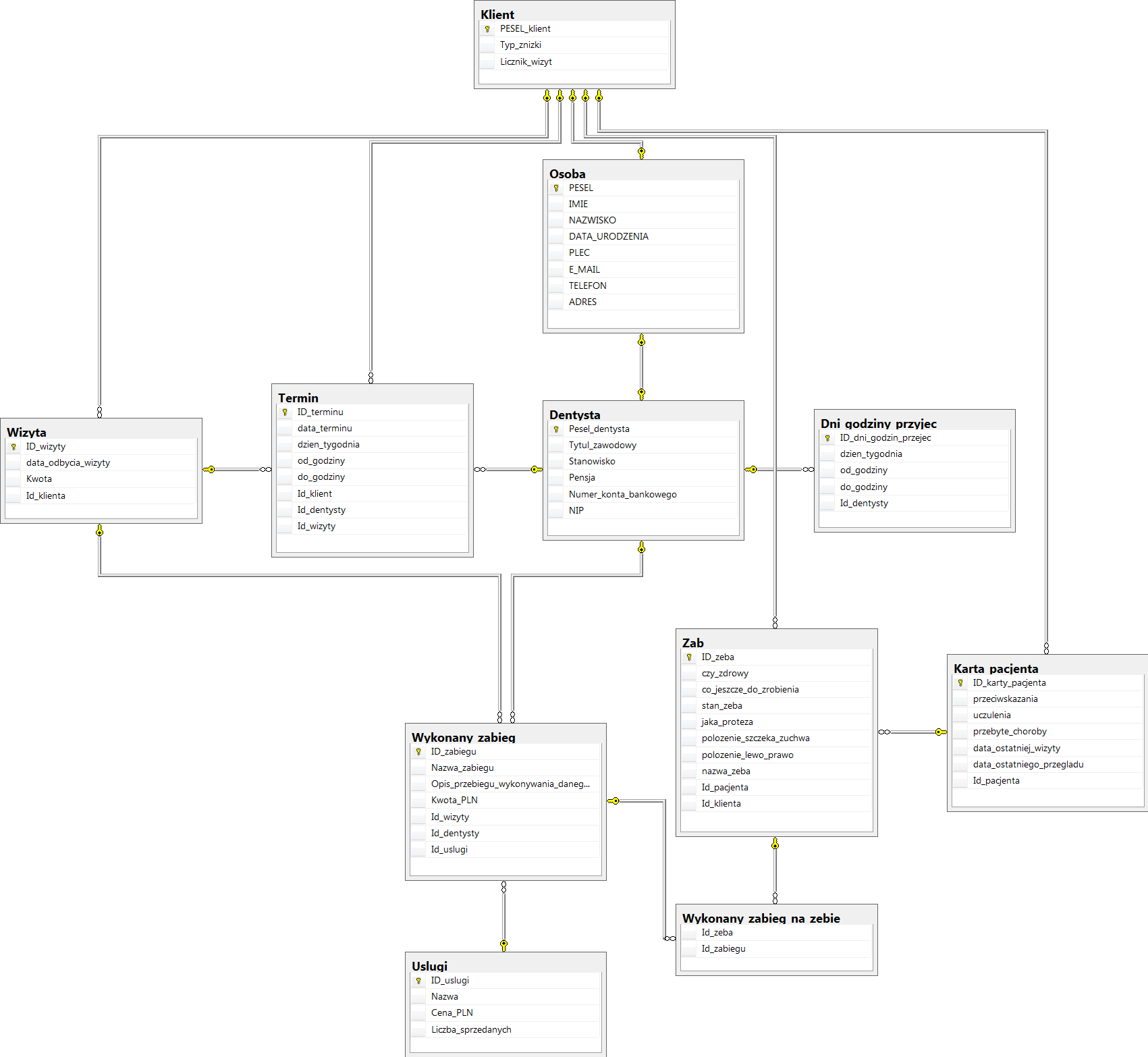
**Paweł Troka, nr indeksu: 132334, grupa dziekańska 5**

# Gabinet dentystyczny, klienci i ich terminy, płatności

Sprawozdanie

Bazy danych Laboratorium

1. Zadanie
   1. Graficzny Schemat Bazy Danych



* 1. Schemat RBD

**Osoba** (PESEL, IMIE, NAZWISKO, DATA\_URODZENIA, PLEC, E\_MAIL, TELEFON, ADRES)

**Klient** (PESEL\_klient REF Osoba, Typ\_znizki, Licznik\_wizyt)

**Dentysta** (Pesel\_dentysta REF Osoba, Tytul\_zawodowy, Stanowisko, Pensja, Numer\_konta\_bankowego, NIP)

**Karta\_pacjenta** (ID\_karty\_pacjenta, przeciwskazania, uczulenia, przebyte\_choroby, data\_ostatniej\_wizyty, data\_ostatniego\_przegladu, Id\_pacjenta REF Klient)

**Zab** (ID\_zeba, czy\_zdrowy, co\_jeszcze\_do\_zrobienia, stan\_zeba, jaka\_proteza, polozenie\_szczeka\_zuchwa, polozenie\_lewo\_prawo, nazwa\_zeba, Id\_pacjenta REF Karta\_pacjenta, Id\_klienta REF Klient)

**Dni\_godziny\_przyjec** (ID\_dni\_godzin\_przejec, dzien\_tygodnia, od\_godziny, do\_godziny, Id\_dentysty REF Dentysta)

**Wizyta** (ID\_wizyty, data\_odbycia\_wizyty, Kwota, Id\_klienta REF Klient)

**Uslugi** (ID\_uslugi, Nazwa 15), Cena\_PLN , Liczba\_sprzedanych)

**Wykonany\_zabieg** (ID\_zabiegu, Nazwa\_zabiegu, Opis\_przebiegu\_wykonywania\_danego\_zabiegu, Kwota\_PLN, Id\_wizyty REF Wizyta, Id\_dentysty REF Dentysta, Id\_uslugi REF Uslugi)

**Termin** (ID\_terminu, data\_terminu, dzien\_tygodnia, od\_godziny, do\_godziny, Id\_klient REF Klient, Id\_dentysty REF Dentysta, Id\_wizyty REF Wizyta)

**Wykonany\_zabieg\_na\_zebie** (Id\_zeba REF Zab, Id\_zabiegu REF Wykonany\_zabieg)

* 1. Instrukcje CREATE

CREATE TABLE Osoba

(PESEL bigint PRIMARY KEY,

IMIE nvarchar(35) NOT NULL,

NAZWISKO nvarchar(35) NOT NULL,

DATA\_URODZENIA DATE NOT NULL,

PLEC varchar(2) NOT NULL,

E\_MAIL varchar(40) NOT NULL,

TELEFON varchar(25) NOT NULL,

ADRES ntext);

CREATE TABLE Klient

(PESEL\_klient bigint PRIMARY KEY REFERENCES Osoba ON DELETE CASCADE NOT NULL,

Typ\_znizki varchar(10),

Licznik\_wizyt int);

CREATE TABLE Dentysta

(Pesel\_dentysta bigint PRIMARY KEY REFERENCES Osoba ON DELETE CASCADE NOT NULL,

Tytul\_zawodowy varchar(35),

Stanowisko varchar(35) NOT NULL,

Pensja money NOT NULL,

Numer\_konta\_bankowego varchar(70),

NIP bigint NOT NULL);

CREATE TABLE Karta\_pacjenta

(ID\_karty\_pacjenta bigint PRIMARY KEY,

przeciwskazania ntext,

uczulenia ntext,

przebyte\_choroby ntext,

data\_ostatniej\_wizyty DATE,

data\_ostatniego\_przegladu DATE,

Id\_pacjenta bigint REFERENCES Klient ON DELETE CASCADE NOT NULL);

CREATE TABLE Zab

(ID\_zeba int PRIMARY KEY,

czy\_zdrowy varchar(3) NOT NULL,

co\_jeszcze\_do\_zrobienia ntext,

stan\_zeba ntext,

jaka\_proteza varchar(35),

polozenie\_szczeka\_zuchwa nvarchar(15) NOT NULL,

polozenie\_lewo\_prawo varchar(15) NOT NULL,

nazwa\_zeba nvarchar(10) NOT NULL,

Id\_pacjenta bigint REFERENCES Karta\_pacjenta ON DELETE CASCADE NOT NULL,

Id\_klienta bigint REFERENCES Klient NOT NULL);

CREATE TABLE Dni\_godziny\_przyjec

(ID\_dni\_godzin\_przejec int PRIMARY KEY,

dzien\_tygodnia varchar(20) NOT NULL,

od\_godziny TIME NOT NULL,

do\_godziny TIME NOT NULL,

Id\_dentysty bigint REFERENCES Dentysta ON DELETE CASCADE NOT NULL);

CREATE TABLE Wizyta

(ID\_wizyty int PRIMARY KEY,

data\_odbycia\_wizyty DATE,

Kwota smallmoney,

Id\_klienta bigint REFERENCES Klient ON DELETE NO ACTION NOT NULL);

CREATE TABLE Uslugi

(ID\_uslugi int PRIMARY KEY,

Nazwa nvarchar(35), /\* nazwa nie jest NOT NULL, gdyż przewidujemy usługi bez nazwy \*/

Cena\_PLN smallmoney NOT NULL,

Liczba\_sprzedanych int);

CREATE TABLE Wykonany\_zabieg

(ID\_zabiegu int PRIMARY KEY,

Nazwa\_zabiegu nvarchar(20) NOT NULL,

Opis\_przebiegu\_wykonywania\_danego\_zabiegu ntext,

Kwota\_PLN smallmoney,

Id\_wizyty int REFERENCES Wizyta ON DELETE CASCADE NOT NULL,

Id\_dentysty bigint REFERENCES Dentysta ON DELETE CASCADE NOT NULL,

Id\_uslugi int REFERENCES Uslugi ON DELETE NO ACTION NOT NULL); /\*z. złożony z usługi\*/

CREATE TABLE Termin

(ID\_terminu int PRIMARY KEY,

data\_terminu DATE NOT NULL,

dzien\_tygodnia varchar(20),

od\_godziny TIME NOT NULL,

do\_godziny TIME NOT NULL,

Id\_klient bigint REFERENCES Klient ON DELETE CASCADE NOT NULL, /\* na termin jest zapisany klient \*/

Id\_dentysty bigint REFERENCES Dentysta ON DELETE NO ACTION NOT NULL, /\* do terminu przypisany jest dentysta \*/

Id\_wizyty int REFERENCES Wizyta ON DELETE SET NULL); /\* termin przechodzi w wizytę \*/

/\* \_\_ Relacje wiele do wiele \_\_ \*/

/\* zabieg na zebie \*/

CREATE TABLE Wykonany\_zabieg\_na\_zebie

(Id\_zeba int REFERENCES Zab ON DELETE CASCADE NOT NULL,

Id\_zabiegu int REFERENCES Wykonany\_zabieg ON DELETE NO ACTION NOT NULL);

* 1. Instrukcje SELECT

/\* widok + porządkowanie \*/

/\* wyświetli widok oferowanych usług wraz z nazwą i ceną, posortowanych według popularności (liczby sprzedanych) usług \*/

CREATE VIEW Oferowane\_uslugi

AS SELECT Nazwa, Cena\_PLN, Liczba\_sprzedanych

FROM Uslugi

WHERE Nazwa IS NOT NULL

WITH CHECK OPTION;

SELECT Nazwa, Cena\_PLN

FROM Oferowane\_uslugi

ORDER BY Liczba\_sprzedanych DESC;

/\* grupowanie + funkcja agregujaca \*/

/\* ile u nas wydał dany klient \*/

SELECT Id\_klienta, SUM(Kwota) AS 'Wydał u nas:'

FROM Wizyta

GROUP BY Id\_klienta

/\* podzapytanie i funkcja agregująca \*/

/\* ilu jeszcze klientów ma cokolwiek do zrobienia z zębami? \*/

SELECT COUNT(\*)

FROM Klient

WHERE PESEL\_Klient IN

(SELECT Id\_klienta

FROM Zab

WHERE co\_jeszcze\_do\_zrobienia IS NOT NULL);

/\* złączenie i podzapytanie \*/

/\* wypisuje pełne informacje osób które mają jeszcze coś do zrobienia z zębami \*/

SELECT \*

FROM Klient JOIN Osoba

ON Klient.PESEL\_klient = Osoba.PESEL

WHERE PESEL\_Klient IN

(SELECT Id\_klienta

FROM Zab

WHERE co\_jeszcze\_do\_zrobienia IS NOT NULL);

/\* złączenie \*/

/\* rezerwacje klientów na terminy (terminy które nie przeszły jeszcze w wizytę) \*/

SELECT PESEL, IMIE, NAZWISKO, data\_terminu, od\_godziny, do\_godziny

FROM Osoba JOIN Termin

ON Osoba.PESEL = Termin.Id\_klient

WHERE Termin.Id\_wizyty IS NULL;

/\* grupowanie \*/

/\* ilość chorych zębów każdego z klientów \*/

SELECT Id\_klienta,COUNT(\*) AS 'Ilość chorych zębów'

FROM Zab

WHERE czy\_zdrowy ='nie'

GROUP BY Id\_klienta;

/\* złączenie \*/

/\* wypisuje co dana osoba ma jeszcze do zrobienia z zębami \*/

SELECT IMIE, NAZWISKO, nazwa\_zeba, polozenie\_szczeka\_zuchwa, polozenie\_lewo\_prawo, co\_jeszcze\_do\_zrobienia

FROM Osoba JOIN Zab

ON Zab.Id\_klienta = Osoba.PESEL

WHERE co\_jeszcze\_do\_zrobienia IS NOT NULL;

1. Zadanie
   1. Wybrane transakcje wraz z wyborem poziomu izolacji transakcji i uzasadnieniem
      1. Dodanie nowego klienta wraz z kartą pacjenta i informacjami o zębach

W tym zadaniu wykonujemy szereg operacji, należy dodać do bazy nową osobę wraz ze wszystkimi danymi, następnie dodać klienta wraz z referencją do osoby (klucz obcy), dodajemy także kartę pacjenta z informacjami na temat klienta, oraz informację o każdym z jego zębów. Na koniec update’ujemy jeszcze kartę pacjenta – symulując odbywającą się wizytę typu przegląd (bez informacji o zabiegu – czysty przegląd). Stosunkowo duża złożoność, waga dokładności operacji (klienci to podstawa dla gabinetu dentystycznego) oraz nieczęste wykonywanie tej transakcji (dodawanie nowego klienta jest mniej częste niż inne operacje w bazie) zmuszają nas do zastosowania maksymalnego poziomu izolacji transakcji (**SERIALIZABLE**). Wymagany jest tak wysoki poziom izolacji transakcji, gdyż gdybyśmy dodawali innego użytkownika w tym samym czasie bez zabezpieczeń (transakcja przeszkadzająca), to bardzo łatwo byłoby nadpisać niektóre referencje (np. karta jednego pacjenta nadpisująca pod tym samym ID kartę innego pacjenta). Rezultat byłby więc tragiczny dla całej bazy – wspólna karta dwóch pacjentów oznaczałaby nadpisywanie informacji pierwszego klienta przez drugiego i późniejsze nieistnienie karty pacjenta pierwszego klienta. W prostszym przypadku (przedstawionym w kodzie transakcji) dodajemy równocześnie dwóch klientów, ich kartę pacjenta dodajemy pod to samo ID, w momencie kiedy dodajemy transakcją przeszkadzającą w trakcie gdy dodaje transakcja 1 bez zabezpieczeń, to część informacji z transakcji przeskadzającej jest dodanych (osoba i klient) – mają one referencję jednak do nadpisanej przez transakcje 1 bez zabezpieczeń karty pacjenta z transakcji 1. W przypadku gdy korzystamy z transakcji 1 z poziomem izolacji SERIALIZABLE to nasza transakcja przeskadzająca nie jest w stanie „zaśmiecić” bazy danych dodając jedynie część informacji, gdyż transakcja 1 ma do niej wyłączny dostęp a transakcja przeszkadzająca w ogóle się nie wykonuje.

* + 1. Odbycie przez pacjenta wizyty wraz z wykonanym zabiegiem na zębie pacjenta, składającym się z jednej usługi i zmieniającym stan zęba

W zadaniu najpierw zwiększany jest licznik wizyt klienta, dodawana jest nowa wizyta oraz uzupełniane są informacje o niej. Następnie termin na który był zapisany klient przechodzi w wizytę (w terminie jest zapisywana referencja do odbywanej wizyty). Na koniec trzeba dodać nowy wykonany zabieg do bazy wraz z referencją do usługi z której składał się zabieg oraz zwiększyć licznik sprzedanej usługi. Ostatecznie, jeżeli zmienił się stan zęba informacja o nim jest update’owana. Transakcja musi mieć wysoką wydajność, gdyż będzie wykonywana stosunkowo często – za każdą wizytą klienta. Proponowany poziom izolacji transakcji to poziom domyślny w wielu systemach zarządzania bazami danych – **READ COMMITTED.** Daje całkiem niezłe zabezpieczenie przed błędami, poza tym nie ogranicza współbieżnego dostępu do bazy. Potrzebny jest taki poziom izolacji, gdyż nie możemy sobie pozwolić na *dirty read*, wyobraźmy sobie sytuację w której inna transakcja dodaje nową wizytę, my natomiast stosując niższy poziom izolacji transakcji moglibyśmy jej nadpisać tę wartość i w niej pojawiłby się błędny odczyt później.

* + 1. Rezerwacja przez klienta terminu w którym pracuje dentysta i który jest wolny

Potrzebujemy sprawdzić najpierw czy w proponowanym terminie pracuje jakiś dentysta. Robimy to przeszukując tabelę ***Dni\_godziny\_przyjec***. Następnie jeżeli tak jest, sprawdzamy czy proponowany przez klienta termin jest terminem wolnym – jeżeli tak to możemy zapisać klienta na termin oraz przypisać do tego terminu dentystę (lub jednego z dentystów, jeżeli obaj pracują i są wolni w tym terminie). Aby to zrobić przeglądamy tabelę ***Termin***, przeglądamy jednak jedynie te które jeszcze referencję do wizyty mają równą **NULL** (jeszcze się nie odbyły) i porównujemy z terminem który chcemy zarezerwować – jeżeli istnieje termin zachodzący na termin który chcemy zarezerwować to odpowiedź brzmi nie. Jeżeli na którymkolwiek etapie otrzymamy odpowiedź nie – to należy wykonać rollback. Jeśli w ciągu całej transakcji nie trzymamy odpowiedzi nie – to termin rezerwujemy – dodajemy nowy z zadaną datą i *Id\_wizyty* równym **NULL**. W tej transakcji jest dla nas najważniejsza aktualność informacji (rezerwacje mogą być składane bardzo dynamicznie, a nie chcemy zapisać pacjenta na termin już zarezerwowany) oraz wydajność gdyż ta transakcja będzie wykonywana bardzo często. Proponowanym poziomem izolacji transakcji jest więc **READ UNCOMMITTED**.

* 1. Kod Transakcji

/\* Transakcja 1 \*/

BEGIN TRANSACTION

SET IMPLICIT\_TRANSACTIONS OFF;

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

/\* Dodanie nowej osoby... \*/

INSERT INTO Osoba VALUES

(92050907888, 'Athena', 'Hristanas', '1992-05-09', 'K', 'athena@fizyka.dk', '607-651-300', '80-404 Gdańsk, ul. Dzielna 11' );

/\* ...która jest klientem... \*/

INSERT INTO Klient

VALUES

(92050907888, 'student',0);

/\* ...ma kartę pacjenta... \*/

INSERT INTO Karta\_pacjenta

VALUES

(6,NULL,NULL,NULL,NULL,NULL,92050907888);

/\* ...i zęby \*/

INSERT INTO Zab

VALUES

(161,'tak',NULL,NULL,NULL,'szczęka','prawo','jedynka',6,92050907888),

(162,'tak',NULL,NULL,NULL,'szczęka','prawo','dwójka',6,92050907888),

(163,'tak',NULL,NULL,NULL,'szczęka','prawo','trójka',6,92050907888),

(164,'tak',NULL,NULL,NULL,'szczęka','prawo','czwórka',6,92050907888),

(165,'tak',NULL,NULL,NULL,'szczęka','prawo','piątka',6,92050907888),

(166,'tak',NULL,NULL,NULL,'szczęka','prawo','szóstka',6,92050907888),

(167,'tak',NULL,NULL,NULL,'szczęka','prawo','siódemka',6,92050907888),

(168,'tak',NULL,NULL,NULL,'szczęka','prawo','ósemka',6,92050907888),

(169,'tak',NULL,NULL,NULL,'szczęka','lewo','jedynka',6,92050907888),

(170,'tak',NULL,NULL,NULL,'szczęka','lewo','dwójka',6,92050907888),

(171,'tak',NULL,NULL,NULL,'szczęka','lewo','trójka',6,92050907888),

(172,'tak',NULL,NULL,NULL,'szczęka','lewo','czwórka',6,92050907888),

(173,'tak',NULL,NULL,NULL,'szczęka','lewo','piątka',6,92050907888),

(174,'tak',NULL,NULL,NULL,'szczęka','lewo','szóstka',6,92050907888),

(175,'tak',NULL,NULL,NULL,'szczęka','lewo','siódemka',6,92050907888),

(176,'tak',NULL,NULL,NULL,'szczęka','lewo','ósemka',6,92050907888)

(177,'tak',NULL,NULL,NULL,'żuchwa','prawo','jedynka',6,92050907888),

(178,'tak',NULL,NULL,NULL,'żuchwa','prawo','dwójka',6,92050907888),

(179,'tak',NULL,NULL,NULL,'żuchwa','prawo','trójka',6,92050907888),

(180,'tak',NULL,NULL,NULL,'żuchwa','prawo','czwórka',6,92050907888),

(181,'tak',NULL,NULL,NULL,'żuchwa','prawo','piątka',6,92050907888),

(182,'tak',NULL,NULL,NULL,'żuchwa','prawo','szóstka',6,92050907888),

(183,'tak',NULL,NULL,NULL,'żuchwa','prawo','siódemka',6,92050907888),

(184,'tak',NULL,NULL,NULL,'żuchwa','prawo','ósemka',6,92050907888),

(185,'tak',NULL,NULL,NULL,'żuchwa','lewo','jedynka',6,92050907888),

(186,'tak',NULL,NULL,NULL,'żuchwa','lewo','dwójka',6,92050907888),

(187,'tak',NULL,NULL,NULL,'żuchwa','lewo','trójka',6,92050907888),

(188,'tak',NULL,NULL,NULL,'żuchwa','lewo','czwórka',6,92050907888),

(189,'tak',NULL,NULL,NULL,'żuchwa','lewo','piątka',6,92050907888),

(190,'tak',NULL,NULL,NULL,'żuchwa','lewo','szóstka',6,92050907888),

(191,'tak',NULL,NULL,NULL,'żuchwa','lewo','siódemka',6,92050907888),

(192,'tak',NULL,NULL,NULL,'żuchwa','lewo','ósemka',6,92050907888);

/\* ostatnio nas odwiedziła, trzeba update'ować pewne informacje \*/

UPDATE Klient SET Licznik\_wizyt=Licznik\_wizyt+1 WHERE PESEL\_klient = 92050907888;

UPDATE Karta\_pacjenta SET data\_ostatniej\_wizyty='2013-01-21'

WHERE ID\_karty\_pacjenta = 6;

UPDATE Karta\_pacjenta SET data\_ostatniego\_przegladu='2013-01-21'

WHERE ID\_karty\_pacjenta = 6;

WAITFOR DELAY '00:00:10';

/\* Pokazanie wszystkich osób, klientów, kart pacjentów \*/

SELECT \* FROM Osoba;

SELECT \* FROM Klient;

SELECT \* FROM Karta\_pacjenta;

COMMIT TRANSACTION

/\* Transakcja 1 - bez zabezpieczeń \*/

BEGIN TRANSACTION

/\* Dodanie nowej osoby... \*/

INSERT INTO Osoba VALUES

(92050907888, 'Athena', 'Hristanas', '1992-05-09', 'K', 'athena@fizyka.dk', '607-651-300', '80-404 Gdańsk, ul. Dzielna 11' );

/\* ...która jest klientem... \*/

INSERT INTO Klient

VALUES

(92050907888, 'student',0);

/\* ...ma kartę pacjenta... \*/

INSERT INTO Karta\_pacjenta

VALUES

(6,NULL,NULL,NULL,NULL,NULL,92050907888);

/\* ...i zęby \*/

INSERT INTO Zab

VALUES

(161,'tak',NULL,NULL,NULL,'szczęka','prawo','jedynka',6,92050907888),

(162,'tak',NULL,NULL,NULL,'szczęka','prawo','dwójka',6,92050907888),

(163,'tak',NULL,NULL,NULL,'szczęka','prawo','trójka',6,92050907888),

(164,'tak',NULL,NULL,NULL,'szczęka','prawo','czwórka',6,92050907888),

(165,'tak',NULL,NULL,NULL,'szczęka','prawo','piątka',6,92050907888),

(166,'tak',NULL,NULL,NULL,'szczęka','prawo','szóstka',6,92050907888),

(167,'tak',NULL,NULL,NULL,'szczęka','prawo','siódemka',6,92050907888),

(168,'tak',NULL,NULL,NULL,'szczęka','prawo','ósemka',6,92050907888),

(169,'tak',NULL,NULL,NULL,'szczęka','lewo','jedynka',6,92050907888),

(170,'tak',NULL,NULL,NULL,'szczęka','lewo','dwójka',6,92050907888),

(171,'tak',NULL,NULL,NULL,'szczęka','lewo','trójka',6,92050907888),

(172,'tak',NULL,NULL,NULL,'szczęka','lewo','czwórka',6,92050907888),

(173,'tak',NULL,NULL,NULL,'szczęka','lewo','piątka',6,92050907888),

(174,'tak',NULL,NULL,NULL,'szczęka','lewo','szóstka',6,92050907888),

(175,'tak',NULL,NULL,NULL,'szczęka','lewo','siódemka',6,92050907888),

(176,'tak',NULL,NULL,NULL,'szczęka','lewo','ósemka',6,92050907888)

(177,'tak',NULL,NULL,NULL,'żuchwa','prawo','jedynka',6,92050907888),

(178,'tak',NULL,NULL,NULL,'żuchwa','prawo','dwójka',6,92050907888),

(179,'tak',NULL,NULL,NULL,'żuchwa','prawo','trójka',6,92050907888),

(180,'tak',NULL,NULL,NULL,'żuchwa','prawo','czwórka',6,92050907888),

(181,'tak',NULL,NULL,NULL,'żuchwa','prawo','piątka',6,92050907888),

(182,'tak',NULL,NULL,NULL,'żuchwa','prawo','szóstka',6,92050907888),

(183,'tak',NULL,NULL,NULL,'żuchwa','prawo','siódemka',6,92050907888),

(184,'tak',NULL,NULL,NULL,'żuchwa','prawo','ósemka',6,92050907888),

(185,'tak',NULL,NULL,NULL,'żuchwa','lewo','jedynka',6,92050907888),

(186,'tak',NULL,NULL,NULL,'żuchwa','lewo','dwójka',6,92050907888),

(187,'tak',NULL,NULL,NULL,'żuchwa','lewo','trójka',6,92050907888),

(188,'tak',NULL,NULL,NULL,'żuchwa','lewo','czwórka',6,92050907888),

(189,'tak',NULL,NULL,NULL,'żuchwa','lewo','piątka',6,92050907888),

(190,'tak',NULL,NULL,NULL,'żuchwa','lewo','szóstka',6,92050907888),

(191,'tak',NULL,NULL,NULL,'żuchwa','lewo','siódemka',6,92050907888),

(192,'tak',NULL,NULL,NULL,'żuchwa','lewo','ósemka',6,92050907888);

/\* ostatnio nas odwiedziła, trzeba update'ować pewne informacje \*/

UPDATE Klient SET Licznik\_wizyt=Licznik\_wizyt+1 WHERE PESEL\_klient = 92050907888;

UPDATE Karta\_pacjenta SET data\_ostatniej\_wizyty='2013-01-21'

WHERE ID\_karty\_pacjenta = 6;

UPDATE Karta\_pacjenta SET data\_ostatniego\_przegladu='2013-01-21'

WHERE ID\_karty\_pacjenta = 6;

WAITFOR DELAY '00:00:10';

/\* Pokazanie wszystkich osób, klientów, kart pacjentów \*/

SELECT \* FROM Osoba;

SELECT \* FROM Klient;

SELECT \* FROM Karta\_pacjenta;

COMMIT TRANSACTION

/\* Transakcja przeszkadzająca transakcji 1 \*/

BEGIN TRANSACTION

/\* Dodanie nowej osoby... \*/

INSERT INTO Osoba VALUES

(91021407790, 'Paweł', 'Troka', '1991-02-14', 'M', 'ptroka@fizyka.dk', '725-656-424', '80-404 Gdańsk, ul. Dzielna 11' );

/\* ...która jest klientem... \*/

INSERT INTO Klient

VALUES

(91021407790, 'student',0);

/\* ...ma kartę pacjenta... \*/

INSERT INTO Karta\_pacjenta

VALUES

(6,NULL,NULL,NULL,NULL,NULL,91021407790); /\* tutaj jest tak naprawdę kluczowa sprawa dotycząca faktu przeszkadzania tej transakcji transakcji 1, otóż dodajemy numer karty pacjenta identyczny z numerem karty pacjenta w transakcji 1, jeżeli transakcja 1 byłaby dobrze zabezpieczona - powinna ta operacja (de facto nadpisująca operację z transakcji 1) się nie wykonać - powinien pojawić się błąd iż Karta\_pacjenta o takim ID już istnieje i powinien być wykonany rollback - transakcja przeszkadzająca powinna się nie wykonać \*/

/\* ...i zęby \*/

INSERT INTO Zab

VALUES

(161,'tak',NULL,NULL,NULL,'szczęka','prawo','jedynka',6,91021407790),

(162,'tak',NULL,NULL,NULL,'szczęka','prawo','dwójka',6,91021407790),

(163,'tak',NULL,NULL,NULL,'szczęka','prawo','trójka',6,91021407790),

(164,'tak',NULL,NULL,NULL,'szczęka','prawo','czwórka',6,91021407790),

(165,'tak',NULL,NULL,NULL,'szczęka','prawo','piątka',6,91021407790),

(166,'tak',NULL,NULL,NULL,'szczęka','prawo','szóstka',6,91021407790),

(167,'tak',NULL,NULL,NULL,'szczęka','prawo','siódemka',6,91021407790),

(168,'tak',NULL,NULL,NULL,'szczęka','prawo','ósemka',6,91021407790),

(169,'tak',NULL,NULL,NULL,'szczęka','lewo','jedynka',6,91021407790),

(170,'tak',NULL,NULL,NULL,'szczęka','lewo','dwójka',6,91021407790),

(171,'tak',NULL,NULL,NULL,'szczęka','lewo','trójka',6,91021407790),

(172,'tak',NULL,NULL,NULL,'szczęka','lewo','czwórka',6,91021407790),

(173,'tak',NULL,NULL,NULL,'szczęka','lewo','piątka',6,91021407790),

(174,'tak',NULL,NULL,NULL,'szczęka','lewo','szóstka',6,91021407790),

(175,'tak',NULL,NULL,NULL,'szczęka','lewo','siódemka',6,91021407790),

(176,'tak',NULL,NULL,NULL,'szczęka','lewo','ósemka',6,91021407790)

(177,'tak',NULL,NULL,NULL,'żuchwa','prawo','jedynka',6,91021407790),

(178,'tak',NULL,NULL,NULL,'żuchwa','prawo','dwójka',6,91021407790),

(179,'tak',NULL,NULL,NULL,'żuchwa','prawo','trójka',6,91021407790),

(180,'tak',NULL,NULL,NULL,'żuchwa','prawo','czwórka',6,91021407790),

(181,'tak',NULL,NULL,NULL,'żuchwa','prawo','piątka',6,91021407790),

(182,'tak',NULL,NULL,NULL,'żuchwa','prawo','szóstka',6,91021407790),

(183,'tak',NULL,NULL,NULL,'żuchwa','prawo','siódemka',6,91021407790),

(184,'tak',NULL,NULL,NULL,'żuchwa','prawo','ósemka',6,91021407790),

(185,'tak',NULL,NULL,NULL,'żuchwa','lewo','jedynka',6,91021407790),

(186,'tak',NULL,NULL,NULL,'żuchwa','lewo','dwójka',6,91021407790),

(187,'tak',NULL,NULL,NULL,'żuchwa','lewo','trójka',6,91021407790),

(188,'tak',NULL,NULL,NULL,'żuchwa','lewo','czwórka',6,91021407790),

(189,'tak',NULL,NULL,NULL,'żuchwa','lewo','piątka',6,91021407790),

(190,'tak',NULL,NULL,NULL,'żuchwa','lewo','szóstka',6,91021407790),

(191,'tak',NULL,NULL,NULL,'żuchwa','lewo','siódemka',6,91021407790),

(192,'tak',NULL,NULL,NULL,'żuchwa','lewo','ósemka',6,91021407790);

/\* ostatnio nas odwiedził, trzeba update'ować pewne informacje \*/

UPDATE Klient SET Licznik\_wizyt=Licznik\_wizyt+1 WHERE PESEL\_klient = 91021407790;

UPDATE Karta\_pacjenta SET data\_ostatniej\_wizyty='2013-01-21'

WHERE ID\_karty\_pacjenta = 6;

UPDATE Karta\_pacjenta SET data\_ostatniego\_przegladu='2013-01-21'

WHERE ID\_karty\_pacjenta = 6;

COMMIT TRANSACTION

/\* Transakcja 2 \*/

BEGIN TRANSACTION

SET IMPLICIT\_TRANSACTIONS OFF;

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

/\* odwiedził nas nasz stały klient, zwiększamy licznik wizyt \*/

UPDATE Klient SET Licznik\_wizyt=Licznik\_wizyt+1 WHERE PESEL\_klient = 67060182700;

INSERT INTO Wizyta /\* dodawana jest nowa wizyta \*/

VALUES

(15,'2012-02-08',105,62091635004);

UPDATE Termin SET Id\_wizyty = 15 WHERE Id\_klient = 67060182700;

INSERT INTO Wykonany\_zabieg /\* składający się z jednej usługi \*/

VALUES

(2,'Wyrywanie trójki','zabieg przebiegł poprawnie',120,15,88121241222,8);/\* wizyta, dentysta, uslugi \*/

/\*zwiększamy licznik sprzedanej usługi\*/

UPDATE Uslugi SET Liczba\_sprzedanych=Liczba\_sprzedanych+1 WHERE ID\_uslugi = 8;

/\* relacja między wykonanym zabiegiem a zębem \*/

INSERT INTO Wykonany\_zabieg\_na\_zebie

VALUES

(11,2);

UPDATE Zab SET stan\_zeba = 'wyrwany' WHERE ID\_zeba = 11;

UPDATE Zab SET co\_jeszcze\_do\_zrobienia = 'wstawić implant' WHERE ID\_zeba = 11;

WAITFOR DELAY '00:00:10';

/\* Pokazanie wszystkich klientów, wizyt, terminów, zębów, wykonanych zabiegów \*/

SELECT \* FROM Klient;

SELECT \* FROM Wizyta;

SELECT \* FROM Termin;

SELECT \* FROM Zab;

SELECT \* FROM Wykonany\_zabieg;

COMMIT TRANSACTION

/\* Transakcja 2 – bez zabezpieczeń \*/

BEGIN TRANSACTION

/\* odwiedził nas nasz stały klient, zwiększamy licznik wizyt \*/

UPDATE Klient SET Licznik\_wizyt=Licznik\_wizyt+1 WHERE PESEL\_klient = 67060182700;

INSERT INTO Wizyta /\* dodawana jest nowa wizyta \*/

VALUES

(15,'2012-02-08',105,62091635004);

UPDATE Termin SET Id\_wizyty = 15 WHERE Id\_klient = 67060182700;

INSERT INTO Wykonany\_zabieg /\* składający się z jednej usługi \*/

VALUES

(2,'Wyrywanie trójki','zabieg przebiegł poprawnie',120,15,88121241222,8);/\* wizyta, dentysta, uslugi \*/

/\*zwiększamy licznik sprzedanej usługi\*/

UPDATE Uslugi SET Liczba\_sprzedanych=Liczba\_sprzedanych+1 WHERE ID\_uslugi = 8;

/\* relacja między wykonanym zabiegiem a zębem \*/

INSERT INTO Wykonany\_zabieg\_na\_zebie

VALUES

(11,2);

UPDATE Zab SET stan\_zeba = 'wyrwany' WHERE ID\_zeba = 11;

UPDATE Zab SET co\_jeszcze\_do\_zrobienia = 'wstawić implant' WHERE ID\_zeba = 11;

WAITFOR DELAY '00:00:10';

/\* Pokazanie wszystkich klientów, wizyt, terminów, zębów, wykonanych zabiegów \*/

SELECT \* FROM Klient;

SELECT \* FROM Wizyta;

SELECT \* FROM Termin;

SELECT \* FROM Zab;

SELECT \* FROM Wykonany\_zabieg;

COMMIT TRANSACTION