



*WROCŁAWSKA WYŻSZA SZKOŁA
INFORMATYKI STOSOWANEJ*

Sprawozdanie z przedmiotu
Zaawansowane technologie bazodanowe

Sprawozdanie nr 3

*"Utworzenie na serwerze bazy danych
przechowującej informacje na temat danych
pacjentów"*

dr hab inż . Krzysztof Pieczarka
mgr inż . Patrycja Stępień

SEMESTR LETNI 2022/2023

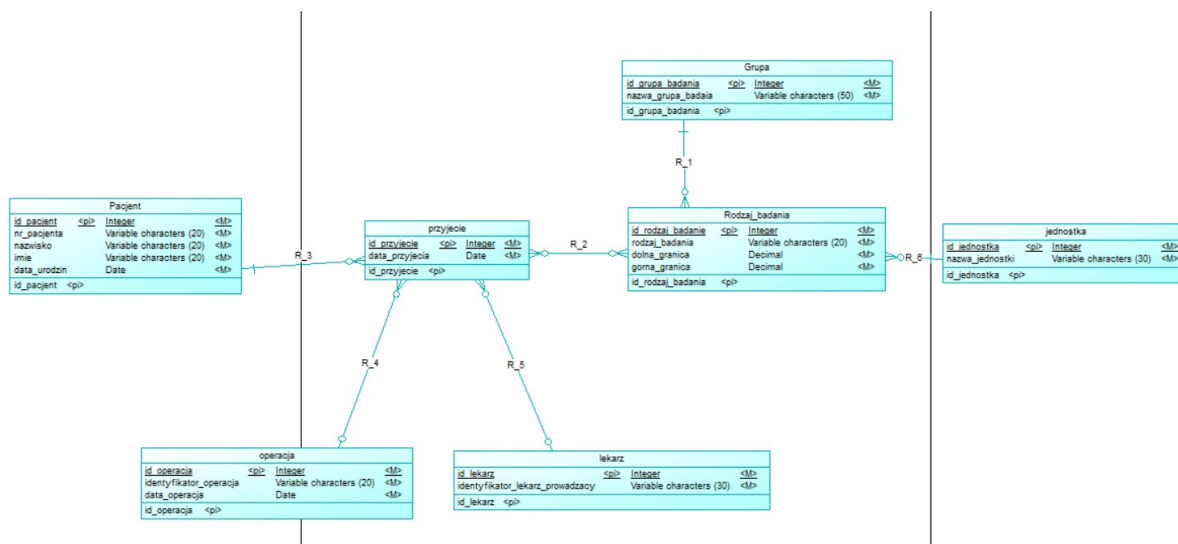
1. Wstęp

Celem niniejszego sprawozdania jest przedstawienie projektu zaawansowanej bazy danych oraz jej implementacji z wykorzystaniem oprogramowania Power Designer oraz SQL Server Management Studio (SSMS). Projektowanie i implementacja zaawansowanej bazy danych to proces, który wymaga sporej wiedzy i doświadczenia z zakresu systemów baz danych. W tym celu, w pierwszej części sprawozdania, zostanie przedstawiona analiza wszelkich wymagań, a następnie projekt bazy danych. W kolejnej części zostanie omówiona implementacja bazy danych w programie Power Designer, wraz z generowaniem skryptów SQL, a także proces ładowania danych. W końcowej części raportu zostaną przedstawione zapytania SQL wykorzystujące różne operacje, takie jak SELECT, INSERT czy ORDER BY, w celu uzyskania odpowiednich informacji z bazy danych. Wszystkie opisane kroki w projekcie i implementacji zostaną wykonane przy użyciu narzędzi Power Designer oraz SQL Server Management Studio.

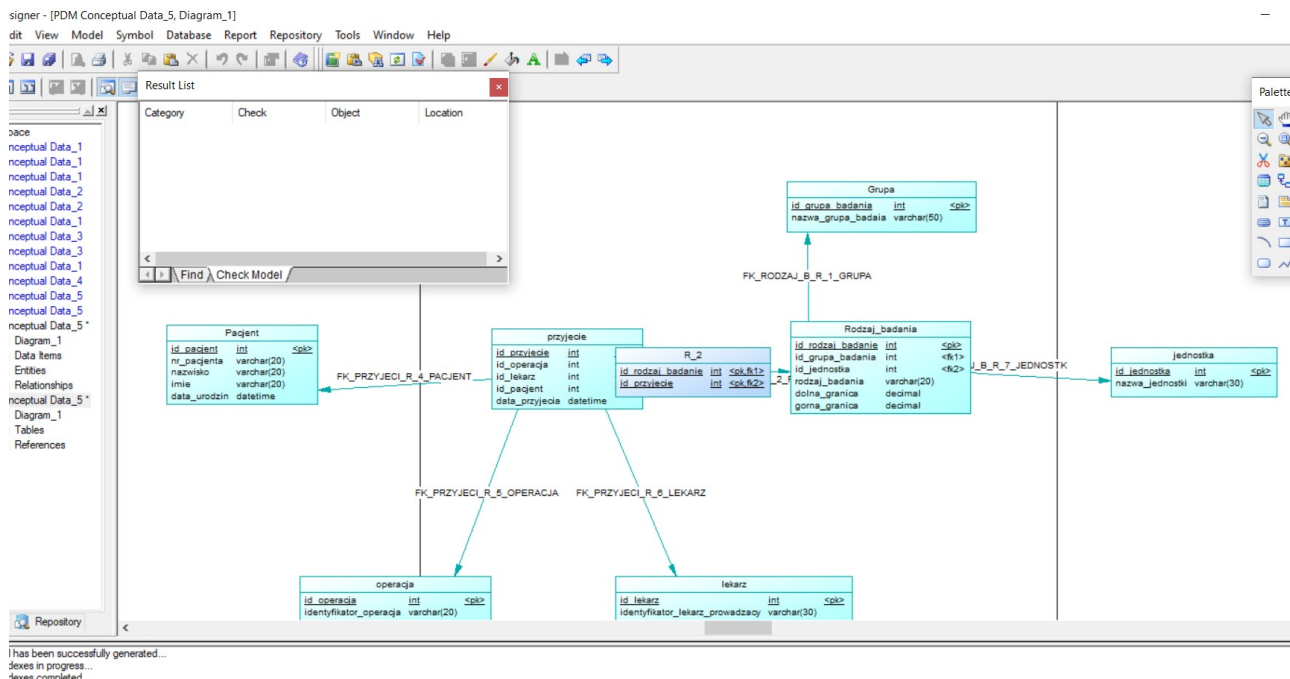
2. Część projektowa

W programie PowerDesigner zaprojektowano diagram konceptualny bazy danych poprzez wykonanie następujących kroków:

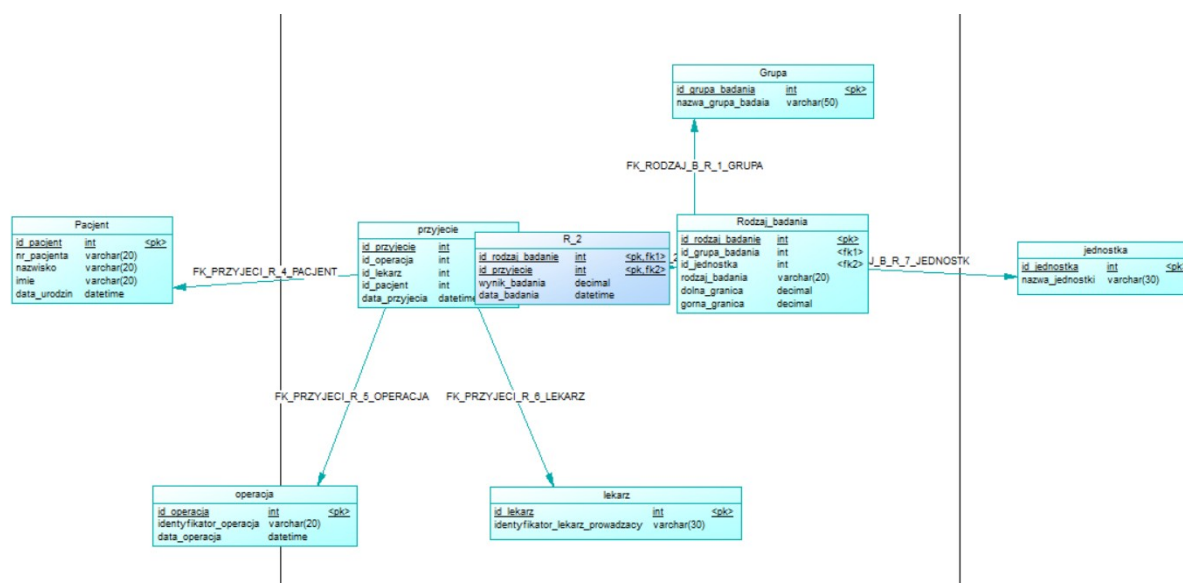
1. Utworzenie nowego modelu danych i wybranie odpowiedniego szablonu projektu.
2. Wybór narzędzia "Diagram konceptualny" z menu "Modelowanie danych".
3. Dodanie encji do diagramu poprzez wybór ikony "Encja" i przeciągnięcie jej na obszar roboczy.
4. Dodanie atrybutów dla każdej encji poprzez wybór ikony "Atrybut" i umieszczenie ich wewnątrz encji.
5. Powiązanie encji ze sobą poprzez wybór ikony "Relacja" i narysowanie linii pomiędzy encjami.
6. Określenie kierunku relacji, rodzaju związku (np. jeden do jednego, jeden do wielu) oraz nazwy relacji.
7. Powtórzenie powyższych kroków dla wszystkich encji i relacji, a następnie zapisanie diagramu konceptualnego.



Rys.2.1.1 Diagram konceptualny określony za pomocą danych wejściowych



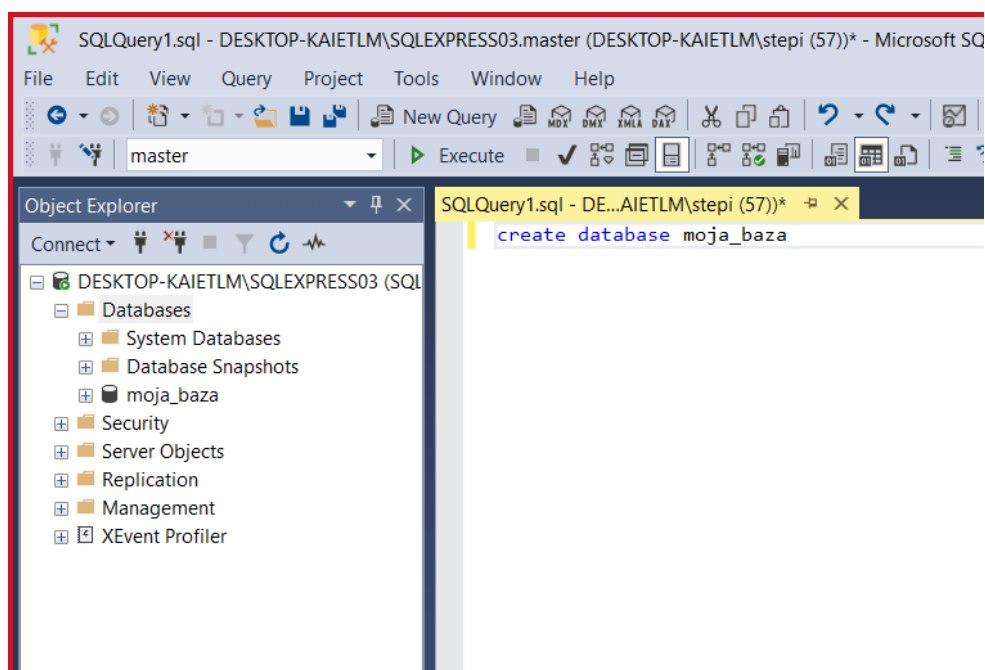
Rys.2.1.2 Generowanie diagramu fizycznego



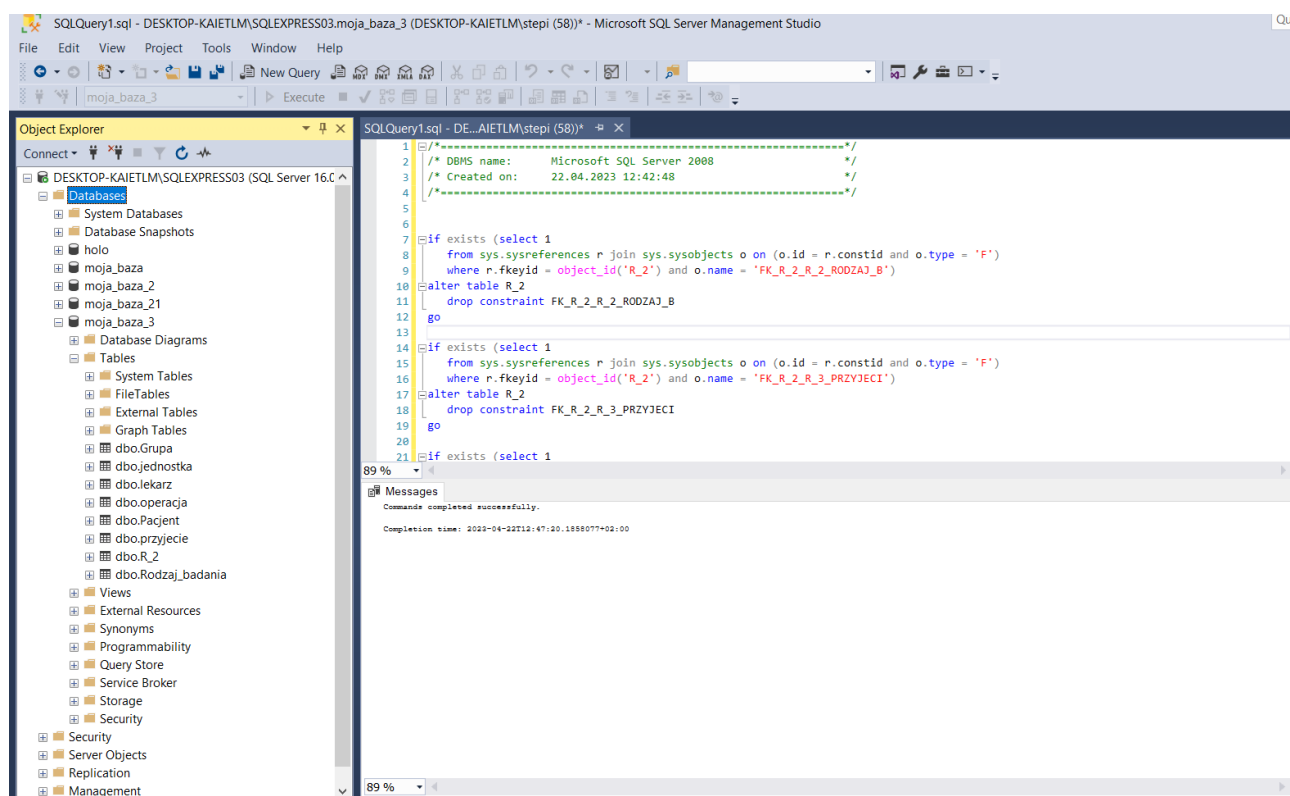
Rys.2.1.3 Diagram fizyczny

W PowerDesigner można również wygenerować diagram fizyczny bazy danych na podstawie diagramu konceptualnego, który zawiera informacje o typach danych.

W oknie "Generate Physical Data Model" wybiera się źródło diagramu konceptualnego i nazwę dla nowego diagramu fizycznego. Następnie klika się przycisk "Generate" i czeka, aż diagram fizyczny zostanie wygenerowany oraz generuje się skrypt SQL

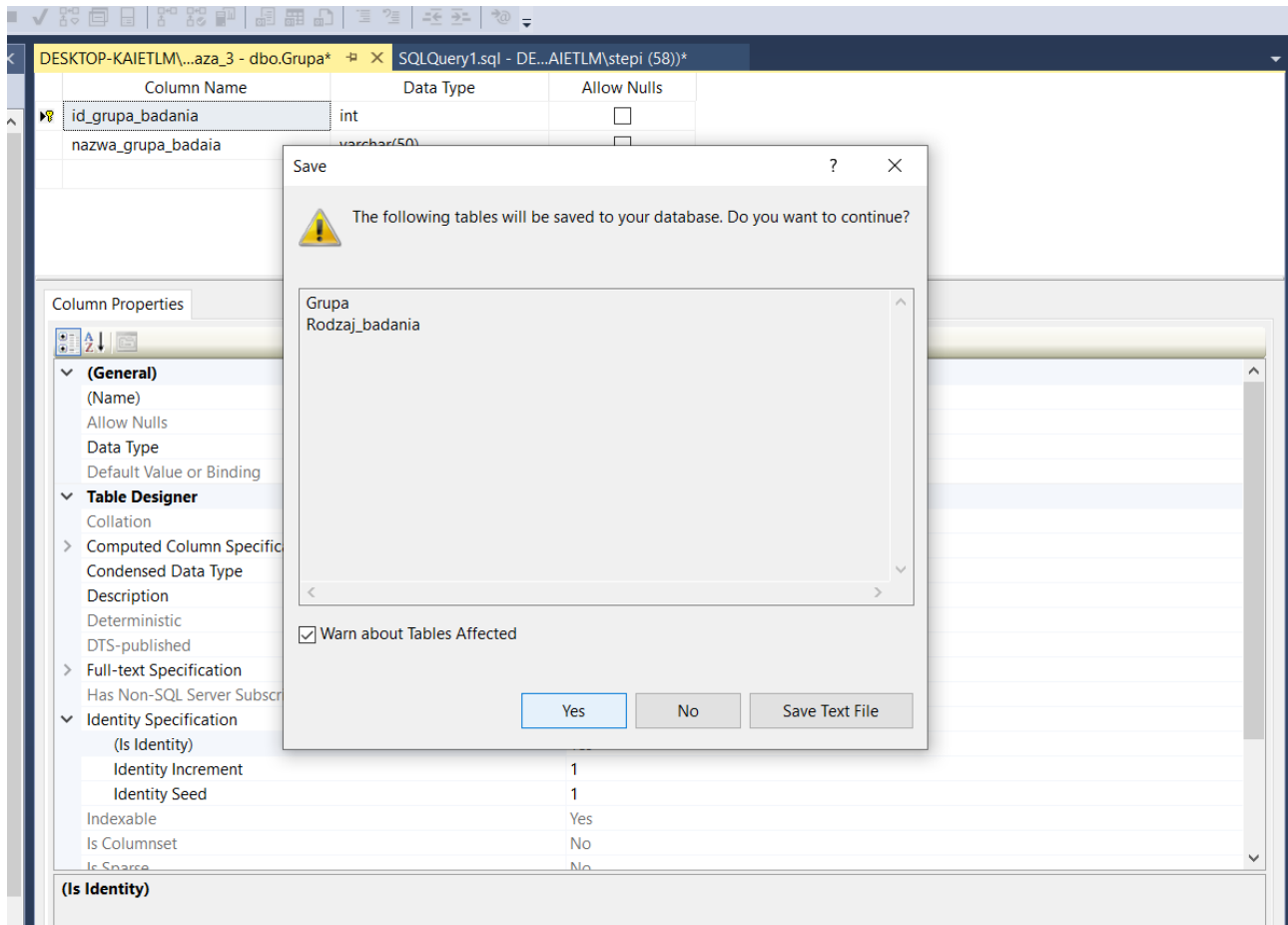


Rys.2.2. Tworzenie nowej bazy danych w systemie zarządzania bazą danych (DBMS)



Rys.2.3. Wklejenie diagramu fizycznego do „mojej bazy” utworzonej w DBMS

Kolejnym krokiem jest określenie czy występuje funkcja „Identity specification” i przypisanie jej do odpowiednich danych z kolumn.



Rys.2.4. Określenie funkcji Identity specification

Aby wprowadzić dane do poszczególnych tabel w języku SQL, należy użyć zapytań INSERT. Ogólna składnia zapytania INSERT wygląda następująco:

INSERT INTO nazwa_tabeli (kolumna1, kolumna2, kolumna3, ...) VALUES (wartość1, wartość2, wartość3, ...);

```

1 delete from R_2
2 delete from przyjecie
3 delete from lekarz
4 delete from operacja
5 delete from Pacjent
6 delete from Rodzaj_badiana
7 delete from jednostka
8 delete from Grupa
9
10
11
12 insert into Grupa values (1,'Hematologia')
13 insert into Grupa values (2,'Immunodiagnostyka')
14 insert into Grupa values (3,'Biochemia')
15 insert into Grupa values (4,'Serologia')
16
17
18 insert into jednostka values (1,'mm/h')
19 insert into jednostka values (2,'ng/ml')
20 insert into jednostka values (3,'pmol/l/s')
21 insert into jednostka values (4,'uderz/min')
22 insert into jednostka values (5,'%')
23
24 insert into Rodzaj_badiana values (1,1,1,'OB',15,20)
25 insert into Rodzaj_badiana values (2,2,2,'TPSA',0,25)
26 insert into Rodzaj_badiana values (3,3,3,'CPK',70,700)
27 insert into Rodzaj_badiana values (4,3,4,'Lipaza',55,55)
28 insert into Rodzaj_badiana values (5,4,5,'ASO',10,20)
29
30
31 insert into Pacjent values (1,'0009','Nowak','Jan','02/10/1938')
32 insert into Pacjent values (2,'01256','Ksiński','Kazimierz','1938/03/15')
33 insert into Pacjent values (3,'56256','Kowalska','Kazimiera','1945/08/28')
34 insert into Pacjent values (4,'0089','Adamiakowa','Franciszka','1954/11/21')
35
36
37 insert into operacja values (1,'1656','2006/09/14')
38 insert into operacja values (2,'2839','2014/08/30')
39 insert into operacja values (3,'1656','2011/12/18')
40
41 insert into lekarz values (1,'3838')
42 insert into lekarz values (2,'1659')
43
44 insert into przyjecie values (1,NULL,NULL,1,'2005/12/12')
45 insert into przyjecie values (2,NULL,NULL,2,'2011/09/09')
46 insert into przyjecie values (3,1,1,1,'2006/09/12')
47 insert into przyjecie values (4,NULL,NULL,3,'2013/02/25')
48 insert into przyjecie values (5,1,2,3,'2014/08/29')
49 insert into przyjecie values (6,2,1,4,'2011/12/16')
50
51
52 insert into R_2 values (1,1,'18','2005/12/14')
53 insert into R_2 values (1,2,'25','2011/09/10')
54 insert into R_2 values (2,3,'38','2006/09/12')
55 insert into R_2 values (3,4,'589','2013/02/27')
56 insert into R_2 values (4,5,'48','2014/08/29')
57 insert into R_2 values (5,6,'16','2011/12/18')
58
59 select * from Grupa
60 select * from jednostka
61 select * from Rodzaj_badiana
62 select * from Pacjent
63 select * from operacja
64 select * from lekarz

```

Results Messages

Rys.2.5. Wprowadzenie danych do kolumny Grupa, jednostka, Rodzaj_badiana, Pacjent, operacja, lekarz, przyjęcie, R_2 oraz użycie funkcji delete i select * from

Przy wprowadzaniu danych do tabeli ważne jest upewnienie się, że wartości wprowadzane do każdej kolumny są zgodne z typem danych określonym dla tej kolumny, a także że wartości dla kolumn wymagających unikalnych wartości (takich jak klucze główne) są rzeczywiście unikalne.

Zapytanie SELECT umożliwia wybranie określonych kolumn z jednej lub kilku tabel, filtrowanie danych, sortowanie wyników i grupowanie wyników w oparciu o określone kolumny.

Results

Messages

	id_grupa_badiana	nazwa_grupa_badaia				
1	1	Hematologia				
2	2	Immunodiagnostyka				
3	3	Biochemia				
4	4	Serologia				

	id_jednostka	nazwa_jednostki				
1	1	mm/h				
2	2	ng/ml				
3	3	pmol/l/s				
4	4	uderz/min				
5	5	%				

	id_rodzaj_badiane	id_grupa_badiana	id_jednostka	rodzaj_badiana	dolna_granica	gorna_granica
1	1	1	1	OB	15	20
2	2	2	2	TPSA	0	25
3	3	3	3	CPK	70	700
4	4	3	4	Lipaza	55	55
5	5	4	5	ASO	10	20

	id_pacjent	nr_pacjenta	nazwisko	imie	data_urodzin	
1	1	0009	Nowak	Jan	1938-02-10 00:00:00.000	
2	2	01256	Iksiński	Kazimierz	1938-03-15 00:00:00.000	
3	3	56256	Kowals...	Kazimiera	1945-08-28 00:00:00.000	
4	4	0089	Adamia...	Francisz...	1954-11-21 00:00:00.000	

	id_operacja	identyfikator_operacja	data_operacja	
1	1	1656	2006-09-14 00:00:00.000	
2	2	2839	2014-08-30 00:00:00.000	
3	3	1656	2011-12-18 00:00:00.000	

	id_lekarz	identyfikator_lekarz_prowadzacy				
1	1	3838				
2	2	1659				

	id_przyjecie	id_operacja	id_lekarz	id_pacjent	data_przyjecia	
1	1	NULL	NULL	1	2005-12-12 00:00:00.000	
2	2	NULL	NULL	2	2011-09-09 00:00:00.000	
3	3	1	1	1	2006-09-12 00:00:00.000	
4	4	NULL	NULL	3	2013-02-25 00:00:00.000	
5	5	1	2	3	2014-08-29 00:00:00.000	

Query executed successfully.

DE

Rys.2.6.1 Wszystkie wyniki dla tabel i Grupa, jednostka, Rodzaj_badania, Pacjent, operacja, przyjecie oraz rodzaj_badania


	id_pacjent	nr_pacjenta	nazwisko	imie	data_urodzin
1	1	0009	Nowak	Jan	1938-02-10 00:00:00.000
2	2	01256	Iksiński	Kazimierz	1938-03-15 00:00:00.000
3	3	56256	Kowals...	Kazimiera	1945-08-28 00:00:00.000
4	4	0089	Adamia...	Francisz...	1954-11-21 00:00:00.000

	id_operacja	identyfikator_operacja	data_operacja
1	1	1656	2006-09-14 00:00:00.000
2	2	2839	2014-08-30 00:00:00.000
3	3	1656	2011-12-18 00:00:00.000

	id_lekarz	identyfikator_lekarz_prowadzacy
1	1	3838
2	2	1659

	id_przyjecie	id_operacja	id_lekarz	id_pacjent	data_przyjecia
1	1	NULL	NULL	1	2005-12-12 00:00:00.000
2	2	NULL	NULL	2	2011-09-09 00:00:00.000
3	3	1	1	1	2006-09-12 00:00:00.000
4	4	NULL	NULL	3	2013-02-25 00:00:00.000
5	5	1	2	3	2014-08-29 00:00:00.000
6	6	2	1	4	2011-12-16 00:00:00.000

	id_rodzaj_badanie	id_przyjecie	wynik_badania	data_badania
1	1	1	18	2005-12-14 00:00:00.000
2	1	2	25	2011-09-10 00:00:00.000
3	2	3	38	2006-09-12 00:00:00.000
4	3	4	589	2013-02-27 00:00:00.000
5	4	5	48	2014-08-29 00:00:00.000
6	5	6	16	2011-12-18 00:00:00.000

 Query executed successfully.

Rys.2.6.2 Wszystkie wyniki dla tabel i Grupa, jednostka, Rodzaj_badania, Pacjent, operacja, przyjecie oraz rodzaj_badania

W końcowym zadaniu użyto funkcji select * (wybierz każde/y) cast ... as ... from ... order by ... asc; Pozwala ona na wybranie konkretnych kolumn i wierszy z tabeli lub zestawu tabel, a także określonych kryteriów. Przedstawione dane zostały ułożone alfabetycznie za pomocą bardziej zaawansowanych funkcji order by.

SQLQuery2.sql - DE...AIETLM\stepi (69))* SQLQuery5.sql - DE...AIETLM\stepi (64))*

```

1  --zapytanie 1
2  SELECT
3      nazwisko + ' ' + imie AS 'Nazwisko i Imie',
4      CAST(data_urodzin AS DATE) AS 'Data urodzenia',
5      nr_pacjenta AS 'Numer pacjenta'
6  FROM
7      Pacjent
8  ORDER BY
9      nazwisko;
10

```

93 %

Results Messages

	Nazwisko i Imie	Data urodzenia	Numer pacjenta
1	Adamiakowa Franciszka	1954-11-21	0089
2	Iksiński Kazimierz	1938-03-15	01256
3	Kowalska Kazimiera	1945-08-28	56256
4	Nowak Jan	1938-02-10	0009

Rys.2.7.1 Utworzenie zapytania select cast...as...from...order by... -zapytanie 1

SQLQuery2.sql - DE...AIETLM\stepi (69))* SQLQuery5.sql - DE...AIETLM\stepi (64))*

```

1  --zapytanie 2
2  SELECT
3      rodzaj_badania.rodzaj_badania AS nazwa_rodzaj_badanie,
4      CAST(rodzaj_badania.dolna_granica AS varchar(10)) + ' - ' +
5      CAST (rodzaj_badania.gorna_granica AS varchar(10))
6      + ' ' + jednostka.nazwa_jednostki AS norma
7  from rodzaj_badania
8  join jednostka on rodzaj_badania.id_jednostka = jednostka.id_jednostka;
9
10

```

93 %

Results Messages

	nazwa_rodzaj_badanie	norma
1	OB	15 - 20 mm/h
2	TPSA	0 - 25 ng/ml
3	CPK	70 - 700 pmol/l/s
4	Lipaza	55 - 55 uderz/min
5	ASO	10 - 20 %

Rys.2.7.2 Utworzenie zapytania select... cast...as...from...order by... -zapytanie 2

SQLQuery2.sql - DE...AIETLM\stepi (69))* SQLQuery5.sql - DE...AIETLM\stepi (64))*

```

1  --zapytanie 3
2  SELECT
3      rodzaj_badian.rodzaj_badian AS nazwa_rodzaj_badiane,
4      CAST(rodzaj_badian.dolna_granica AS varchar(10)) + ' - ' +
5      CAST (rodzaj_badian.gorna_granica AS varchar(10))
6      + ' ' + jednostka.nazwa_jednostki AS norma
7  from rodzaj_badian
8  join jednostka on rodzaj_badian.id_jednostka = jednostka.id_jednostka
9  join Grupa on rodzaj_badian.id_grupa_badian = grupa.id_grupa_badian
10 where grupa.nazwa_grupa_badaia = 'Biochemia'
11

```

93 %

Results Messages

	nazwa_rodzaj_badiane	norma
1	CPK	70 - 700 pmol/l/s
2	Lipaza	55 - 55 uderz/min

Rys.2.7.3 Utworzenie zapytania select... cast...as...from...order by... - zapytanie 3

SQLQuery2.sql - DE...AIETLM\stepi (69))* SQLQuery5.sql - DE...AIETLM\stepi (64))*

```

1  --zapytanie 4
2  SELECT
3      nazwisko AS [nazwisko_pacjent],
4      imie AS imie_pacjent,
5      convert (date, przyjecie.data_przyjecia) as [data_przyjecie]
6  from Pacjent
7  join przyjecie on przyjecie.id_pacjent = pacjent.id_pacjent
8  order by nazwisko_pacjent asc,
9  [data_przyjecie] desc

```

93 %

Results Messages

	nazwisko_pacjent	imie_pacjent	data_przyjecie
1	Adamiakowa	Franciszka	2011-12-16
2	Iksiński	Kazimierz	2011-09-09
3	Kowalska	Kazimiera	2014-08-29
4	Kowalska	Kazimiera	2013-02-25
5	Nowak	Jan	2006-09-12
6	Nowak	Jan	2005-12-12

Rys.2.7.4 Utworzenie zapytania select... cast...as...from...order by... -zapytanie 4

```

1  --zapytanie 5
2  SELECT
3  nazwisko AS [nazwisko_pacjent],
4  imie AS imie_pacjent,
5  convert (date, przyjecie.data_przyjecia) as [data_przyjecie],
6  rodzaj_badiana.rodzaj_badiana as nazwa_rodzaj_badianie,
7  r_2.wynik_badiana,
8  r_2.data_badiana
9  from Pacjent
10 join przyjecie on przyjecie.id_pacjent = pacjent.id_pacjent
11 join r_2 on r_2.id_przyjecie = przyjecie.id_przyjecie
12 join rodzaj_badiana on r_2.id_rodzaj_badianie = rodzaj_badiana.id_rodzaj_badianie
13 order by nazwisko_pacjent asc,
14 [data_przyjecie] desc

```

Rys.2.7.5 Utworzenie zapytania select... cast...as...from...order by... -zapytanie 5

93 %

Results Messages

	nazwisko_pacjent	imie_pacjent	data_przyjecie	nazwa_rodzaj_badianie	wynik_badiana	data_badiana
1	Adamiakowa	Franciszka	2011-12-16	ASO	16	2011-12-18 00:00:00.000
2	Ikściński	Kazimierz	2011-09-09	OB	25	2011-09-10 00:00:00.000
3	Kowalska	Kazimiera	2014-08-29	Lipaza	48	2014-08-29 00:00:00.000
4	Kowalska	Kazimiera	2013-02-25	CPK	589	2013-02-27 00:00:00.000
5	Nowak	Jan	2006-09-12	TPSA	38	2006-09-12 00:00:00.000
6	Nowak	Jan	2005-12-12	OB	18	2005-12-14 00:00:00.000

SQLQuery2.sql - DE...AIETLM\stepi (69))* SQLQuery5.sql - DE...AIETLM\stepi (64))*

```

1  --zapytanie 6
2  SELECT
3  nazwisko AS [nazwisko_pacjent],
4  imie AS imie_pacjent,
5  convert (date, przyjecie.data_przyjecia) as [data_przyjecie],
6  rodzaj_badiana.rodzaj_badiana as nazwa_rodzaj_badianie,
7  r_2.wynik_badiana,
8  r_2.data_badiana
9  from Pacjent
10 join przyjecie on przyjecie.id_pacjent = pacjent.id_pacjent
11 join r_2 on r_2.id_przyjecie = przyjecie.id_przyjecie
12 join rodzaj_badiana on r_2.id_rodzaj_badianie = rodzaj_badiana.id_rodzaj_badianie
13 where wynik_badiana > rodzaj_badiana.dolna_granica
14 and wynik_badiana < rodzaj_badiana.gorna_granica
15 order by nazwisko_pacjent asc,
16 [data_przyjecie] desc

```

Rys.2.7.6 Utworzenie zapytania select... cast...as...from...order by... -zapytanie 6

93 %

Results Messages

	nazwisko_pacjent	imie_pacjent	data_przyjecie	nazwa_rodzaj_badianie	wynik_badiana	data_badiana
1	Adamiakowa	Franciszka	2011-12-16	ASO	16	2011-12-18 00:00:00.000
2	Kowalska	Kazimiera	2013-02-25	CPK	589	2013-02-27 00:00:00.000
3	Nowak	Jan	2005-12-12	OB	18	2005-12-14 00:00:00.000

SQLQuery2.sql - DE...AIETLM\stepi (69))* SQLQuery5.sql - DE...AIETLM\stepi (64))*

```

1  --zapytanie 7
2  SELECT
3      nazwisko AS [nazwisko_pacjent],
4      imie AS imie_pacjent,
5      convert (date, przyjecie.data_przyjecia) as [data_przyjecie],
6      rodzaj_badian.rodzaj_badian as nazwa_rodzaj_badiane,
7      r_2.wynik_badian,
8      r_2.data_badian
9  from Pacjent
10 join przyjecie on przyjecie.id_pacjent = pacjent.id_pacjent
11 join r_2 on r_2.id_przyjecie = przyjecie.id_przyjecie
12 join rodzaj_badian on r_2.id_rodzaj_badiane = rodzaj_badian.id_rodzaj_badiane
13 where pacjent.imie='Kazimiera'
14 and pacjent.nazwisko='Kowalska'
15 order by nazwisko_pacjent asc,
16 [data_przyjecie] desc

```

93 %

Results Messages

	nazwisko_pacjent	imie_pacjent	data_przyjecie	nazwa_rodzaj_badiane	wynik_badian	data_badian
1	Kowalska	Kazimiera	2014-08-29	Lipaza	48	2014-08-29 00:00:00.000
2	Kowalska	Kazimiera	2013-02-25	CPK	589	2013-02-27 00:00:00.000

Rys.2.7.7 Utworzenie zapytania select... cast...as...from...order by... -zapytanie 7

SQLQuery2.sql - DE...AIETLM\stepi (69))* SQLQuery5.sql - DE...AIETLM\stepi (64))*

```

1  --zapytanie 8
2  SELECT pacjent.nr_pacjenta,
3      pacjent.nazwisko + ' ' + pacjent.imie AS [nazwisko i imie pacjenta],
4      convert (date, pacjent.data_urodzin) as [data_urodzenia],
5      convert (date, przyjecie.data_przyjecia) as [data_przyjecie],
6      grupa.nazwa_grupa_badian as nazwa_grupa_badiane,
7      rodzaj_badian.rodzaj_badian as nazwa_rodzaj_badiane,
8      r_2.wynik_badian as wynik,
9      cast(rodzaj_badian.dolna_granica as varchar(10)) + ' ' + cast(rodzaj_badian.gorna_granica as varchar(10))
10 + ' ' + jednostka.nazwa_jednostki as [norma badania],
11      convert (date, r_2.data_badian) as [data],
12      operacja.identyfikator_operacja,
13      operacja.data_operacja,
14      lekarz.identyfikator_lekarz_prowadzacy
15 from Pacjent
16 join przyjecie on przyjecie.id_pacjent = pacjent.id_pacjent
17 join r_2 on r_2.id_przyjecie = przyjecie.id_przyjecie
18 join rodzaj_badian on r_2.id_rodzaj_badiane = rodzaj_badian.id_rodzaj_badiane
19 join grupa on rodzaj_badian.id_grupa_badian = grupa.id_grupa_badian
20 join jednostka on rodzaj_badian.id_jednostka = jednostka.id_jednostka
21 left join operacja on operacja.id_operacja = przyjecie.id_operacja
22 left join lekarz on lekarz.id_lekarz = przyjecie.id_lekarz
23
24

```

70 %

Results Messages

	nr_pacjenta	nazwisko i imie pacjenta	data_urodzenia	data_przyjecie	nazwa_grupa_badiane	nazwa_rodzaj_badiane	wynik	norma badania	data	identyfikator_operacja	data_operacja	identyfikator_lekarz_prowadzacy
1	0009	Nowak Jan	1938-02-10	2005-12-12	Hematologia	OB	18	15 20 mm/h	2005-12-14	NULL	NULL	NULL
2	0009	Nowak Jan	1938-02-10	2006-09-12	Immunodiagnostyka	TPSA	38	0 25 ng/ml	2006-09-12	1656	2006-09-14 00:00:00.000	3838
3	01256	Iksiński Kazimierz	1938-03-15	2011-09-09	Hematologia	OB	25	15 20 mm/h	2011-09-10	NULL	NULL	NULL
4	56256	Kowalska Kazimiera	1945-08-28	2013-02-25	Biochemia	CPK	589	70 700 pmol/l/s	2013-02-27	NULL	NULL	NULL
5	56256	Kowalska Kazimiera	1945-08-28	2014-08-29	Biochemia	Lipaza	48	55 55 uderz/min	2014-08-29	1656	2006-09-14 00:00:00.000	1659
6	0089	Adamiakowa Franciszka	1954-11-21	2011-12-16	Serologia	ASO	16	10 20 %	2011-12-18	2839	2014-08-30 00:00:00.000	3838

Rys.2.7.8 Utworzenie zapytania select... cast...as...from...order by... -zapytanie 8

3. Wnioski

Niniejsze sprawozdanie dotyczy projektowania i tworzenia bazy danych przy wykorzystaniu narzędzi PowerDesigner i SQL Server Management Studio. W ramach pracy wykonano diagram konceptualny oraz fizyczny bazy danych, a także za pomocą skryptów SQL utworzono strukturę bazy oraz wypełniono ją przykładowymi danymi.

W wyniku pracy z narzędziami PowerDesigner i SSMS udało się uzyskać przejrzysty i funkcjonalny schemat. Dzięki temu możliwe jest skuteczne przechowywanie oraz przetwarzanie danych, co może być kluczowe dla wielu dziedzin takich jak medycyna i dane związane z historią przyjęć pacjentów .

Projektowanie i tworzenie bazy danych wymaga czasu oraz wysiłku, ale dzięki narzędziom takim jak PowerDesigner i SSMS, proces ten może być zautomatyzowany. Dzięki temu możliwe jest skuteczne zarządzanie danymi i tworzenie aplikacji, które w pełni wykorzystują potencjał przechowywanych informacji.

Podsumowując, praca ta pokazała, jak ważne jest projektowanie bazy danych oraz jakie korzyści może przynieść to dla różnych dziedzin. Narzędzia takie jak PowerDesigner i SQL Server Management Studio są niezbędne do efektywnego tworzenia i zarządzania bazami danych, co może mieć kluczowe znaczenie dla sukcesu wielu projektów informatycznych.