



**POLITECHNIKA  
RZESZOWSKA**  
im. IGNACEGO ŁUKASIEWICZA



**WYDZIAŁ  
MATEMATYKI  
I FIZYKI STOSOWANEJ**  
POLITECHNIKI RZESZOWSKIEJ

## **Bazy danych**

---

### **Bazy danych kliniki medycyny estetycznej**

Ilona Krupa  
Dawid Stachiewicz  
2 rok FS0-DI  
Grupa P5

## Spis treści

1.	Koncepcja i metodyka projektu.....	3
1.1.	Sformułowanie zadania projektowego .....	3
1.2.	Założenia.....	3
1.3.	Diagram przypadków użycia .....	5
1.4.	Identyfikacja funkcji .....	6
1.5.	Diagram ERD .....	7
2.	<b>Projekt logiczny</b> .....	8
2.1.	Przejście z modelu ERD na model relacyjny (projekt logiczny).....	8
2.2.	Normalizacja .....	8
2.3.	Diagram relacyjnej bazy danych po normalizacji .....	13
3.	<b>ETAP TRZECI</b> .....	13
3.1.	Kwerendy.....	13
3.2	Skrypt DDL .....	19
3.2	Bibliografia .....	25

# 1. Koncepcja i metodyka projektu

## 1.1. Sformułowanie zadania projektowego

Celem niniejszego projektu jest utworzenie zarówno planu bazy danych oraz samej bazy dla kliniki medycyny estetycznej. Baza ta ma wspomóc funkcjonowanie kliniki, w szczególności w aspekcie obsługi klienta, rejestracji wizyt, komunikacji pomiędzy pacjentem a lekarzem bądź kosmetologiem, kontroli nad wizytami oraz ilością preparatów i aktywów trwałych. Baza ta może również spełniać rolę informacyjną o opiniach klientów. Wszystkie dane muszą być aktualizowane na bieżąco, aby klinika mogła być zarządzana w sposób efektywny.

## 1.2. Założenia

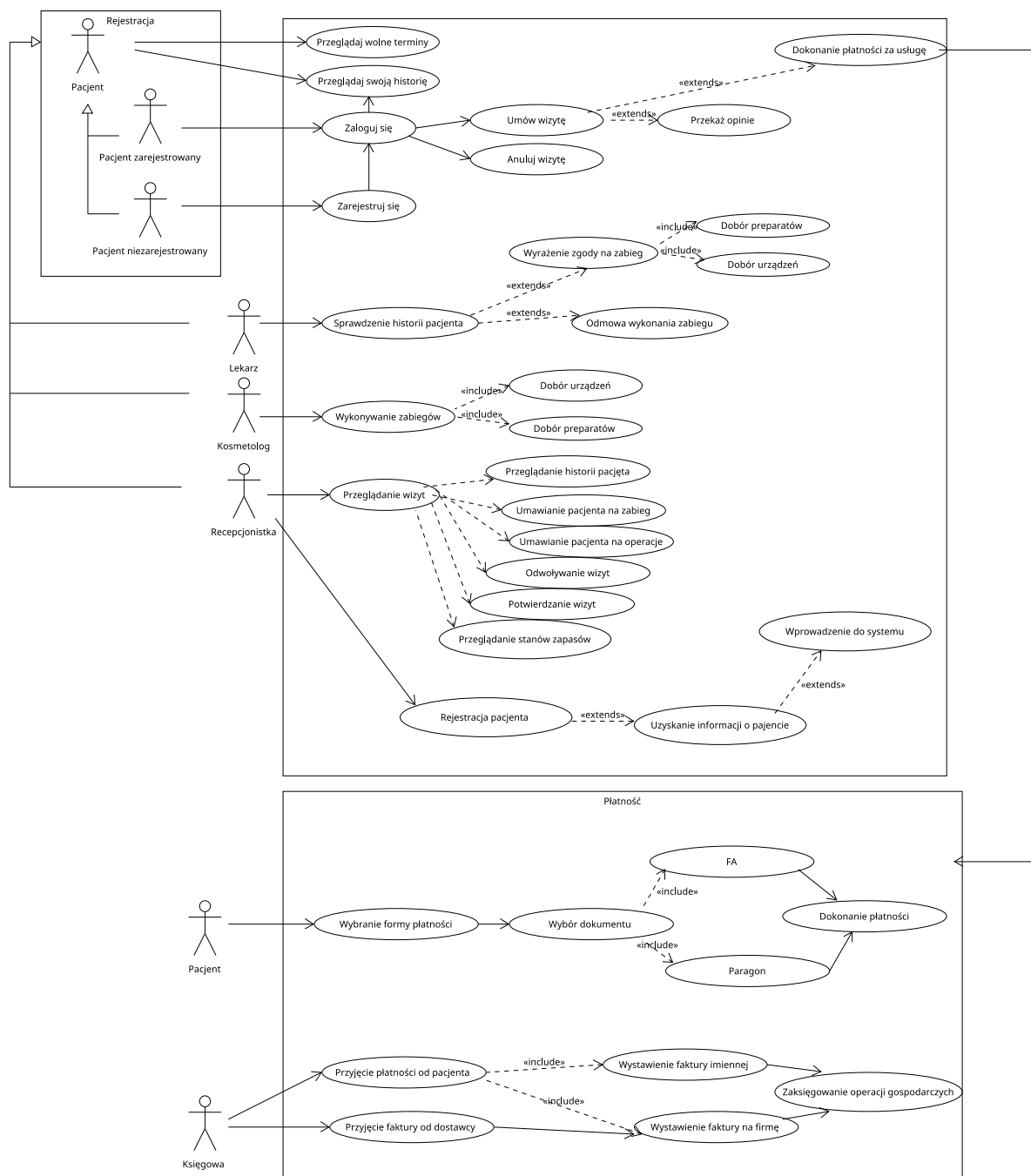
W klinice rejestrowani są pacjenci na wizyty z ustalonym terminem na poszczególne zabiegi i operacje. Każdy pacjent może być umówiony na kilka wizyt oraz posiadać szeroką historię zabiegów. Klinika oferuje wiele zabiegów wykonywanych przez kosmetologów i operacji wykonywanych przez lekarzy. Jeden lekarz jest uprawniony do wykonywania wielu zabiegów i operacji, kosmetolog jest uprawniony do wykonywania wielu zabiegów. Podczas zabiegów i operacji wykorzystywane są preparaty, wykonuje się je przy użyciu maszyny i urządzenia. Uwzględnienie stanu preparatów umożliwi kontrolowanie stanu zapasów, aby był on optymalny. Tabela monitorująca urządzenia i maszyny pozwoli na uniknięcie zapisu na zabiegi w jednym czasie, które wykonywane są na jednej maszynie, a także pozwoli na kontrole amortyzacyjną urządzeń.

Utworzone zostały następujące tabele:

- Wizyta – encja pośrednia ta przechowuje informacje o wizytach, które zostały zarezerwowane przez pacjentów. Tabela zawiera ID wizyty (klucz główny), datę i godzinę wizyty, status wizyty (zarezerwowana, odwołana, zrealizowana), opis problemu, komentarz osoby wykonującej zabieg.
- Pacjent – tabela asocjacyjna poświęcona jest na przechowywanie podstawowych danych o przyjmowanych pacjentach. Zawiera ID nadawane każdemu pacjentowi (klucz główny), imię, nazwisko, datę urodzenia, adres, numer telefonu, email;
- Pracownik – tabela ma na celu dobór specjalisty do danego zabiegu/operacji. Zawiera ID pracownika (klucz główny), imię, nazwisko, stanowisko (np. lekarz, pielęgniarka, kosmetolog, personel administracyjny), specjalizacja (np. dermatolog, chirurg plastyczny), data zatrudnienia, numer telefonu, email;

- Kategoria zabiegu – tabela pozwala sklasyfikować zabiegi wykonywane w placówce. Zawarte zostały w niej ID kategorii (klucz główny), nazwa kategorii, opis kategorii;
- Zabieg – tabela stworzona została w celu przechowywania pełnej oferty zabiegowej placówki.
- Historia pacjenta – encja pośrednia gromadzi informacje o wszystkich zabiegach wykonanych przez danego pacjenta w klinice. Zawiera ID historii (klucz główny), ID pacjenta (Klucz obcy wskazujący na pacjenta), ID zabiegu (klucz obcy wskazujący na zabieg), data wykonania zabiegu oraz komentarz od osoby wykonującej zabieg.
- Preparaty – tabela asocjacyjna stworzona została w celu audytowania stanów zapasów danych preparatów, pozwoli także na optymalizację stanów magazynowych. Zawiera ID preparatu (klucz główny), nazwa preparatu, opis, cena, data produkcji, data ważności, stany magazynowe.
- Maszyny i urządzenia – tabela asocjacyjna stworzona została w celu kontroli liczby urządzeń w klinice, wykluczenia przydzielenia jednej maszyny do wielu zabiegów równocześnie. Tabela posiada ID maszyny (klucz główny), ilość, zastosowanie oraz datę zakupu.
- Zestawy\_preparatów – encja pośrednia stworzona w celu łączenia wielu preparatów w wielu zabiegach i operacjach .
- Zestawy\_urządzeń – encja pośrednia asocjacyjna stworzona w celu łączenia wiele urządzeń w wielu zabiegach i operacjach.
- Opinie o zabiegach – tabela gromadzi opinie pacjentów o wykonanych zabiegach. Zawiera OpiniaID (Klucz główny),
- Operacje – tabela przedstawia oferowane w klinice operację. Zawiera ID operacji (klucz główny), nazwę operacji, opis, cenę, czas trwania;

### 1.3. Diagram przypadków użycia



Opis:

W klinice przyjmowani są zarówno pacjenci niezarejestrowani jak pacjenci już zarejestrowani. Po założeniu karty w klinice pacjenci dorastają możliwość zalogowania się do portalu pacjenta, gdzie mają dostęp do umówienia się na wizytę lub anulować, przeglądania swojej historii medycznej w której zawarte są komentarze od osób wykonujących zabieg/operacje, mogą również przekazać opinie.

W klinice pracują lekarz i kosmetolodzy, przeprowadzający odpowiednio operacji i zabiegi. Lekarz przed przystąpieniem do operacji musi sprawdzić historię pacjenta i podjąć decyzję czy operacja zostanie odwołana czy też przeprowadzona. Jeżeli zgoda na operację zostanie wyrażona lekarz musi podać na jakich preparatach będzie wykonywał operację oraz ich ilość, a także jakich maszyn i urządzeń potrzebuje. Kosmetolog również musi wprowadzić do systemu ilość preparatów potrzebnych do wykonania zabiegu i dobrać urządzenia do zabiegu.

Recepcjonistka posiada pełen wgląd w wizyty, przez co może przeglądać historię pacjentów, co usprawni rejestrację na kolejne zabiegi. Posiada także możliwość umawiania pacjentów na wizyty, odwoływania wizyt, potwierdzania wizyt. Prowadzi ona także kontrole nad stanami magazynowymi.

Pacjent może wybrać formę płatności za usługę oraz dokument zapłaty (faktura, paragon). Księgowy wystawia faktury, zaksięgowuje operacje na dokumentach księgowych własnych i obcych.

#### 1.4. Identyfikacja funkcji

Zarządzanie pacjentami:

- Dodawanie nowych pacjentów do systemu.
- Aktualizacja informacji pacjentów, takich jak dane osobowe, historie medyczne, itp.
- Przeglądanie listy pacjentów.

Rejestracja wizyt na zabiegi i operacje:

- Rejestracja wizyt pacjentów.
- Przypisywanie pacjentom konkretnych zabiegów.
- Monitorowanie dostępności zabiegów i operacji.
- Dodawanie i usuwanie pracowników medycznych.

Zarządzanie kadrą:

- Przypisywanie specjalizacji i uprawnień pracownikom.
- Przeglądanie listy pracowników medycznych.

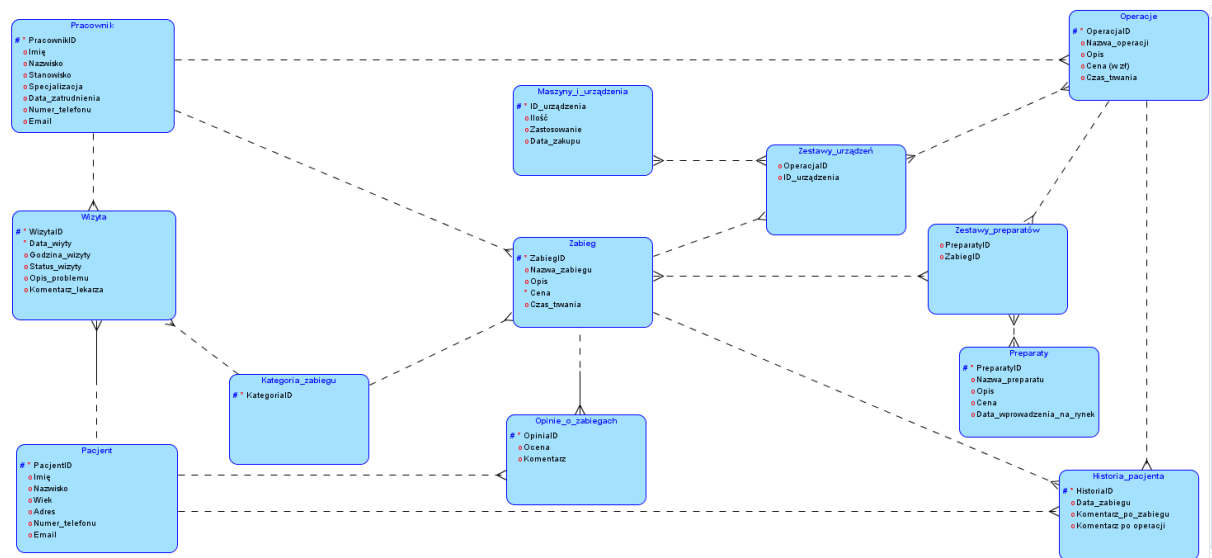
Prowadzenie historii pacjentów:

- Monitorowanie historii zabiegów pacjenta.
- Przeglądanie przeprowadzonych operacji i zabiegów.

Audyt preparatów:

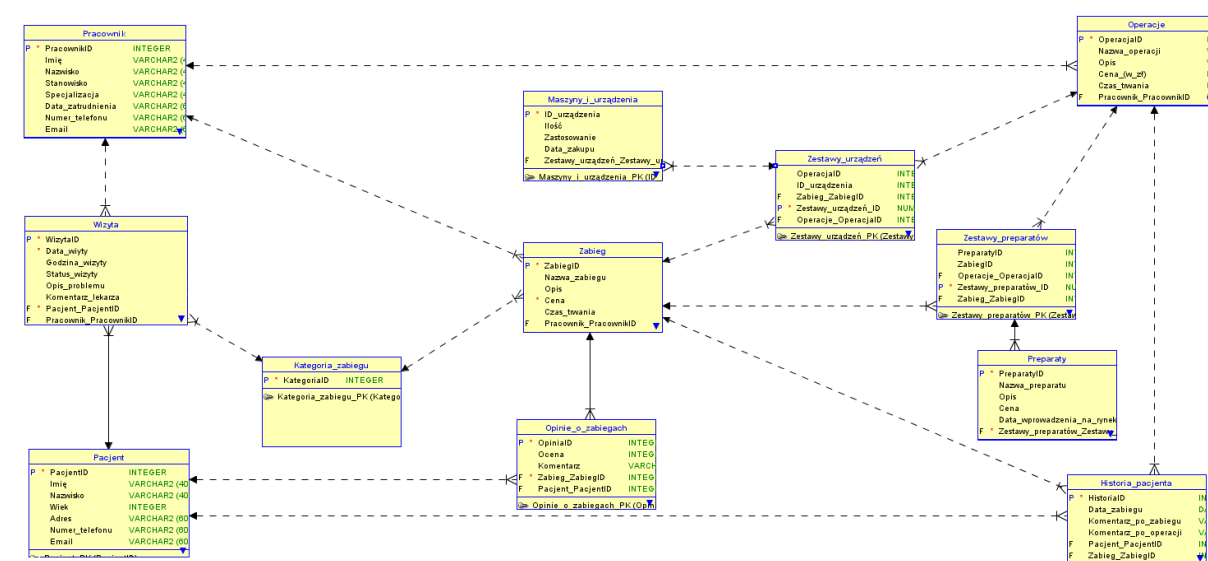
- Dodawanie nowych preparatów medycznych do magazynu.

- ### 1.5. Diagram ERD



## 2. Projekt logiczny

### 2.1. Przejście z modelu ERD na model relacyjny (projekt logiczny)



### 2.2. Normalizacja

- Postać zdenormalizowana:**

Celem sprawdzenia poprawności stworzonych wcześniej diagramów dokonano normalizacji. Encje związane zostały do 4 głównych tabel: Pracownik, Pacjent, Operacja i Zabieg.

Zdenormalizowana forma wyznaczonych tabel:

Pracownicy	Pacjent	Operacje	Zabieg
Imie	Imie	Nazwa_operacji	Nazwa_zab
Nazwisko	Nazwisko	Opis	Opis
Wiek	Wiek	Cena	Cena
Adres	Adres	Czas_trwania	Czas_trwania
Telefon	Telefon	Nazwa_urzadzenia	Nazwa_urzadzenia
E-mail	E-mail	Ilosc_urz	Ilosc_urz
Data_wiz	Data_wiz	Nazwa_preparatu	Nazwa_preparatu
Godzina_wiz	Godzina_wiz	Cena_prep	Cena_prep
Status	Status		
Nazwa_zabiegu	Nazwa_zabiegu		
Ocena	Ocena		
Komentarz	Komentarz		
Nazwa_op	Nazwa_op		
Data_zabieg	Data_zabieg		
Data_op	Data_op		



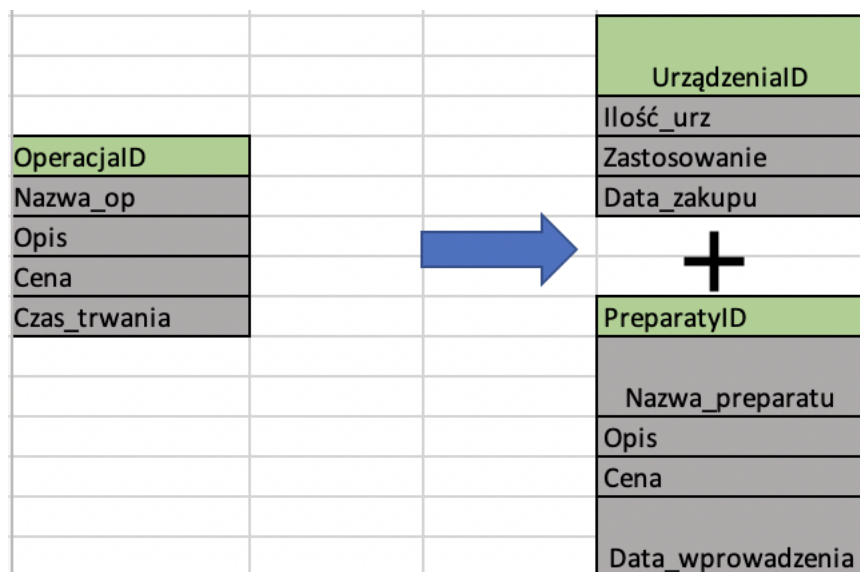
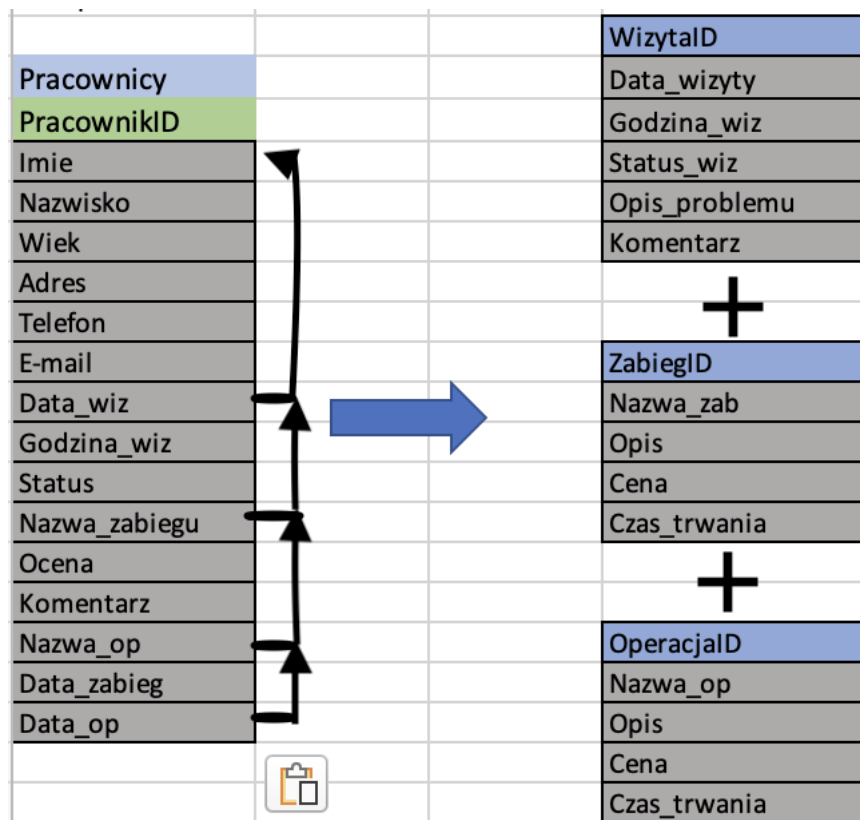
- **Normalizacja pierwszego stopnia:**

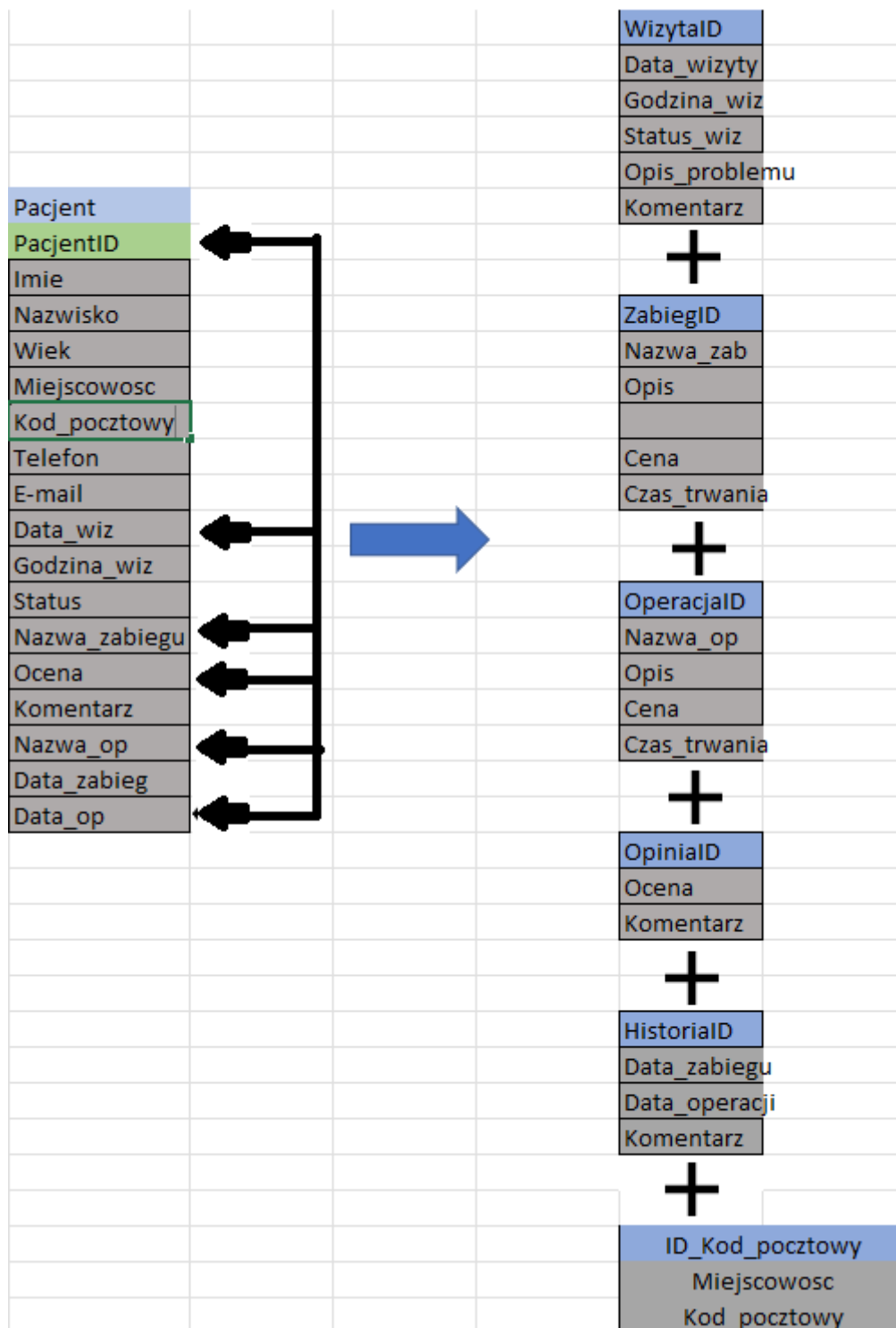
W celu przeprowadzenia normalizacji pierwszego stopnia, konieczne jest, aby wszystkie wartości w tabelach były atomowe i miały klucz główny. W przypadku opisanego powyżej, kolumna „Adres” nie spełnia tych kryteriów, gdyż zawiera zarówno ulicę i numer, jak i kod pocztowy i miasto. Aby dostosować ją do standardów, konieczne jest rozdzielenie jej na unikalne wartości. Po dokonaniu tego podziału, nasze tabele spełniają wymogi normalizacji pierwszego stopnia.

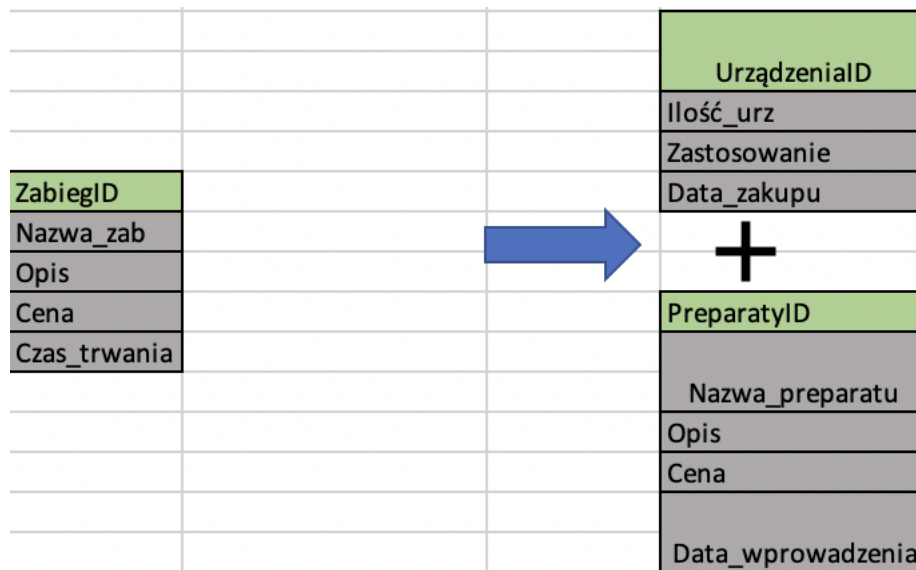
Pracownicy	Pacjent	Operacje	Zabieg
PracownikID	PacjentID	OperacjaID	ZabiegID
Imie	Imie	Nazwa_operacji	Nazwa_zab
Nazwisko	Nazwisko	Opis	Opis
Wiek	Wiek	Cena	Cena
Adres	Adres	Czas_trwania	Czas_trwania
Telefon	Telefon	Nazwa_urzadzenia	Nazwa_urzadzenia
E-mail	E-mail	Ilość_urz	Ilość_urz
Data_wiz	Data_wiz	Nazwa_preparatu	Nazwa_preparatu
Godzina_wiz	Godzina_wiz		
Status	Status	Cena_prep	Cena_prep
Nazwa_zabiegu	Nazwa_zabiegu		
Ocena	Ocena		
Komentarz	Komentarz		
Nazwa_op	Nazwa_op		
Data_zabieg	Data_zabieg		
Data_op	Data_op		

- **Normalizacja drugiego stopnia**

Normalizacja stopnia drugiego polega na spełnieniu warunku, aby zdefiniowany był każdy klucz główny a atrybuty zawarte w tabeli były zależne od tego klucza.  
(wartości zielone to klucze główne; wartości niebieskie to klucze obce)





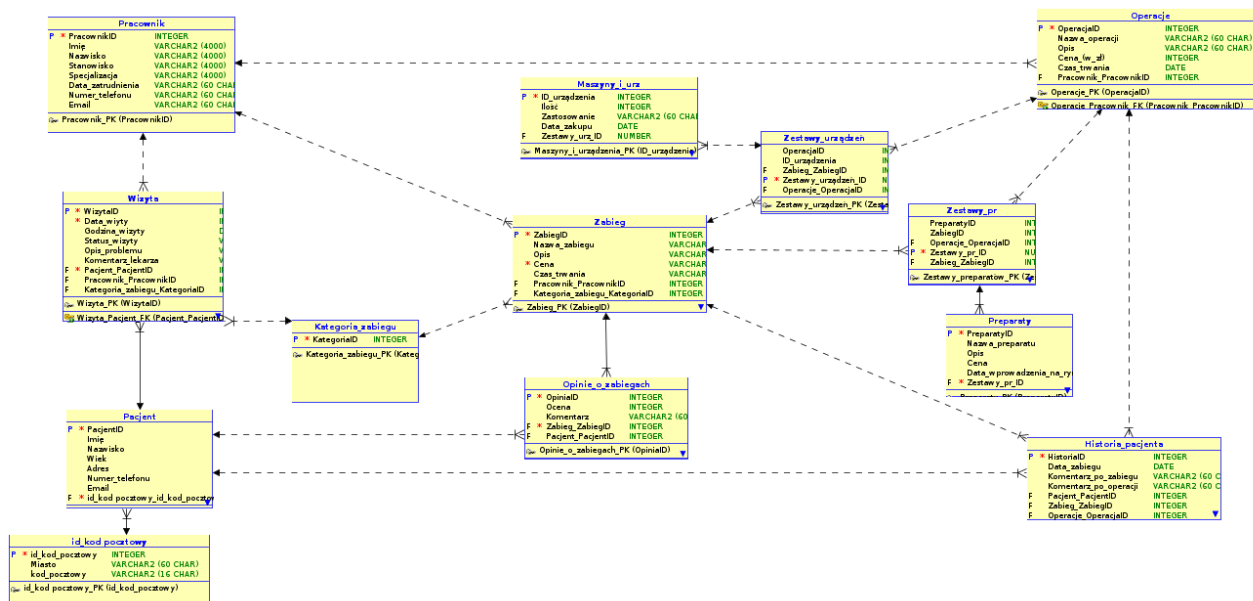


Tabele spełniają kryteria drugiej normalizacji.

W trzeciej normalizacji kluczowym aspektem jest zagwarantowanie, że kolumny w tabeli nie zależą przejściowo od klucza głównego. Po dokładnej analizie tabeli stwierdzono, że spełnia ona kryteria trzeciej normalizacji, z wyjątkiem atrybutów "Miejscowość" i "Kod pocztowy". Aby rozwiązać to wyzwanie, można by utworzyć dodatkową tabelę o nazwie "Kody pocztowe".

Niemniej jednak, w praktyce rzadko normalizuje się kody pocztowe do trzeciej postaci normalnej. Łączenie tabel jest kosztowne pod względem wydajności, dlatego często decyduje się na utrzymanie modelu w drugiej postaci normalnej w celu zoptymalizowania działania systemu.

## 2.3. Diagram relacyjnej bazy danych po normalizacji



## 3. ETAP TRZECI

### 3.1. Kwerendy

Dokonano 10 zapytań. Każde zapytanie zawiera kod SQL, algebrę relacyjną oraz wynik zapytania.

- Ilość umówionych wizyt danego pacjenta malejąco:  
 $\pi_{\text{imie, nazwisko, iloscwizyt}}(\sigma_{\text{data} \geq \text{current\_date}}(\rho_{\text{imie, nazwisko}}(\gamma_{\text{pacjent\_id, COUNT(wizyta\_id)} \text{ AS iloscwizyt}}(\text{Wizyta} \bowtie \text{pacjent\_id} = \text{pacjent\_pacjentid}(\text{Pacjent}) \bowtie \text{Pacjent})))$

IMIE	NAZWISKO	ILOSCWIZYT
1	Katarzyna Kwiatkowska	2
2	Patrycja Górka	2
3	Oliwia Jabłońska	2
4	Anna Nowak	2
5	Jan Kowalski	2
6	Michał Zając	1
7	Karolina Pawlak	1
8	Marek Tomczak	1
9	Magdalena Wójcik	1
10	Adam Kaczmarek	1
11	David Zalewski	1
12	Robert Jankowski	1
13	Piotr Wiśniewski	1
14	Łukasz Szymański	1

- Najgorsze oceny zabiegów:  
 $\pi_{ocena, nazwa\_zabiegu}(\sigma_{ocena < 3}$   
 $(Opinie\_o\_zabiegach \bowtie_{zabieg\_zabiegid = zabiegidZabieg})$

Worksheet | Query Builder

```
SELECT Opinie_o_zabiegach.Ocena, zabieg.nazwa_zabiegu
FROM Opinie_o_zabiegach
INNER JOIN zabieg ON opinie_o_zabiegach.zabieg_zabiegid = Zabieg.zabiegid
WHERE opinie_o_zabiegach.ocena < 3
```

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0.005 seconds

	OCENA	NAZWA_ZABIEGU
1	2	Korekta nosa
2	2	Modelowanie ust

- Pokazanie dostępnych maszyn z uwzględnienie ilości oraz zastosowania:  
 $\pi_{id\_urzadzenia, zastosowanie}(Maszyny\_i\_Urz)$

Worksheet | Query Builder

```
SELECT Maszyny_i_urz.ID_urzadzenia, Maszyny_i_urz.Ilosc, Maszyny_i_urz.Zastosowanie
FROM Maszyny_i_urz;
```

Script Output x Query Result x

SQL | All Rows Fetched: 20 in 0.007 seconds

	ID_URZADZENIA	ILOSC	ZASTOSOWANIE
1	1	5	Drukarka laserowa
2	2	3	Komputer stacjonarny
3	3	2	Skaner
4	4	1	Kserokopiarka
5	5	4	Laptop
6	6	2	Drukarka atramentowa
7	7	1	Projektor multimedialny
8	8	3	Telefon konferencyjny
9	9	1	Kamera przemysłowa
10	10	5	Tablet
11	11	2	Drukarka wielofunkcyjna
12	12	1	Fax
13	13	4	Smartfon
14	14	2	Laser do grawerowania
15	15	1	Drukarka termiczna
16	16	3	Router sieciowy
17	17	1	Kserokopiarka kolorowa
18	18	4	Komputer przenośny
19	19	1	Drukarka do etykiet
20	20	5	Laptop hybrydowy

- Obliczenie średnich ocen dla każdego zabiegu oraz wyświetlenie malejąco:

$\pi$ nazwa\_zabiegu, AVG(ocena) AS sredniaocena

( $\gamma$ zabiegid, AVG(ocena) AS sredniaocena

(Opinie\_o\_zabiegach  $\bowtie$  zabieg\_zabiegid=zabiegidZabieg)  $\bowtie$  Zabieg)

Worksheet    Query Builder	
<pre>SELECT Z.Nazwa_zabiegu, AVG(O.Ocena) AS SredniaOcena FROM Zabieg Z JOIN Opinie_o_zabiegach O ON Z.ZabiegID = O.Zabieg_ZabiegID GROUP BY Z.Nazwa_zabiegu ORDER BY SredniaOcena DESC</pre>	
Script Output    Query Result	
SQL   All Rows Fetched: 20 in 0.431 seconds	
NAZWA_ZABIEGU	SREDNIAOCENA
1 Zmarszczki twarzy	5
2 Leczenie trądziku bliznowaciejącego	5
3 Wypełnianie zmarszczek kwasem hialuronowym	5
4 Laserowe usuwanie owłosienia	5
5 Usuwanie blizn	5
6 Peeling chemiczny	5
7 Mezoterapia skóry	5
8 Mezoterapia igłowa	4
9 Mezoterapia mikroigłowa	4
10 Modelowanie ciała	4
11 Cellulit na udach	4
12 Odmładzanie skóry	4
13 Powiększanie ust	4
14 Przebarwienia skóry	4
15 Laserowe usuwanie przebarwień	3
16 Korekcja brwi	3
17 Leczenie trądziku różowatego	3
18 Redukcja zmarszczek mimicznych	3
19 Modelowanie ust	2
20 Korekta nosa	2

- Obliczenie i pokazanie pozostałego czasu do całkowitej amortyzacji urządzeń (przyjęto amortyzację liniową, całkowita amortyzacja to 2 lata od zakupu):  
 $\pi ID\_urządzenia, Ilość, Zastosowanie, Data\_Zakupu, PozostałyCzasZamortyzowaniaMiesiące$   
 $a Miesiące (\sigma PozostałyCzasZamortyzowaniaMiesiące \leftarrow (2 \times 12) - MONTHS\_BETWEEN(SYSDATE, Data\_Zakupu)(Maszyny\_i\_urz))$

sys

Worksheet Query Builder

```
SELECT
  ID_urządzenia,
  Ilość,
  Zastosowanie,
  Data_zakupu,
  (2 * 12) - MONTHS_BETWEEN(SYSDATE, Data_zakupu) AS PozostałyCzasZamortyzowaniaMiesiące
FROM Maszyny_i_urz;
```

Script Output x Query Result x

All Rows Fetched: 20 in 0,005 seconds

ID_URZĄDZENIA	ILOSC	ZASTOSOWANIE	DATA_ZAKUPU	POZ...
1	1	5 Drukarka laserowa	2023-01-15	11,...
2	2	3 Komputer stacjonarny	2022-09-20	8
3	3	2 Skaner	2023-05-10	15,...
4	4	1 Kserokopiarka	2023-03-08	13,...
5	5	4 Laptop	2022-11-25	10,...
6	6	2 Drukarka atramentowa	2023-02-14	12,...
7	7	1 Projektor multimedialny	2022-12-30	11,...
8	8	3 Telefon konferencyjny	2023-06-05	16,...
9	9	1 Kamera przemysłowa	2023-04-18	14,...
10	10	5 Tablet	2022-10-12	8,7...
11	11	2 Drukarka wielofunkcyjna	2023-01-05	11,...
12	12	1 Fax	2023-07-22	18,...
13	13	4 Smartfon	2022-08-15	6,8...
14	14	2 Laser do grawerowania	2023-03-30	14,...
15	15	1 Drukarka termiczna	2022-11-10	9,6...
16	16	3 Router sieciowy	2023-05-28	16,...
17	17	1 Kserokopiarka kolorowa	2023-02-02	12

- Wyświetlenie kobiet zarejestrowanych w systemie:  
 $\pi Pacjent.ID\_Pacjenta, Pacjent.Imie, Pacjent.Nazwisko, Pacjent.Data\_Urodzenia, Pacjen$   
 $t.Adres (\sigma Pacjent.Imie LIKE '%a' (Pacjent))$

Worksheet Query Builder

```
SELECT * FROM Pacjent where imie LIKE '%a'
```

Script Output x Query Result x

All Rows Fetched: 10 in 0,013 seconds

PACJENTID	IMIE	NAZWISKO	WIEK	ADRES	NUMER_TELEFONU	EMAIL
1	2 Anna	Nowak	25	ul. Słoneczna 2	987-654-321	anna.nowak@email.com
2	4 Ewa	Dąbrowska	28	ul. Polna 4	444-555-666	ewa.dabrowska@email.com
3	6 Katarzyna	Kwiatkowska	22	ul. Różana 6	333-222-111	katarzyna.kwiatkowska@email.com
4	8 Magdalena	Wójcik	27	ul. Mokra 8	999-888-777	magdalena.wojcik@email.com
5	10 Karolina	Pawlak	29	ul. Biała 10	666-777-888	karolina.pawlak@email.com
6	12 Monika	Lewandowska	26	ul. Cicha 12	888-555-222	monika.lewandowska@email.com
7	14 Natalia	Kowalczyk	24	ul. Wiosenna 14	444-222-888	natalia.kowalczyk@email.com
8	16 Agnieszka	Sawicka	23	ul. Krótka 16	222-888-555	agnieszka.sawicka@email.com
9	18 Oliwia	Jabłońska	21	ul. Parkowa 18	888-444-222	oliwia.jablonska@email.com
10	20 Patrycja	Górska	28	ul. Malinowa 20	444-999-666	patrycja.gorska@email.com



- Wyświetlenie preparatów które są na rynku dłużej niż rok  
 $\pi$  nazwa\_preparatu, data\_wprowadzenia\_na\_rynek ( $\sigma$  12 months\_between(current\_date, data\_wprowadzenia\_na\_rynek) > 1 (Preparaty))

Worksheet Query Builder

```
SELECT Nazwa_preparatu, Data_wprowadzenia_na_rynek
FROM Preparaty
WHERE MONTHS_BETWEEN(SYSDATE, Data_wprowadzenia_na_rynek)/12 > 1;
```

Script Output x Query Result x

SQL | All Rows Fetched: 13 in 0,009 seconds

NAZWA_PREPARATU	DATA_WPROWADZENIA_NA_RYNEK
1 Botox	22/01/01
2 Hyaluronidase	22/02/15
3 Retinol Cream	22/03/10
4 Peptydy kolagenowe	22/04/25
5 Kwas hialuronowy	22/05/20
6 Witamina C Serum	22/06/12
7 Mikrodermabrazja kawitacyjna	22/07/08
8 Zinc Oxide Sunscreen	22/08/14
9 Krem przeciwstarzeniowy	22/09/30
10 Ziołowe toniki do twarzy	22/10/25
11 Peeling enzymatyczny	22/11/12
12 Krem nawilżający z kwasem hialuronowym	22/12/03
13 Kwas salicylowy	23/01/08

- Wyświetlenie umówionych wizyt dla pacjenta o konkretnym identyfikatorze:  
 $\pi$  pacjent.pacjentID, pacjent.imie, pacjent.nazwisko, wizyta.opis\_problemu  
 $(\sigma$  pacjent.PacjentID LIKE '1' (pacjent  $\bowtie$  pacjent.pacjentID = wizyta.Pacjent\_PacjentID  
wizyta))

Worksheet Query Builder

```
SELECT
pacjent.pacjentid,
pacjent.imie,
pacjent.nazwisko,
wizyta.opis_problemu
FROM Pacjent
INNER JOIN Wizyta ON Pacjent.PacjentID = Wizyta.PACJENT_PacjentID
WHERE Pacjent.PacjentID LIKE '1';
```

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0,006 seconds

PACJENTID	IMIE	NAZWISKO	OPIS_PROBLEMU
1	1 Jan	Kowalski	Zmarszczki twarzy
2	1 Jan	Kowalski	Korekcja brwi

- Wyświetlenie nazwy, ceny oraz czasu dla każdego zabiegu w klinice:  
 $\pi_{nazwa\_zabiegu, opis, czas\_trwania}(Zabieg)$

NAZWA_ZABIEGU	OPIS	CZAS_TRWANIA
1 Zmarszczki twarzy	Redukcja zmarszczek mimicznych na twarzy	60 min
2 Cellulit na udach	Zabieg redukujący widoczność cellulitu na udach	45 min
3 Przebarwienia skóry	Zabieg niwelujący przebarwienia skóry	50 min
4 Modelowanie ust	Zabieg modelowania i korekty kształtu ust	75 min
5 Redukcja zmarszczek mimicznych	Zabieg redukujący zmarszczki mimiczne	70 min
6 Usuwanie blizn	Zabieg usuwający blizny na skórze	55 min
7 Mezoterapia skóry	Zabieg poprawiający kondycję skóry przez mezoterapię	40 min
8 Leczenie trądziku różowatego	Zabieg mający na celu łagodzenie trądziku różowatego	45 min
9 Korekta nosa	Zabieg korekty kształtu nosa	90 min
10 Odmładzanie skóry	Zabieg odmładzający skórę	75 min
11 Mezoterapia igłowa	Zabieg mezoterapii wykonany za pomocą igieł	50 min
12 Laserowe usuwanie owłosienia	Zabieg usuwania owłosienia za pomocą lasera	100 min
13 Peeling chemiczny	Zabieg złuszczący skórę poprzez peeling chemiczny	55 min
14 Powiększanie ust	Zabieg powiększania ust	70 min
15 Korekcja brwi	Zabieg korekcji kształtu brwi	45 min
16 Laserowe usuwanie przebarwień	Zabieg usuwania przebarwień skóry za pomocą lasera	60 min

- Zapytanie jaką część zabiegu stanowi cena wykorzystanego do niego preparatu:  
 $\pi_{nazwa\_zabiegu, nazwa\_preparatu, cena} \rightarrow CenaZabiegu, cena \rightarrow CenaPreparatu, (cena / cena) \times 100 \rightarrow ProcentCenyPreparatuWZabiegu$   
 $(\sigma_{p.zabiegID=z.zabiegID \wedge p.PreparatyID=p.PreparatyID})$   
 $(Zestaw_{pr} \bowtie_{p.zabiegID=z.zabiegID \wedge p.PreparatyID=p.PreparatyID} preparaty))$

NAZWA_ZABIEGU	NAZWA_PREPARATU	CENAZABIEGU	CENAPREPARATU	PROCENTCENYPREPARATUWZABIEGU
1 Zmarszczki twarzy	Botox	500	1200	240
2 Cellulit na udach	Hyaluronidase	350	800	228.571428571428571428571428571429
3 Przebarwienia skóry	Retinol Cream	400	150	37.5
4 Modelowanie ust	Peptydy kolagenowe	600	300	50
5 Redukcja zmarszczek mimicznych	Kwas hialuronowy	550	500	90.90909090909090909090909090909091
6 Usuwanie blizn	Witamina C Serum	450	250	55.55555555555555555555555555555556
7 Mezoterapia skóry	Mikrodermabrazja kawitacyjna	300	400	133.33333333333333333333333333333333
8 Leczenie trądziku różowatego	Zinc Oxide Sunscreen	350	30	8.57142857142857142857142857142857
9 Korekta nosa	Krem przeciwstarzeniowy	700	180	25.71428571428571428571428571428571
10 Odmładzanie skóry	Ziołowe toniki do twarzy	600	40	6.66666666666666666666666666666667
11 Mezoterapia igłowa	Peeling enzymatyczny	400	70	17.5
12 Laserowe usuwanie owłosienia	Krem nawilżający z kwasem hialuronowym	800	120	15
13 Peeling chemiczny	Kwas salicylowy	450	60	13.33333333333333333333333333333333
14 Powiększanie ust	Kwas migdałowy	550	90	16.36363636363636363636363636363636

Dzięki temu zapytaniu klinika wie ceny których zabiegów należy podnieść .

### 3.2 Skrypt DDL

```
-- Generated by Oracle SQL Developer Data Modeler 21.2.0.183.1957
-- at:      2024-01-11 11:20:13 CET
-- site:    Oracle Database 12c
-- type:    Oracle Database 12c
```

```
-- predefined type, no DDL - MDSYS.SDO_GEOMETRY
```

```
-- predefined type, no DDL - XMLTYPE
```

```
CREATE TABLE historia_pacjenta (
  historiaid      INTEGER NOT NULL,
  data_zabiegu    VARCHAR2(60 CHAR),
  komentarz_po_zabiegu VARCHAR2(60 CHAR),
  komentarz_po_operacji VARCHAR2(60 CHAR),
  pacjent_pacjentid INTEGER,
  zabieg_zabiegid  INTEGER,
  operacje_operacjaid INTEGER
);
```

```
ALTER TABLE historia_pacjenta ADD CONSTRAINT historia_pacjenta_pk PRIMARY
KEY ( historiaid );
```

```
CREATE TABLE kategoria_zabiegu (
  kategoriaid INTEGER NOT NULL
);
```

```
ALTER TABLE kategoria_zabiegu ADD CONSTRAINT kategoria_zabiegu_pk PRIMARY
KEY ( kategoriaid );
```

```
CREATE TABLE maszyny_i_urz (
  id_urzadzenia INTEGER NOT NULL,
  ilosc          INTEGER,
  zastosowanie  VARCHAR2(60 CHAR),
  data_zakupu   VARCHAR2(60 CHAR),
  zestawy_urz_id NUMBER
);
```

```
ALTER TABLE maszyny_i_urz ADD CONSTRAINT maszyny_i_urzadzenia_pk PRIMARY
KEY ( id_urzadzenia );
```

```
CREATE TABLE operacje (
  operacjaid      INTEGER NOT NULL,
  nazwa_operacji  VARCHAR2(60 CHAR),
  opis            VARCHAR2(60 CHAR),
  "Cena_(w_zl)"   INTEGER,
  czas_trwania    integer,
```

```
    pracownik_pracownikid INTEGER
);
```

```
ALTER TABLE operacje ADD CONSTRAINT operacje_pk PRIMARY KEY ( operacjaid );
```

```
CREATE TABLE opinie_o_zabiegach (
    opiniaid      INTEGER NOT NULL,
    ocena         INTEGER,
    komentarz     VARCHAR2(60 CHAR),
    zabieg_zabiegid INTEGER NOT NULL,
    pacjent_pacjentid INTEGER
);
```

```
ALTER TABLE opinie_o_zabiegach ADD CONSTRAINT opinie_o_zabiegach_pk
PRIMARY KEY ( opiniaid );
```

```
CREATE TABLE pacjent (
    pacjentid    INTEGER NOT NULL,
    imie         VARCHAR2(4000),
    nazwisko     VARCHAR2(4000),
    wiek         INTEGER,
    adres        VARCHAR2(60 CHAR),
    numer_telefonu VARCHAR2(60 CHAR),
    email        VARCHAR2(60 CHAR)
);
```

```
ALTER TABLE pacjent ADD CONSTRAINT pacjent_pk PRIMARY KEY ( pacjentid );
```

```
CREATE TABLE pracownik (
    pracownikid  INTEGER NOT NULL,
    imie         VARCHAR2(4000),
    nazwisko     VARCHAR2(4000),
    stanowisko   VARCHAR2(4000),
    specjalizacja VARCHAR2(4000),
    data_zatrudnienia VARCHAR2(60 CHAR),
    numer_telefonu VARCHAR2(60 CHAR),
    email        VARCHAR2(60 CHAR)
);
```

```
COMMENT ON COLUMN pracownik.nazwisko IS
    '
    ';
```

```
ALTER TABLE pracownik ADD CONSTRAINT pracownik_pk PRIMARY KEY (
pracownikid );
```

```
CREATE TABLE preparaty (
    preparatyid  INTEGER NOT NULL,
    nazwa_preparatu VARCHAR2(60 CHAR),
    opis        VARCHAR2(60 CHAR),
    cena        VARCHAR2(60 CHAR),
```

```
data_wprowadzenia_na_rynek VARCHAR2(60 CHAR),
zestawy_pr_id              NUMBER NOT NULL
);
```

```
ALTER TABLE preparaty ADD CONSTRAINT preparaty_pk PRIMARY KEY ( preparatyid
);
```

```
CREATE TABLE wizyta (
wizytaid                INTEGER NOT NULL,
data_wizyty             VARCHAR2(60 CHAR) NOT NULL,
godzina_wizyty           VARCHAR2(60 CHAR),
status_wizyty            VARCHAR2(60 CHAR),
opis_problemu            VARCHAR2(60 CHAR),
komentarz_lekarza        VARCHAR2(60 CHAR),
pacjent_pacjentid        INTEGER NOT NULL,
pracownik_pracownikid    INTEGER,
kategoria_zabiegu_kategoriaid INTEGER
);
```

```
ALTER TABLE wizyta ADD CONSTRAINT wizyta_pk PRIMARY KEY ( wizytaid );
```

```
CREATE TABLE zabieg (
zabiegid                INTEGER NOT NULL,
nazwa_zabiegu            VARCHAR2(60 CHAR),
opis                     VARCHAR2(60 CHAR),
cena                     VARCHAR2(60 CHAR) NOT NULL,
czas_trwania             VARCHAR2(60 CHAR),
pracownik_pracownikid    INTEGER,
kategoria_zabiegu_kategoriaid INTEGER
);
```

```
ALTER TABLE zabieg ADD CONSTRAINT zabieg_pk PRIMARY KEY ( zabiegid );
```

```
CREATE TABLE zestawy_pr (
preparatyid             INTEGER,
zabiegid                 INTEGER,
operacje_operacjaid      INTEGER,
zestawy_pr_id            NUMBER NOT NULL,
zabieg_zabiegid           INTEGER
);
```

```
ALTER TABLE zestawy_pr ADD CONSTRAINT zestawy_preparatow_pk PRIMARY KEY
( zestawy_pr_id );
```

```
CREATE TABLE zestawy_urzadzen (
operacjaid              INTEGER,
id_urzadzenia           INTEGER,
zabieg_zabiegid          INTEGER,
zestawy_urzadzen_id      NUMBER NOT NULL,
operacje_operacjaid      INTEGER
);
```

);

ALTER TABLE zestawy\_urzadzen ADD CONSTRAINT zestawy\_urzadzen\_pk PRIMARY  
KEY ( zestawy\_urzadzen\_id );

CREATE TABLE id\_kod\_pocztowy (  
id\_kod\_pocztowy INTEGER,  
misto VARCHAR(255)  
);

ALTER TABLE id\_kod\_pocztowy ADD CONSTRAINT id\_kod\_pocztowy\_pk PRIMARY  
KEY ( id\_kod\_pocztowy );

ALTER TABLE id\_kod\_pocztowy  
ADD CONSTRAINT id\_kod\_pocztowy\_fk FOREIGN KEY ( Pacjent\_PacjentID )  
REFERENCES Pacjent ( Pacjentid );

ALTER TABLE historia\_pacjenta  
ADD CONSTRAINT historia\_pacjenta\_operacje\_fk FOREIGN KEY (  
operacje\_operacjaid )  
REFERENCES operacje ( operacjaid );

ALTER TABLE historia\_pacjenta  
ADD CONSTRAINT historia\_pacjenta\_pacjent\_fk FOREIGN KEY ( pacjent\_pacjentid )  
REFERENCES pacjent ( pacjentid );

ALTER TABLE historia\_pacjenta  
ADD CONSTRAINT historia\_pac\_zabb\_fk FOREIGN KEY ( zabieg\_zabiegid )  
REFERENCES zabieg ( zabiegid );

ALTER TABLE maszyny\_i\_urz  
ADD CONSTRAINT maszyny\_i\_urzadz\_zes\_urz\_fk FOREIGN KEY ( zestawy\_urz\_id )  
REFERENCES zestawy\_urzadzen ( zestawy\_urzadzen\_id );

ALTER TABLE operacje  
ADD CONSTRAINT operacje\_pracownik\_fk FOREIGN KEY ( pracownik\_pracownikid )  
REFERENCES pracownik ( pracownikid );

ALTER TABLE opinie\_o\_zabiegach  
ADD CONSTRAINT opinie\_o\_zabiegach\_pacjent\_fk FOREIGN KEY ( pacjent\_pacjentid  
)  
REFERENCES pacjent ( pacjentid );

ALTER TABLE opinie\_o\_zabiegach  
ADD CONSTRAINT opinie\_o\_zabieg\_fk FOREIGN KEY ( zabieg\_zabiegid )  
REFERENCES zabieg ( zabiegid );

ALTER TABLE preparaty  
ADD CONSTRAINT preparaty\_zestawy\_preparatow\_fk FOREIGN KEY ( zestawy\_pr\_id  
)

```

REFERENCES zestawy_pr ( zestawy_pr_id );

ALTER TABLE wizyta
  ADD CONSTRAINT wizyta_kategoria_zabiegu_fk FOREIGN KEY (
kategoria_zabiegu_kategoriaid )
  REFERENCES kategoria_zabiegu ( kategoriaid );

ALTER TABLE wizyta
  ADD CONSTRAINT wizyta_pacjent_fk FOREIGN KEY ( pacjent_pacjentid )
  REFERENCES pacjent ( pacjentid );

ALTER TABLE wizyta
  ADD CONSTRAINT wizyta_pracownik_fk FOREIGN KEY ( pracownik_pracownikid )
  REFERENCES pracownik ( pracownikid );

ALTER TABLE zabieg
  ADD CONSTRAINT zabieg_kategoria_zabiegu_fk FOREIGN KEY (
kategoria_zabiegu_kategoriaid )
  REFERENCES kategoria_zabiegu ( kategoriaid );

ALTER TABLE zabieg
  ADD CONSTRAINT zabieg_pracownik_fk FOREIGN KEY ( pracownik_pracownikid )
  REFERENCES pracownik ( pracownikid );

ALTER TABLE zestawy_pr
  ADD CONSTRAINT zestawy_p_operacje_fk FOREIGN KEY ( operacje_operacjaid )
  REFERENCES operacje ( operacjaid );

ALTER TABLE zestawy_pr
  ADD CONSTRAINT zestawy_p_zabieg_fk FOREIGN KEY ( zabieg_zabiegid )
  REFERENCES zabieg ( zabiegid );

ALTER TABLE zestawy_urzadzen
  ADD CONSTRAINT zestawy_urzadzen_operacje_fk FOREIGN KEY (
operacje_operacjaid )
  REFERENCES operacje ( operacjaid );

ALTER TABLE zestawy_urzadzen
  ADD CONSTRAINT zestawy_urzadzen_zabieg_fk FOREIGN KEY ( zabieg_zabiegid )
  REFERENCES zabieg ( zabiegid );

CREATE SEQUENCE zestawy_pr_zestawy_pr_id_seq START WITH 1 NOCACHE
ORDER;

CREATE OR REPLACE TRIGGER zestawy_pr_zestawy_pr_id_trg BEFORE
  INSERT ON zestawy_pr
  FOR EACH ROW
  WHEN ( new.zestawy_pr_id IS NULL )
BEGIN
  :new.zestawy_pr_id := zestawy_pr_zestawy_pr_id_seq.nextval;

```

END;

/

CREATE SEQUENCE zestawy\_urzadzen\_zestawy\_urzad START WITH 1 NOCACHE  
ORDER;

CREATE OR REPLACE TRIGGER zestawy\_urzadzen\_zestawy\_urzad BEFORE  
INSERT ON zestawy\_urzadzen  
FOR EACH ROW  
WHEN ( new.zestawy\_urzadzen\_id IS NULL )

BEGIN

:new.zestawy\_urzadzen\_id := zestawy\_urzadzen\_zestawy\_urzad.nextval;

END;

/

-- Oracle SQL Developer Data Modeler Summary Report:

--

-- CREATE TABLE	12		
-- CREATE INDEX	0		
-- ALTER TABLE	29		
-- CREATE VIEW	0		
-- ALTER VIEW	0		
-- CREATE PACKAGE	0		
-- CREATE PACKAGE BODY		0	
-- CREATE PROCEDURE	0		
-- CREATE FUNCTION	0		
-- CREATE TRIGGER	2		
-- ALTER TRIGGER	0		
-- CREATE COLLECTION TYPE		0	
-- CREATE STRUCTURED TYPE		0	
-- CREATE STRUCTURED TYPE BODY			0
-- CREATE CLUSTER	0		
-- CREATE CONTEXT	0		
-- CREATE DATABASE	0		
-- CREATE DIMENSION	0		
-- CREATE DIRECTORY	0		
-- CREATE DISK GROUP	0		
-- CREATE ROLE	0		
-- CREATE ROLLBACK SEGMENT		0	
-- CREATE SEQUENCE	2		
-- CREATE MATERIALIZED VIEW		0	
-- CREATE MATERIALIZED VIEW LOG			0
-- CREATE SYNONYM	0		
-- CREATE TABLESPACE	0		
-- CREATE USER	0		
--			
-- DROP TABLESPACE	0		
-- DROP DATABASE	0		



```

--
-- REDACTION POLICY          0
-- TSDP POLICY              0
--
-- ORDS DROP SCHEMA          0
-- ORDS ENABLE SCHEMA        0
-- ORDS ENABLE OBJECT        0
--
-- ERRORS                    0
-- WARNINGS                  0

```

### 3.2 Bibliografia

1. Łacheciński, S. (2020). *Składowanie i przetwarzanie danych temporalnych w świetle wymagań standardu języka SQL ISO/IEC 9075*.
2. Rymarski Piotr, K. G. (2021). *Analiza możliwości optymalizacji zapytań SQL*.
3. Wiesław, D. (2006). *Bazy danych SQL: teoria i praktyka*. Helion.