

Bazy danych 1

Projekt – serwis znany lekarz – etap 2

Mateusz Hryciów, nr 283365

1. Zadanie do zrealizowania

Celem drugiego etapu projektu było stworzenie funkcjonalnej bazy danych o temacie „serwis znany lekarz”.

2. Zmiany względem koncepcji w pierwszym etapie

Stworzona w pierwszym etapie koncepcja bazy danych była w dużej mierze uproszczona. Z tego powodu zdecydowano o wprowadzeniu dodatkowych tabel, które będą przechowywały opisane niżej informacje:

- SPECJALIZACJA – powiązanie specjalizacji z jej id, aby nie przechowywać w tabeli LEKARZE wielokrotnie tej samej informacji,
- PLACOWKA – podstawowe informacje o adresie placówek, w których przyjmują lekarze,
- ZATRUDNIENIE – powiązanie lekarzy z placówkami, w których pracują,
- CHOROBY – powiązanie choroby z jej id, aby nie przechowywać w tabeli PACJENCI wielokrotnie tej samej informacji,
- CHORZY – powiązanie pacjentów z obecnymi u nich chorobami.

Dodatkowo wprowadzono pewne modyfikacje dotyczące funkcjonalności.

- Procedury

Zachowano procedurę pozwalającą na zapisywanie przez pacjenta na badanie do wybranego lekarza. Przy czym występuje ograniczenie, że każdy lekarz może przeprowadzić w ciągu jednego dnia maksymalnie 5 badań.

Ze względu na zbyt dużą prostotę zrezygnowano z procedury pozwalającej na dodanie oceny oraz wyświetlanie opinii o wybranym lekarzu.

W zamian zaproponowano dwie procedury, które w pierwszym etapie miały być funkcjami. Procedury te pobierają na wejściu id specjalizacji lekarza oraz liczbę lekarzy, która będzie wyświetlona. Procedury te wybierają zadaną ilość najlepiej ocenianych lekarzy lub tych o najniższym koszcie wizyty. Następnie przy pomocy kursorów wyświetlają na ekranie wybrane informacje o lekarzach i specjalizacji.

- Funkcje

Jak wcześniej wspomniano, koncepcje funkcji z pierwszego etapu zostały całkowicie porzucone. Zdecydowano się na stworzenie całkiem nowych funkcji.

Pierwsza z nich zwraca najlepszą placówkę, czyli taką dla której średnia ocena wszystkich pracowników jest najwyższa. Zadaniem funkcji jest zwrócenie takiej placówki, w której jak najwięcej lekarzy jest dobrze ocenianych. Zatem przyjęto, że kryterium wyboru nie będzie średnia ze wszystkich opinii dotyczących lekarzy pracujących w danym szpitalu, lecz średnia

z średnich ocen każdego z nich. W funkcji tej zaproponowano użycie złączeń tabel w celu uzyskania pożądanych informacji.

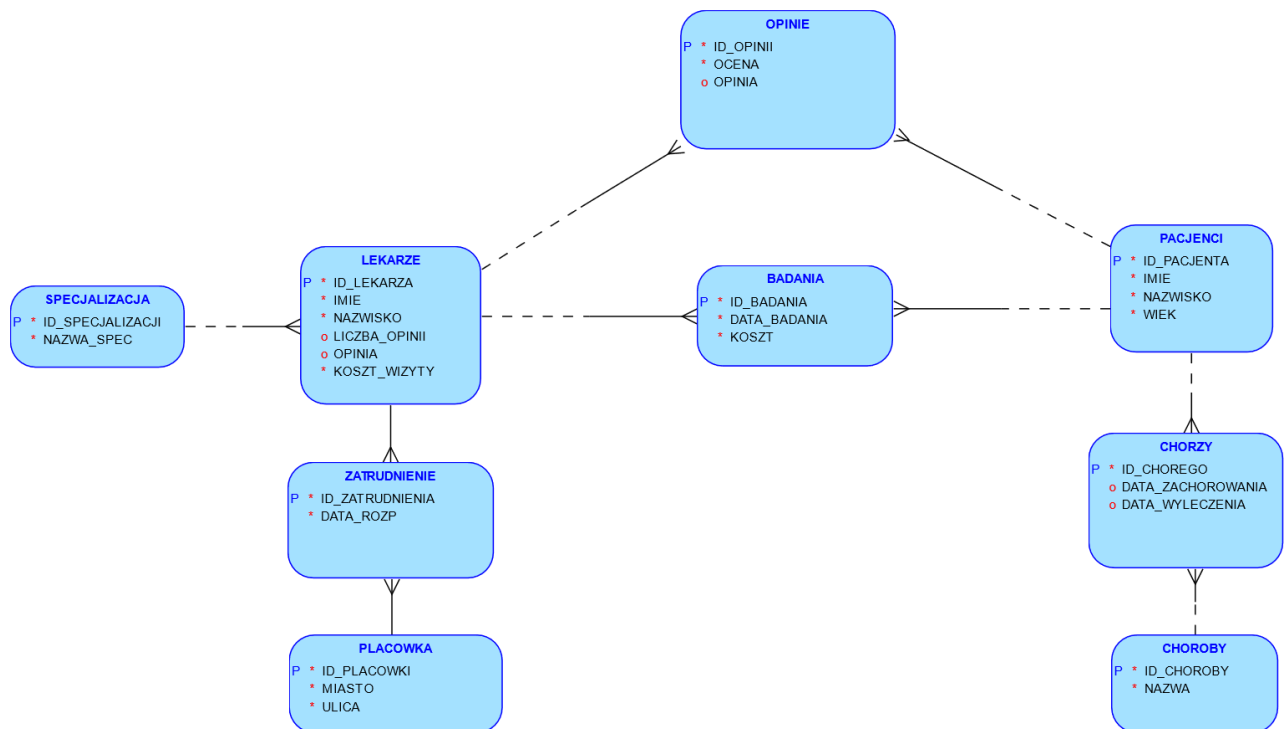
Zadaniem drugiej funkcji było zwracanie najczęstszej choroby występujących u ludzi w danym wieku. Przyjmuje ona na wejściu pewną liczbę reprezentującą wiek. Następnie wykorzystywane jest złączenie kilku tabel w celu zliczenia ile razy występuje choroba o danym id. Uwzględniane są tylko osoby, które przeszły jakiekolwiek badanie w ciągu ostatniego roku. Dodatkowo założono, że zliczane są wszystkie wystąpienia choroby w zakresie ± 10 lat względem parametru wejściowego.

- Wyzwalacze

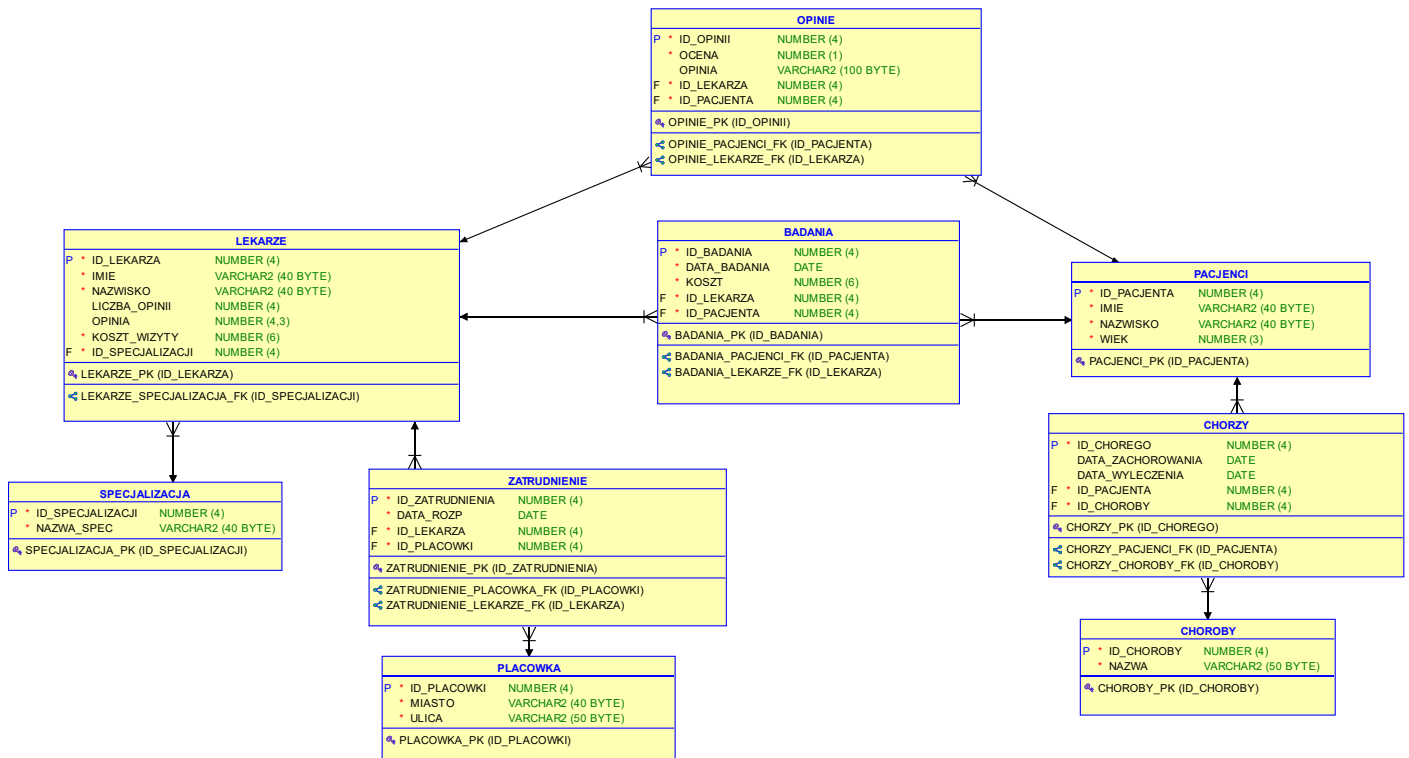
Koncepcja dwóch przewidzianych wyzwalaczy pozostała bez zmian. Pierwszy będzie aktualizował dane w tabeli LEKARZE dotyczące ich oceny po dodaniu nowej opinii. Natomiast drugi będzie udzielał zniżek na badanie jeżeli dany pacjent w ciągu ostatnich 2 lat miał wizytę u wybranego lekarza.

3. Modele

Jak wcześniej wspomniano dokonano pewnych modyfikacji względem koncepcji w pierwszym etapie, m. in. dodano nowe tabele oraz powiązano tabelę OPINIE z tabelą PACJENCI. Z tego względu należało stworzyć nowy diagram ER:



Po jego przekształceniu uzyskano diagram z modelem relacyjnym:



Na podstawie stworzonego modelu automatycznie wygenerowano skrypt DDL służący do stworzenia tabel wraz z ograniczeniami. Umieszczony on został w **Załączniku 1**.

Stworzone tabele należy wypełnić danymi. W tym celu należy wykorzystać skrypt znajdujący się w **Załączniku 2**.

4. Testowanie funkcjonalności stworzonej bazy danych

Testowanie zaproponowanej bazy danych odbędzie się w dwóch krokach. W pierwszym zaprezentowane zostanie działanie zdefiniowanych procedur, funkcji i wyzwalaczy. Natomiast w drugim przedstawione będą różne zapytania. W dalszej części pracy będą przedstawiane wycinki kodu w celu sprawniejszego omówienia funkcjonalności. Pełny skrypt testujący dołączono w **Załączniku 3**.

W pierwszym kroku przetestowano działanie procedury *zapisz_badanie*. Wymaga ona podania na wejściu *id_lekarza*, *id_pacjenta* oraz daty badania. Procedura powinna uniemożliwić zapisanie się na badanie jeżeli wybrany lekarz ma danego dnia umówione już 5 wizyt. W celu przetestowania wykorzystano skrypt:

```

BEGIN
znany_lekarz.zapisz_badanie(108,102,'20/07/13');
znany_lekarz.zapisz_badanie(108,103,'20/07/13');
znany_lekarz.zapisz_badanie(108,104,'20/07/13');
znany_lekarz.zapisz_badanie(108,105,'20/07/13');
znany_lekarz.zapisz_badanie(108,106,'20/07/13');
znany_lekarz.zapisz_badanie(108,107,'20/07/13');

znany_lekarz.zapisz_badanie(103,109,'20/03/13');
znany_lekarz.zapisz_badanie(103,109,'20/07/13');

znany_lekarz.zapisz_badanie(1030,110,'20/03/13');
znany_lekarz.zapisz_badanie(103,1100,'20/03/13');
END;
  
```

Pierwsze 6 wywołań procedury to zapisy różnych pacjentów do tego samego lekarza na ten sam dzień. Poniżej przedstawiono otrzymaną informację:

```
Pierwsza wizyta pacjenta 102
Pierwsza wizyta pacjenta 103
Pierwsza wizyta pacjenta 104
Pierwsza wizyta pacjenta 105
Pierwsza wizyta pacjenta 106
Lekarz 108 ma juz umowione 5 wizyt 20/07/13
```

Zatem zgodnie z oczekiwaniami próba zapisu szóstej osoby zakończyła się niepowodzeniem. Dodatkowo wyświetlona została informacja o pierwszej wizycie pacjentów u danego lekarza.

W dalszej części skryptu wprowadzono do bazy dwa badania tego samego pacjenta u tego samego lekarza. Otrzymano informację:

```
Pierwsza wizyta pacjenta 109
Pacjentowi 109 przyznano zniżkę w wysokości 11
```

Odstęp między wizytami był mniejszy niż 2 lata, a więc pacjentowi należała się zniżka, która została przydzielona automatycznie. Uwidocznilo się tu zatem działanie wyzwalacza *znizka_badanie*.

Procedura dodatkowo informuje jeżeli osoba wprowadziła błędne dane:

```
Sprawdz ponownie ID pracownika lub ID lekarza
Sprawdz ponownie ID pracownika lub ID lekarza
```

Po zastosowaniu skryptu, tabela PRACOWNICY wygląda następująco:

	ID_BADANIA	DATA_BADANIA	KOSZT	ID_LEKARZA	ID_PACJENTA
21	121	15/07/07	100	110	104
22	122	16/03/08	100	101	107
23	123	14/12/13	250	102	104
24	124	20/07/13	120	108	102
25	125	20/07/13	120	108	103
26	126	20/07/13	120	108	104
27	127	20/07/13	120	108	105
28	128	20/07/13	120	108	106
29	129	20/03/13	99	103	110
30	130	20/07/13	99	103	110
31	131	20/03/13	110	103	109
32	132	20/07/13	99	103	109

Widoczne jest zatem, że wpisywany koszt wizyty jest obniżony oraz, że nie doszło do zapisania szóstego pacjenta na ten sam dzień do lekarza o id 108.

W drugim kroku przetestowano procedurę *najtańsi_lekarze* za pomocą skryptu:

```
BEGIN
znany_lekarz.najtansilekarze(103,4);
znany_lekarz.najtansilekarze(103,2);
znany_lekarz.najtansilekarze(1040,2);
END;
```

W przypadku pierwszego wywołania otrzymano odpowiedź:

```
Najtansi lekarze na specjalizacji Pediatra
ID Lekarza: 106, imie i nazwisko: Piotr Chomicki, koszt wizyty: 300
ID Lekarza: 117, imie i nazwisko: Pawel Puc, koszt wizyty: 300
ID Lekarza: 120, imie i nazwisko: Mateusz Sakowicz, koszt wizyty: 390
```

Widoczne jest zatem, że otrzymano tylko 3 propozycje mimo faktu, że zażądano 4. Jest to spowodowane faktem, że nie ma więcej pediatrów w bazie danych. Dodatkowo można zaobserwować, że zaproponowani lekarze podani są w kolejności od najtańszego do najdroższego. W przypadku drugiego wywołania otrzymano odpowiedź:

```
Najtansi lekarze na specjalizacji Pediatra
ID Lekarza: 106, imie i nazwisko: Piotr Chomicki, koszt wizyty: 300
ID Lekarza: 117, imie i nazwisko: Pawel Puc, koszt wizyty: 300
```

Tym razem liczba poleconych lekarzy wynosiła zgodnie z oczekiwaniem 2. Procedura jest odporna na wprowadzenie niepoprawnych danych.

Sprawdz ponownie wprowadzone dane

Następna procedura – *najlepsi_lekarze* – działa podobnie do tej przedstawionej poprzednio. Tym razem wyświetlana jest informacja o zadanej liczbie lekarzy, którzy posiadają najlepszą średnią ocenę. Jednakże początkowo żaden lekarz nie jest oceniony.

	ID_LEKARZA	IMIE	NAZWISKO	LICZBA_OPINII	OPINIA	KOSZT_WIZYTY	ID_SPECJALIZACJI
1	101	Bartosz	Mak	(null)	(null)	150	101
2	102	Michał	Konik	(null)	(null)	200	105
3	103	Michał	Szyska	(null)	(null)	110	111
4	104	Agata	Kuligowska	(null)	(null)	180	107
5	105	Bogdan	Lipin	(null)	(null)	600	112
6	106	Piotr	Chomicki	(null)	(null)	300	103
7	107	Monika	Markiewicz	(null)	(null)	500	101
8	108	Kacper	Laskowski	(null)	(null)	120	109
9	109	Damian	Bienert	(null)	(null)	500	107
10	110	Katarzyna	Torzewska	(null)	(null)	1000	110
11	111	Weronika	Karpinska	(null)	(null)	700	105
12	112	Mateusz	Gunia	(null)	(null)	100	105
13	113	Maria	Malicha	(null)	(null)	450	105
14	114	Szymon	Konik	(null)	(null)	90	101
15	115	Rafał	Skrobich	(null)	(null)	700	104
16	116	Monika	Bylicka	(null)	(null)	1300	106
17	117	Paweł	Puc	(null)	(null)	300	103
18	118	Roman	Jakubowicz	(null)	(null)	200	108
19	119	Klaudia	Daroszewska	(null)	(null)	340	111
20	120	Mateusz	Sakowicz	(null)	(null)	390	103

Z tego powodu należy najpierw wprowadzić nowe dane do tabeli OPINIE. Zaobserwowane zostanie wówczas działanie wyzwalacza *ocena_upd*. Wprowadzono następujące opinie:

```
INSERT INTO OPINIE VALUES (101,5,'Bardzo przyjemna wizyta', 101, 103);
INSERT INTO OPINIE VALUES (102,2,null, 101, 101);
INSERT INTO OPINIE VALUES (103,1,'Nie mogło być gorzej', 101, 118);
INSERT INTO OPINIE VALUES (104,5,'Lekarz to gbur', 114, 114);
INSERT INTO OPINIE VALUES (105,3,null, 114, 103);
```

```

INSERT INTO OPINIE VALUES (106,2,'Nie dowiedziałem sie nic nowego', 107, 104);
INSERT INTO OPINIE VALUES (107,5,'Bylo w porzadku', 109, 120);
INSERT INTO OPINIE VALUES (108,3,null, 111, 109);
INSERT INTO OPINIE VALUES (109,4,null, 102, 113);
INSERT INTO OPINIE VALUES (110,7,'Najlepszy lekarz na jakiego trafiam', 118, 106);

```

Po ich wprowadzeniu tabela OPINIE wygląda następująco:

	ID_OPINII	O...	OPINIA	ID_LEKARZA	ID_PACJENTA
1	101	5	Bardzo przyjemna wizyta	101	103
2	102	2 (null)		101	101
3	103	1	Nie moglo byc gorzej	101	118
4	104	5	Lekarz to gbur	114	114
5	105	3 (null)		114	103
6	106	2	Nie dowiedziałem sie nic nowego	107	104
7	107	5	Bylo w porzadku	109	120
8	108	3 (null)		111	109
9	109	4 (null)		102	113

Można zauważyć, że nie udało się dodać opinii o id 110. Jest to spowodowane faktem naruszenia ograniczeń, gdyż jak wcześniej wspomniano oceniać można jedynie w skali 1-5.

Tabela lekarze po wprowadzeniu nowych opinii wygląda następująco:

	ID_LEKA...	IMIE	NAZWISKO	LICZBA_OPINII	OPINIA	KOSZT_WIZYTY	ID_SPECJALIZACJI
1	101	Bartosz	Mak	3	2,667	150	101
2	102	Michal	Konik	1	4	200	105
3	103	Michal	Szyska	(null)	(null)	110	111
4	104	Agata	Kuligowska	(null)	(null)	180	107
5	105	Bogdan	Lipin	(null)	(null)	600	112
6	106	Piotr	Chomicki	(null)	(null)	300	103
7	107	Monika	Markiewicz	1	2	500	101
8	108	Kacper	Laskowski	(null)	(null)	120	109
9	109	Damian	Bienert	1	5	500	107
10	110	Katarzyna	Torzewska	(null)	(null)	1000	110
11	111	Weronika	Karpinska	1	3	700	105
12	112	Mateusz	Gunia	(null)	(null)	100	105
13	113	Maria	Malicha	(null)	(null)	450	105
14	114	Szymon	Konik	2	4	90	101
15	115	Rafal	Skrobich	(null)	(null)	700	104
16	116	Monika	Bylicka	(null)	(null)	1300	106
17	117	Pawel	Puc	(null)	(null)	300	103
18	118	Roman	Jakubowicz	(null)	(null)	200	108
19	119	Klaudia	Daroszewska	(null)	(null)	340	111
20	120	Mateusz	Sakowicz	(null)	(null)	390	103

Widoczne jest zatem, że uaktualniona została liczba opinii oraz ich średnia wartość. Poprawność można zweryfikować na podstawie ocen lekarza o id 101. Otrzymał on trzy oceny: 5, 1, 2. Ich średnia arytmetyczna zgadza się z tą znajdującą się w tabeli i wynosi 2,667.

Posiadając pewne informacje o ocenach lekarzy można przejść do prezentacji działania procedury *najlepsi_lekarze*. Zastosowano skrypt:

```
BEGIN
znany_lekarz.najlepsi_lekarze(101,2);
END;
```

W dalszych procedurach i skryptach również istnieją zabezpieczenie przed niepoprawnymi danymi wejściowymi, jednak nie będzie to już prezentowane. Podobnie jak w przypadku najtańszych lekarzy, tak samo w przypadku najlepiej ocenianych, wyświetlana jest lista w kolejności od najlepszego do najgorszego. Otrzymano:

```
Najlepiej oceniani lekarze na specjalizacji Dermatolog
ID Lekarza: 114, imie i nazwisko: Szymon Konik, srednia ocena: 4
ID Lekarza: 101, imie i nazwisko: Bartosz Mak, srednia ocena: 2,667
```

Następnie przetestowano działanie funkcji *najlepsza_placowka*. Zwraca ona id placówki, w której średnia z ocen dla każdego z lekarzy jest najwyższa. Aby zweryfikować, czy funkcja działa poprawnie wyświetlono wszystkich najlepiej ocenionych lekarzy wraz z ich miejscem pracy. Użyto zapytania:

```
SELECT * FROM LEKARZE L
JOIN ZATRUDNIENIE Z ON z.ID_LEKARZA = l.ID_LEKARZA
ORDER BY OPINIA DESC NULLS LAST;
```

W odpowiedzi otrzymano tabelę:

ID_LEKARZA	IMIE	NAZWISKO	LICZBA_OPINII	OPINIA	KOSZT_WIZYTY	ID_SPECJALIZACJI	ID_ZATRUDNIENIA	DATA_ROZP	ID_LEKARZA_1	ID_PLACOWKI
1	109	Damian	Bienert	1	5	500	107	111 16/12/24	109	109
2	102	Michał	Konik	1	4	200	105	116 12/08/11	102	106
3	111	Weronika	Karpinska	1	3	700	105	105 15/08/01	111	101
4	111	Weronika	Karpinska	1	3	700	105	114 19/02/27	111	104
5	101	Bartosz	Mak	3	2,667	150	101	102 16/03/21	101	102
6	101	Bartosz	Mak	3	2,667	150	101	118 04/01/11	101	106
7	101	Bartosz	Mak	3	2,667	150	101	117 08/07/25	101	107
8	107	Monika	Markiewicz	1	2	500	101	121 15/11/01	107	111
9	107	Monika	Markiewicz	1	2	500	101	110 13/04/12	107	110
10	104	Agata	Kuligowska	(null)	(null)	180	107	115 13/01/26	104	103
11	108	Kacper	Laskowski	(null)	(null)	120	109	119 04/04/10	108	110

Dzięki temu, że liczba danych jest niewielka można przewidzieć, że funkcja jako najlepszą placówkę zwróci tę o id 109. Gdyż jedyny oceniony w niej lekarz posiada opinię 5. Zatem użyto skryptu:

```
DECLARE
v_placowka NUMBER;
BEGIN
v_placowka := znany_lekarz.najlepsza_placowka;
dbms_output.put_line('Placowka w ktorej pracuja najelpsi lekarze posiada id ' || v_placowka);
END;
```

W odpowiedzi otrzymano:

```
Placowka w ktorej pracuja najelpsi lekarze posiada id 109
```

Zatem można stwierdzić, że funkcja działa w sposób poprawny. Następnie przetestowano ostatnią zaimplementowaną funkcję *najczestsza_choroba_wiek*. Pobiera ona na wejściu wiek oraz zwraca nazwę choroby, z którą najczęściej zgłaszali się pacjenci. Przy czym zliczane były

wystąpienia chorób u pacjentów w zakresie ± 10 lat względem podanego parametru. Do przetestowania użyto skryptu:

```
DECLARE
v_choroba CHOROBY.NAZWA%TYPE;
v_choroba2 CHOROBY.NAZWA%TYPE;
BEGIN
v_choroba := znany_lekarz.najczestsza_choroba_wiek(40);
dbms_output.put_line('Najczesciej wystepujaca choroba w ostanim roku dla ludzi w wieku 40+-10 lat jest ' || v_choroba);
v_choroba := znany_lekarz.najczestsza_choroba_wiek(60);
dbms_output.put_line('Najczesciej wystepujaca choroba w ostanim roku dla ludzi w wieku 60+-10 lat jest ' || v_choroba);
END;
```

Otrzymano w odpowiedzi:

```
Najczesciej wystepujaca choroba w ostanim roku dla ludzi w wieku 40+-10 lat jest Zaczma
Najczesciej wystepujaca choroba w ostanim roku dla ludzi w wieku 60+-10 lat jest Rak nerki
```

Zatem można zauważyć, że dla ludzi w różnym wieku dominują inne choroby. W ten sposób zakończono testowanie wyzwalaczy, funkcji i procedur. Drugim etapem przedstawiania funkcjonalności bazy danych jest prezentacja różnych zapytań.

Pierwszym jest zapytanie korzystające ze złączenia tabeli PACJENCI oraz BADANIA. Pyta ono o id wszystkich pacjentów, którzy wydali na badania więcej niż 500. Użyto zapytania:

```
SELECT ID_PACJENTA, WYDATKI
FROM(
SELECT b.ID_PACJENTA, sum(KOSZT) as WYDATKI FROM PACJENCI P
JOIN BADANIA B on b.ID_PACJENTA = p.ID_PACJENTA
GROUP BY b.ID_PACJENTA)
WHERE WYDATKI > 500;
```

W odpowiedzi otrzymano:

	ID_PACJENTA	WYDATKI
1	105	1120
2	104	1850

Następne zapytanie powinno zwrócić imiona, nazwiska oraz koszty wizyty lekarzy, których koszt wizyty jest wyższy niż średni koszt wizyty u stomatologa. Zastosowano zapytanie:

```
SELECT IMIE, NAZWISKO, KOSZT_WIZYT
FROM LEKARZE
WHERE KOSZT_WIZYT > (SELECT avg(KOSZT_WIZYT) FROM LEKARZE
NATURAL JOIN SPECJALIZACJA
WHERE NAZWA_SPEC = 'Stomatolog');
```


W odpowiedzi otrzymano:

	IMIE	NAZWISKO	KOSZT_WIZYTY
1	Bogdan	Lipin	600
2	Monika	Markiewicz	500
3	Damian	Bienert	500
4	Katarzyna	Torzewska	1000
5	Weronika	Karpinska	700
6	Maria	Malicha	450
7	Rafal	Skrobich	700
8	Monika	Bylicka	1300
9	Mateusz	Sakowicz	390

Należy zauważyć, że użyto złączenia NATURAL JOIN, które w sposób automatyczny łączy tabele na podstawie nazw kolumn. Wiedząc że id specjalizacji – stomatolog to 105 można sprawdzić, czy otrzymano poprawne wyniki.

7	115	Rafal	Skrobich	(null)	(null)	700	104
8	102	Michal	Konik	1	4	200	105
9	113	Maria	Malicha	(null)	(null)	450	105
10	112	Mateusz	Gunia	(null)	(null)	100	105
11	111	Weronika	Karpinska	1	3	700	105
12	116	Monika	Bylicka	(null)	(null)	1300	106

Na podstawie fragmentu tabeli LEKARZE można ręcznie obliczyć, że średni koszt wizyty u stomatologa to 362,5. Zatem otrzymane wyniki są poprawne.

Ostatnie prezentowane zapytanie ma na celu otrzymanie odpowiedzi, ile było takich przypadków, że różnica między datą zachorowania oraz datą wyleczenia pacjenta była mniejsza niż rok. W odpowiedzi chcemy otrzymać nazwy chorób, dla których zaszła taka sytuacja i liczbę wystąpień.

```
SELECT NAZWA, count(NAZWA) as LICZBA_WYLECZEN FROM CHORZY
NATURAL JOIN CHOROBY
WHERE (DATA_WYLECZENIA - DATA_ZACHOROWANIA) < 365
GROUP BY NAZWA
ORDER BY LICZBA_WYLECZEN DESC;
```

W odpowiedzi otrzymano:

	NAZWA	LICZBA_WYLECZEN
1	Białaczka	1
2	Rak piersi	1
3	Ospa	2

5. Wnioski

Analizując przyjęte założenia dotyczące wyzwalaczy, procedur oraz funkcji można stwierdzić, że udało się je zaprojektować zgodnie z zamysłem. Każda z zaimplementowanych metod działa w sposób oczekiwany i jest odporna na wprowadzenie danych, które nie znajdują się w bazie danych. Oczywiście, na pewno istnieją pewne aspekty, których modyfikacja prowadziłaby do optymalizacji działania bazy danych. Jednakże zrealizowane funkcjonalności bazy danych o temacie „**serwis znany lekarz**” można uznać za satysfakcjonujące.

6. Załączniki

Załącznik 1

```
DROP TABLE OPINIE CASCADE CONSTRAINTS;
DROP TABLE LEKARZE CASCADE CONSTRAINTS;
DROP TABLE ZATRUDNIENIE CASCADE CONSTRAINTS;
DROP TABLE PLACOWKA CASCADE CONSTRAINTS;
DROP TABLE BADANIA CASCADE CONSTRAINTS;
DROP TABLE PACJENCI CASCADE CONSTRAINTS;
DROP TABLE CHOROZY CASCADE CONSTRAINTS;
DROP TABLE CHOROBY CASCADE CONSTRAINTS;
DROP TABLE SPECJALIZACJA CASCADE CONSTRAINTS;
```

```
CREATE TABLE badania (
    id_badiana    NUMBER(4) NOT NULL,
    data_badiana  DATE NOT NULL,
    koszt         NUMBER(6) NOT NULL,
    id_lekarza    NUMBER(4) NOT NULL,
    id_pacjenta   NUMBER(4) NOT NULL
);
```

```
ALTER TABLE badania ADD CONSTRAINT badania_pk PRIMARY KEY ( id_badiana );
```

```
CREATE TABLE choroby (
    id_choroby    NUMBER(4) NOT NULL,
    nazwa         VARCHAR2(50 BYTE) NOT NULL
);
```

```
ALTER TABLE choroby ADD CONSTRAINT choroby_pk PRIMARY KEY ( id_choroby );
```

```
CREATE TABLE chorzy (
    id_chorego    NUMBER(4) NOT NULL,
    data_zachorowania  DATE,
    data_wyleczenia   DATE,
    id_pacjenta    NUMBER(4) NOT NULL,
    id_choroby     NUMBER(4) NOT NULL
);
```

```
ALTER TABLE chorzy ADD CONSTRAINT chorzy_pk PRIMARY KEY ( id_chorego );
```

```
CREATE TABLE lekarze (
    id_lekarza    NUMBER(4) NOT NULL,
    imie          VARCHAR2(40 BYTE) NOT NULL,
    nazwisko      VARCHAR2(40 BYTE) NOT NULL,
    liczba_opinii  NUMBER(4),
    opinia         NUMBER(4, 3),
    koszt_wizyty   NUMBER(6) NOT NULL,
    id_specjalizacji  NUMBER(4) NOT NULL
);
```

```
ALTER TABLE lekarze ADD CONSTRAINT lekarze_pk PRIMARY KEY ( id_lekarza );
```

```
CREATE TABLE opinie (
    id_opinii     NUMBER(4) NOT NULL,
    ocena         NUMBER(1) NOT NULL,
```

```
    opinia    VARCHAR2(100 BYTE),
    id_lekarza NUMBER(4) NOT NULL,
    id_pacjenta NUMBER(4) NOT NULL
);
```

```
ALTER TABLE opinie ADD CONSTRAINT opinie_pk PRIMARY KEY ( id_opinii );
```

```
CREATE TABLE pacjenci (
    id_pacjenta NUMBER(4) NOT NULL,
    imie        VARCHAR2(40 BYTE) NOT NULL,
    nazwisko    VARCHAR2(40 BYTE) NOT NULL,
    wiek        NUMBER(3) NOT NULL
);
```

```
ALTER TABLE pacjenci ADD CONSTRAINT pacjenci_pk PRIMARY KEY ( id_pacjenta );
```

```
CREATE TABLE placowka (
    id_placowki NUMBER(4) NOT NULL,
    miasto      VARCHAR2(40 BYTE) NOT NULL,
    ulica       VARCHAR2(50 BYTE) NOT NULL
);
```

```
ALTER TABLE placowka ADD CONSTRAINT placowka_pk PRIMARY KEY ( id_placowki );
```

```
CREATE TABLE specjalizacja (
    id_specjalizacji NUMBER(4) NOT NULL,
    nazwa_spec       VARCHAR2(40 BYTE) NOT NULL
);
```

```
ALTER TABLE specjalizacja ADD CONSTRAINT specjalizacja_pk PRIMARY KEY ( id_specjalizacji );
```

```
CREATE TABLE zatrudnienie (
    id_zatrudnienia NUMBER(4) NOT NULL,
    data_rozp       DATE NOT NULL,
    id_lekarza      NUMBER(4) NOT NULL,
    id_placowki     NUMBER(4) NOT NULL
);
```

```
ALTER TABLE zatrudnienie ADD CONSTRAINT zatrudnienie_pk PRIMARY KEY ( id_zatrudnienia );
```

```
ALTER TABLE badania
    ADD CONSTRAINT badania_lekarze_fk FOREIGN KEY ( id_lekarza )
    REFERENCES lekarze ( id_lekarza );
```

```
ALTER TABLE badania
    ADD CONSTRAINT badania_pacjenci_fk FOREIGN KEY ( id_pacjenta )
    REFERENCES pacjenci ( id_pacjenta );
```

```
ALTER TABLE chorzy
    ADD CONSTRAINT chorzy_choroby_fk FOREIGN KEY ( id_choroby )
    REFERENCES choroby ( id_choroby );
```

```
ALTER TABLE chorzy
    ADD CONSTRAINT chorzy_pacjenci_fk FOREIGN KEY ( id_pacjenta )
    REFERENCES pacjenci ( id_pacjenta );
```

```
ALTER TABLE lekarze
    ADD CONSTRAINT lekarze_specjalizacja_fk FOREIGN KEY ( id_specjalizacji )
    REFERENCES specjalizacja ( id_specjalizacji );
```

```
ALTER TABLE opinie
    ADD CONSTRAINT opinie_lekarze_fk FOREIGN KEY ( id_lekarza )
    REFERENCES lekarze ( id_lekarza );
```

```
ALTER TABLE opinie
    ADD CONSTRAINT opinie_pacjenci_fk FOREIGN KEY ( id_pacjenta )
    REFERENCES pacjenci ( id_pacjenta );
```

```
ALTER TABLE zatrudnienie
    ADD CONSTRAINT zatrudnienie_lekarze_fk FOREIGN KEY ( id_lekarza )
    REFERENCES lekarze ( id_lekarza );
```

```
ALTER TABLE zatrudnienie
ADD CONSTRAINT zatrudnienie_placowka_fk FOREIGN KEY ( id_placowki )
REFERENCES placowka ( id_placowki );
```

```
ALTER TABLE opinie
ADD CONSTRAINT oceny_zakres CHECK (OCENA IN (1, 2, 3, 4, 5));
```

Załącznik 2

```
ALTER SESSION SET NLS_DATE_FORMAT = "yy/mm/dd";
```

```
DELETE FROM OPINIE;
DELETE FROM ZATRUDNIENIE;
DELETE FROM PLACOWKA;
DELETE FROM BADANIA;
DELETE FROM CHORZY;
DELETE FROM CHOROBY;
DELETE FROM LEKARZE;
DELETE FROM SPECJALIZACJA;
DELETE FROM PACJENCI;
```

```
INSERT INTO SPECJALIZACJA VALUES (101, 'Dermatolog');
INSERT INTO SPECJALIZACJA VALUES (102, 'Kardiolog');
INSERT INTO SPECJALIZACJA VALUES (103, 'Pediatria');
INSERT INTO SPECJALIZACJA VALUES (104, 'Psycholog');
INSERT INTO SPECJALIZACJA VALUES (105, 'Stomatolog');
INSERT INTO SPECJALIZACJA VALUES (106, 'Ortopeda');
INSERT INTO SPECJALIZACJA VALUES (107, 'Chirurg');
INSERT INTO SPECJALIZACJA VALUES (108, 'Laryngolog');
INSERT INTO SPECJALIZACJA VALUES (109, 'Neurolog');
INSERT INTO SPECJALIZACJA VALUES (110, 'Onkolog');
INSERT INTO SPECJALIZACJA VALUES (111, 'Ginekolog');
INSERT INTO SPECJALIZACJA VALUES (112, 'Okulista');
INSERT INTO SPECJALIZACJA VALUES (113, 'Internista');
```

```
INSERT INTO LEKARZE VALUES (101, 'Bartosz', 'Mak', null, null, 150, 101);
INSERT INTO LEKARZE VALUES (102, 'Michał', 'Konik', null, null, 200, 105);
INSERT INTO LEKARZE VALUES (103, 'Michał', 'Szyszka', null, null, 110, 111);
INSERT INTO LEKARZE VALUES (104, 'Agata', 'Kuligowska', null, null, 180, 107);
INSERT INTO LEKARZE VALUES (105, 'Bogdan', 'Lipin', null, null, 600, 112);
INSERT INTO LEKARZE VALUES (106, 'Piotr', 'Chomicki', null, null, 300, 103);
INSERT INTO LEKARZE VALUES (107, 'Monika', 'Markiewicz', null, null, 500, 101);
INSERT INTO LEKARZE VALUES (108, 'Kacper', 'Łaskowski', null, null, 120, 109);
INSERT INTO LEKARZE VALUES (109, 'Damian', 'Bienert', null, null, 500, 107);
INSERT INTO LEKARZE VALUES (110, 'Katarzyna', 'Torzewska', null, null, 1000, 110);
INSERT INTO LEKARZE VALUES (111, 'Weronika', 'Karpinska', null, null, 700, 105);
INSERT INTO LEKARZE VALUES (112, 'Mateusz', 'Gunia', null, null, 100, 105);
INSERT INTO LEKARZE VALUES (113, 'Maria', 'Malicha', null, null, 450, 105);
INSERT INTO LEKARZE VALUES (114, 'Szymon', 'Konik', null, null, 90, 101);
INSERT INTO LEKARZE VALUES (115, 'Rafał', 'Skrobich', null, null, 700, 104);
INSERT INTO LEKARZE VALUES (116, 'Monika', 'Bylicka', null, null, 1300, 106);
INSERT INTO LEKARZE VALUES (117, 'Paweł', 'Puc', null, null, 300, 103);
INSERT INTO LEKARZE VALUES (118, 'Roman', 'Jakubowicz', null, null, 200, 108);
INSERT INTO LEKARZE VALUES (119, 'Klaudia', 'Daroszeńska', null, null, 340, 111);
INSERT INTO LEKARZE VALUES (120, 'Mateusz', 'Sakowicz', null, null, 390, 103);
```

```
INSERT INTO PLACOWKA VALUES (101, 'Warszawa', 'Karłowicza 3');
INSERT INTO PLACOWKA VALUES (102, 'Białystok', 'Spółdzielcza 2/8');
INSERT INTO PLACOWKA VALUES (103, 'Warszawa', 'Marszałkowska 48/21');
INSERT INTO PLACOWKA VALUES (104, 'Poznań', 'Głębowa 38');
INSERT INTO PLACOWKA VALUES (105, 'Gdańsk', 'Fenikowskiego 21/1');
INSERT INTO PLACOWKA VALUES (106, 'Gdynia', 'Władysława 9');
INSERT INTO PLACOWKA VALUES (107, 'Warszawa', 'Krawiecka 1/2');
INSERT INTO PLACOWKA VALUES (108, 'Kraków', 'Rogóżńskiego 5/1');
INSERT INTO PLACOWKA VALUES (109, 'Katowice', 'Gdańska 112/104');
INSERT INTO PLACOWKA VALUES (110, 'Rybnik', 'Kamińskiego 103');
INSERT INTO PLACOWKA VALUES (111, 'Warszawa', 'Marcinkowska 103/31');
INSERT INTO PLACOWKA VALUES (112, 'Warszawa', 'Filtrowa 62/12');
INSERT INTO PLACOWKA VALUES (113, 'Kraków', 'Wspólna 54/12');
INSERT INTO PLACOWKA VALUES (114, 'Kalisz', 'Tylca 59C/12');
```

```

INSERT INTO ZATRUDNIENIE VALUES (101,'12/01/12', 103, 101);
INSERT INTO ZATRUDNIENIE VALUES (102,'16/03/21', 101, 102);
INSERT INTO ZATRUDNIENIE VALUES (103,'20/10/09', 104, 105);
INSERT INTO ZATRUDNIENIE VALUES (104,'14/09/04', 106, 106);
INSERT INTO ZATRUDNIENIE VALUES (105,'15/08/01', 111, 101);
INSERT INTO ZATRUDNIENIE VALUES (106,'08/02/01', 120, 105);
INSERT INTO ZATRUDNIENIE VALUES (107,'04/03/06', 118, 110);
INSERT INTO ZATRUDNIENIE VALUES (108,'15/04/07', 119, 112);
INSERT INTO ZATRUDNIENIE VALUES (109,'14/06/11', 115, 114);
INSERT INTO ZATRUDNIENIE VALUES (110,'13/04/12', 107, 110);
INSERT INTO ZATRUDNIENIE VALUES (111,'16/12/24', 109, 109);
INSERT INTO ZATRUDNIENIE VALUES (112,'17/11/17', 108, 106);
INSERT INTO ZATRUDNIENIE VALUES (113,'18/11/18', 110, 107);
INSERT INTO ZATRUDNIENIE VALUES (114,'19/02/27', 111, 104);
INSERT INTO ZATRUDNIENIE VALUES (115,'13/01/26', 104, 103);
INSERT INTO ZATRUDNIENIE VALUES (116,'12/08/11', 102, 106);
INSERT INTO ZATRUDNIENIE VALUES (117,'08/07/25', 101, 107);
INSERT INTO ZATRUDNIENIE VALUES (118,'04/01/11', 101, 106);
INSERT INTO ZATRUDNIENIE VALUES (119,'04/04/10', 108, 110);
INSERT INTO ZATRUDNIENIE VALUES (120,'14/04/14', 104, 112);
INSERT INTO ZATRUDNIENIE VALUES (121,'15/11/01', 107, 111);
INSERT INTO ZATRUDNIENIE VALUES (122,'20/07/01', 113, 105);
INSERT INTO ZATRUDNIENIE VALUES (123,'14/06/05', 115, 103);
INSERT INTO ZATRUDNIENIE VALUES (124,'15/05/06', 118, 103);

```

```

INSERT INTO CHOROBY VALUES (101, 'Cukrzyca');
INSERT INTO CHOROBY VALUES (102, 'Ospa');
INSERT INTO CHOROBY VALUES (103, 'Malaria');
INSERT INTO CHOROBY VALUES (104, 'Grypa');
INSERT INTO CHOROBY VALUES (105, 'Rak jelita grubego');
INSERT INTO CHOROBY VALUES (106, 'Rak nerki');
INSERT INTO CHOROBY VALUES (107, 'Rak piersi');
INSERT INTO CHOROBY VALUES (108, 'Złamanie obojczyka');
INSERT INTO CHOROBY VALUES (109, 'Zwichnięcie kostki');
INSERT INTO CHOROBY VALUES (110, 'Białaczka');
INSERT INTO CHOROBY VALUES (111, 'Torbiel');
INSERT INTO CHOROBY VALUES (112, 'Depresja');
INSERT INTO CHOROBY VALUES (113, 'Zacma');

```

```

INSERT INTO PACJENCI VALUES (101, 'Kamil', 'Kuligowski', 43);
INSERT INTO PACJENCI VALUES (102, 'Karolina', 'Lena', 34);
INSERT INTO PACJENCI VALUES (103, 'Michał', 'Lipin', 21);
INSERT INTO PACJENCI VALUES (104, 'Andrzej', 'Markiewicz', 63);
INSERT INTO PACJENCI VALUES (105, 'Kamila', 'Klich', 81);
INSERT INTO PACJENCI VALUES (106, 'Barbara', 'Zdanowska', 13);
INSERT INTO PACJENCI VALUES (107, 'Kacper', 'Kriszer', 53);
INSERT INTO PACJENCI VALUES (108, 'Radosław', 'Słoka', 57);
INSERT INTO PACJENCI VALUES (109, 'Mirosław', 'Sutkowski', 54);
INSERT INTO PACJENCI VALUES (110, 'Konrad', 'Karbowski', 43);
INSERT INTO PACJENCI VALUES (111, 'Anna', 'Paprzycka', 49);
INSERT INTO PACJENCI VALUES (112, 'Maria', 'Lesniewska', 8);
INSERT INTO PACJENCI VALUES (113, 'Maciej', 'Gleba', 65);
INSERT INTO PACJENCI VALUES (114, 'Kinga', 'Kolczewska', 63);
INSERT INTO PACJENCI VALUES (115, 'Rafał', 'Cybulski', 51);
INSERT INTO PACJENCI VALUES (116, 'Marek', 'Mroczkowski', 23);
INSERT INTO PACJENCI VALUES (117, 'Patryk', 'Bialek', 31);
INSERT INTO PACJENCI VALUES (118, 'Marcin', 'Kmita', 22);
INSERT INTO PACJENCI VALUES (119, 'Agata', 'Kaczorowska', 33);
INSERT INTO PACJENCI VALUES (120, 'Oliwia', 'Sosnowska', 64);
INSERT INTO PACJENCI VALUES (121, 'Gabriela', 'Nowakowska', 68);
INSERT INTO PACJENCI VALUES (122, 'Janusz', 'Krapiec', 65);
INSERT INTO PACJENCI VALUES (123, 'Wiktoria', 'Oziebło', 15);
INSERT INTO PACJENCI VALUES (124, 'Piotr', 'Bitowski', 19);
INSERT INTO PACJENCI VALUES (125, 'Paweł', 'Hojrzak', 65);
INSERT INTO PACJENCI VALUES (126, 'Andrzej', 'Zosik', 37);
INSERT INTO PACJENCI VALUES (127, 'Kinga', 'Jurkiewicz', 43);

```

```
INSERT INTO PACJENCI VALUES (128, 'Roman', 'Dabrowski', 89);
INSERT INTO PACJENCI VALUES (129, 'Pawel', 'Misztal', 99);
INSERT INTO PACJENCI VALUES (130, 'Karolina', 'Mazur', 23);
```

```
INSERT INTO CHORZY VALUES (101, '20/03/03', '20/05/03', 117, 102);
INSERT INTO CHORZY VALUES (102, '20/01/03', null, 123, 101);
INSERT INTO CHORZY VALUES (103, '15/11/05', '16/04/02', 101, 102);
INSERT INTO CHORZY VALUES (104, null, null, 104, 106);
INSERT INTO CHORZY VALUES (105, '19/05/12', '20/08/01', 106, 107);
INSERT INTO CHORZY VALUES (106, '15/05/23', '18/04/21', 111, 101);
INSERT INTO CHORZY VALUES (107, '14/02/17', '15/04/27', 125, 101);
INSERT INTO CHORZY VALUES (108, '15/01/03', null, 126, 102);
INSERT INTO CHORZY VALUES (109, '16/09/03', '18/04/04', 130, 103);
INSERT INTO CHORZY VALUES (110, '20/02/03', '20/03/23', 111, 107);
INSERT INTO CHORZY VALUES (111, '20/05/03', null, 104, 106);
INSERT INTO CHORZY VALUES (112, null, null, 101, 113);
INSERT INTO CHORZY VALUES (113, '11/04/03', '20/04/05', 105, 102);
INSERT INTO CHORZY VALUES (114, null, '12/05/01', 106, 112);
INSERT INTO CHORZY VALUES (115, '09/04/03', null, 115, 111);
INSERT INTO CHORZY VALUES (116, '12/06/03', '12/06/06', 125, 110);
INSERT INTO CHORZY VALUES (117, '09/05/03', '13/09/05', 123, 101);
INSERT INTO CHORZY VALUES (118, '17/12/03', '19/05/05', 115, 105);
```

```
INSERT INTO BADANIA VALUES (101, '18/12/03', 300, 103, 110);
INSERT INTO BADANIA VALUES (102, '20/09/05', 100, 112, 111);
INSERT INTO BADANIA VALUES (103, '19/10/12', 280, 118, 103);
INSERT INTO BADANIA VALUES (104, '17/03/24', 100, 112, 116);
INSERT INTO BADANIA VALUES (105, '19/04/05', 90, 101, 121);
INSERT INTO BADANIA VALUES (106, '20/09/02', 500, 107, 105);
INSERT INTO BADANIA VALUES (107, '17/04/21', 200, 104, 104);
INSERT INTO BADANIA VALUES (108, '17/05/27', 120, 107, 122);
INSERT INTO BADANIA VALUES (109, '15/01/25', 280, 101, 104);
INSERT INTO BADANIA VALUES (110, '16/05/08', 180, 103, 116);
INSERT INTO BADANIA VALUES (111, '18/07/04', 120, 104, 123);
INSERT INTO BADANIA VALUES (112, '20/01/01', 200, 106, 119);
INSERT INTO BADANIA VALUES (113, '13/03/06', 120, 105, 122);
INSERT INTO BADANIA VALUES (114, '09/11/11', 500, 109, 104);
INSERT INTO BADANIA VALUES (115, '05/05/17', 100, 111, 119);
INSERT INTO BADANIA VALUES (116, '15/07/16', 230, 101, 123);
INSERT INTO BADANIA VALUES (117, '19/01/09', 90, 113, 107);
INSERT INTO BADANIA VALUES (118, '13/06/01', 400, 113, 104);
INSERT INTO BADANIA VALUES (119, '15/05/14', 500, 114, 105);
INSERT INTO BADANIA VALUES (120, '17/04/23', 100, 111, 101);
INSERT INTO BADANIA VALUES (121, '15/07/07', 100, 110, 104);
INSERT INTO BADANIA VALUES (122, '16/03/08', 100, 101, 107);
INSERT INTO BADANIA VALUES (123, '14/12/13', 250, 102, 104);
```

Załącznik 3

```
--Testowanie procedury zapisz_badanie oraz wyzwalacza znizka_badanie
BEGIN
```

```
znany_lekarz.zapisz_badanie(108,102,'20/07/13');
znany_lekarz.zapisz_badanie(108,103,'20/07/13');
znany_lekarz.zapisz_badanie(108,104,'20/07/13');
znany_lekarz.zapisz_badanie(108,105,'20/07/13');
znany_lekarz.zapisz_badanie(108,106,'20/07/13');
znany_lekarz.zapisz_badanie(108,107,'20/07/13');
```

```
znany_lekarz.zapisz_badanie(103,109,'20/03/13');
znany_lekarz.zapisz_badanie(103,109,'20/07/13');
```

```
znany_lekarz.zapisz_badanie(1030,110,'20/03/13');
znany_lekarz.zapisz_badanie(103,1100,'20/03/13');
END;
```

```
--Testowanie procedury najtansi_lekarze
```

```
BEGIN
znany_lekarz.najtansi_lekarze(103,4);
znany_lekarz.najtansi_lekarze(103,2);
znany_lekarz.najtansi_lekarze(1040,2);
```

END;

--Testowanie wyzwalacza ocena_upd

```
INSERT INTO OPINIE VALUES (101,5,'Bardzo przyjemna wizyta', 101, 103);
INSERT INTO OPINIE VALUES (102,2,null, 101, 101);
INSERT INTO OPINIE VALUES (103,1,'Nie mogło być gorzej', 101, 118);
INSERT INTO OPINIE VALUES (104,5,'Lekarz to gbur', 114, 114);
INSERT INTO OPINIE VALUES (105,3,null, 114, 103);
INSERT INTO OPINIE VALUES (106,2,'Nie dowiedziałem się nic nowego', 107, 104);
INSERT INTO OPINIE VALUES (107,5,'Było w porządku', 109, 120);
INSERT INTO OPINIE VALUES (108,3,null, 111, 109);
INSERT INTO OPINIE VALUES (109,4,null, 102, 113);
INSERT INTO OPINIE VALUES (110,7,'Najlepszy lekarz na jakiego trafiłam', 118, 106);
```

--Testowanie procedury najlepsi_lekarze

```
BEGIN
znany_lekarz.najlepsi_lekarze(101,2);
END;
```

--Sprawdzenie ręczne gdzie pracują najlepsi lekarze

```
SELECT * FROM LEKARZE L
JOIN ZATRUDNIENIE Z ON z.ID_LEKARZA = l.ID_LEKARZA
ORDER BY OPINIA DESC NULLS LAST;
```

--Testowanie funkcji najlepsza_placowka

```
DECLARE
v_placowka NUMBER;
BEGIN
v_placowka := znany_lekarz.najlepsza_placowka;
dbms_output.put_line('Placowka w której pracują najlepsi lekarze posiada id ' || v_placowka);
END;
```

--Testowanie funkcji najczestsza_choroba_wiek

```
DECLARE
v_choroba CHOROBY.NAZWA%TYPE;
v_choroba2 CHOROBY.NAZWA%TYPE;
BEGIN
v_choroba := znany_lekarz.najczestsza_choroba_wiek(40);
dbms_output.put_line('Najczęściej występująca choroba w ostatnim roku dla ludzi w wieku 40+-10 lat jest ' || v_choroba);
v_choroba := znany_lekarz.najczestsza_choroba_wiek(60);
dbms_output.put_line('Najczęściej występująca choroba w ostatnim roku dla ludzi w wieku 60+-10 lat jest ' || v_choroba);
END;
```

--Zapytanie o pacjentów którzy wydali więcej niż 500

```
SELECT ID_PACJENTA, WYDATKI
FROM(
SELECT b.ID_PACJENTA, sum(KOSZT) as WYDATKI FROM PACJENCI P
JOIN BADANIA B on b.ID_PACJENTA = p.ID_PACJENTA
GROUP BY b.ID_PACJENTA)
WHERE WYDATKI > 500;
```

--Zapytanie o lekarzy których koszt wizyty jest wyższy niż średni koszt wizyty u stomatologa

```
SELECT IMIE, NAZWISKO, KOSZT_WIZYT
FROM LEKARZE
WHERE KOSZT_WIZYT > (SELECT avg(KOSZT_WIZYT) FROM LEKARZE
NATURAL JOIN SPECJALIZACJA
WHERE NAZWA_SPEC = 'Stomatolog');
```

--Zapytanie ile razy i jaka choroba udało się wyleczyć w mniej niż rok

```
SELECT NAZWA, count(NAZWA) as LICZBA_WYLECZEN FROM CHOROZY
NATURAL JOIN CHOROBY
WHERE (DATA_WYLECZENIA - DATA_ZACHOROWANIA) < 365
GROUP BY NAZWA
ORDER BY LICZBA_WYLECZEN DESC;
```