

Akademia Górniczo-Hutnicza

im. Stanisława Staszica w Krakowie

Wydział Informatyki, Elektroniki i Telekomunikacji

Instytut Informatyki



STUDIA PODYPLOMOWE

SYSTEMY BAZ DANYCH

Projekt dyplomowy

*„Aplikacja bazodanowa do zarządzania zleceniami na badania i ich realizacją
w laboratorium chemicznym”*

Autor: Anna Gomułczak

Opiekun Projektu : dr inż. Robert Marcjan

Kierownik Studiów : dr inż. Anna Zygmunt

Kraków 2023

Spis treści

1. OPIS APLIKACJI	3
2. PRZYPADKI UŻYCIA I SCENARIUSZ	4
2.1. Przypadki użycia	4
2.2. Scenariusze przypadków użycia	5
3. MAKIETY INTERFEJSU POTENCJALNEJ APLIKACJI (UX)	10
3.1. Logowanie	10
3.2. Widok panelu Klienta	12
3.3. Widok panelu kierownika	14
3.4. Widok panelu pracownika	16
4. PROJEKT BAZY DANYCH	17
4.1. Projekt bazy danych	17
4.2. Opis tabel	18
5. OPIS WARSTWY DOSTĘPU DO DANYCH	35
5.1. Widoki	35
5.2. Triggery	41
5.3. Procedury	42
5.4. Funkcje	45
6. OPIS ZAPIIMPLEMENTOWANYCH ELEMENTÓW APLIKACJI	46
6.1. Zaimplementowane elementy HTML	47
6.2. Zaimplementowane elementy kodu Python.	65
7. PODSUMOWANIA, WNIOSKI	68
7.1. Dalsze prace	69
8. LITERATURA	69
9. SKRYPT BAZY DANYCH	70

1. OPIS APLIKACJI.

Aplikacja dotyczy obsługi zamówień na badania oraz rejestracji wyników w laboratorium chemicznym. Interfejs aplikacji po stronie klienta i laboratorium ma różne funkcjonalności.

Przed pierwszym logowaniem klient musi dokonać rejestracji. Po zalogowaniu klient może:

- wysłać zapytanie na badania,
- przeglądać wysłane zapytania na badania,
- przeglądać otrzymane oferty na badanie,
- złożyć zlecenie na badanie,
- przeglądać otrzymane raporty badań
- przeglądać otrzymane faktury.

Kierownik laboratorium dokonuje przeglądu zapytań ofertowych i przygotowuje ofertę dla klienta. Po akceptacji warunków oferty klient generuje zlecenie na badanie. Opisuje w nim dane dotyczące próbki. Kiedy próbka fizycznie dotrze do laboratorium, sekretariat uzupełnia w systemie dane dotyczące próbki które nie są zawarte w zleceniu, takie jak stan próbki i data dostawy.

Kierownik laboratorium dokonując przeglądu zleceń na badanie przydziela pracowników do wykonania odpowiednich badań.

Pracownicy laboratorium dokonując przeglądu zleceń na badania dowiadują się do jakich badań zostali przydzieleni. Wykonują badania, wprowadzają wyniki badań do systemu.

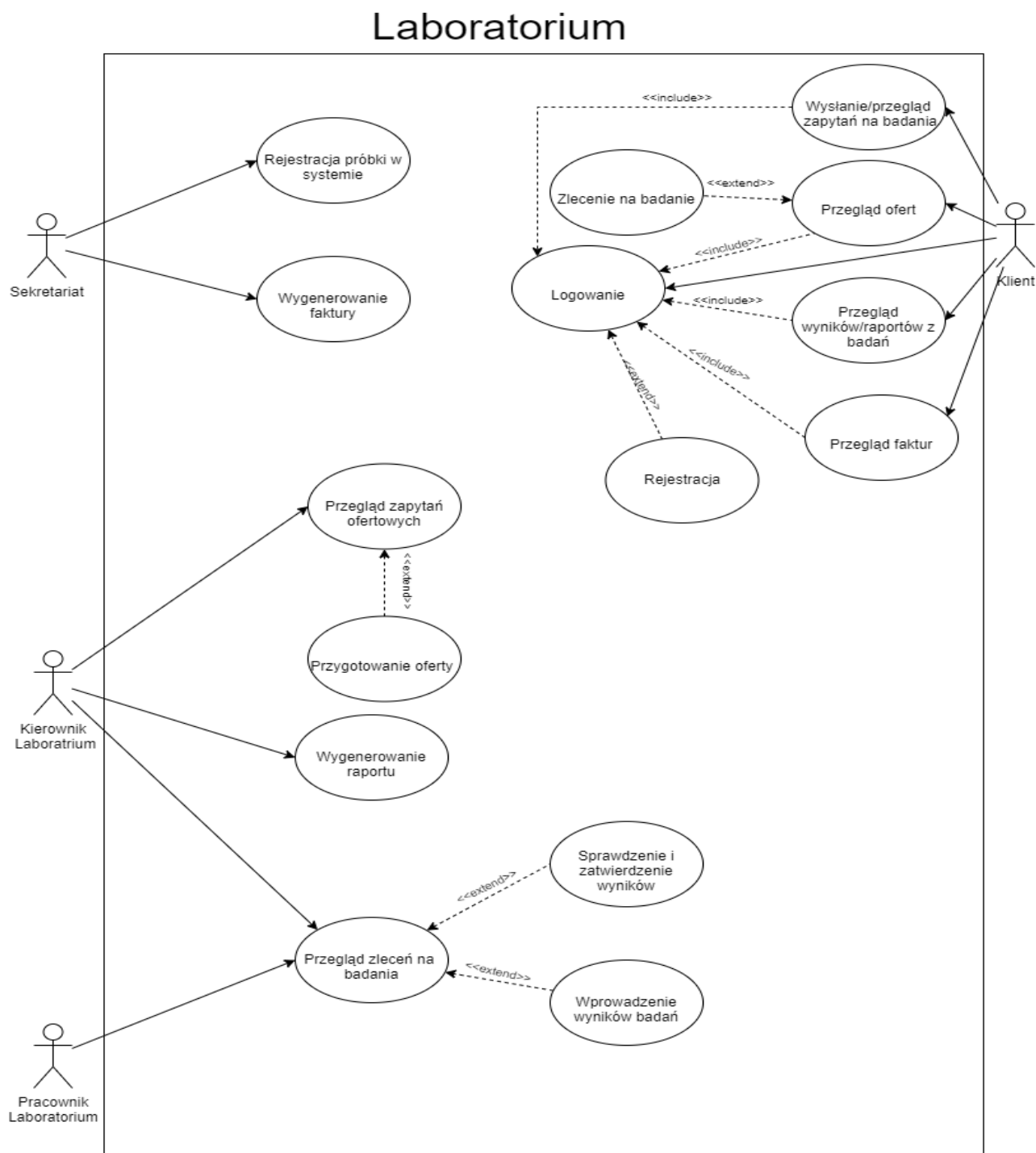
Kierownik laboratorium sprawdza sprawozdanie/sprawozdania z badań i zatwierdza wyniki badań. Generuje raport z badania, który pojawia się na panelu klienta.

Po pojawieniu się raportu z badań w systemie sekretariat generuje elektroniczną fakturę za badania która pojawia się w panelu klienta.

2. PRZYPADKI UŻYCIA I SCENARIUSZ

2.1.Przypadki użycia

Przypadki użycia opisują interakcje pomiędzy aktorami i projektowanym systemem oraz obrazują funkcjonalność systemu. Przypadki użycia opisujące analizowaną aplikację zawarte zostały na diagramie na Rys.1



Rys. 1 Diagram przypadków użycia dla aplikacji do zarządzania zleceniami i wynikami badań w laboratorium

2.2. Scenariusze przypadków użycia

Scenariusze przypadków użycia zawierają sekwencje kroków lub akcji, które przedstawiają, jak użytkownik i system współdziałają w konkretnym przypadku użycia.

W powyższej aplikacji występują następujący aktorzy:

- klient,
- pracownik laboratorium,
- kierownik laboratorium,
- sekretariat .

Przypadek: Logowanie

Aktorzy: Klient

Warunek początkowy: Klient dokonał wcześniej rejestracji w systemie.

Opis:

- 1a. Klient wchodzi na stronę internetową wybiera zakładkę „Zaloguj się” i zostaje przekierowany na stronę logowania.
2. Klient wprowadza swoje dane: login i hasło i zatwierdza je.
3. W przypadku pierwszego logowania Klient dokonuje rejestracji w systemie (PU Rejestracja).

Przypadek: Rejestracja

Aktorzy: Klient

Warunek początkowy: Klient po raz pierwszy składa zapytanie ofertowe do laboratorium.

Opis:

1. Klient znajduje się na stronie do logowania i wybiera opcję „Załącz konto”
2. Klient wprowadza następujące dane:
 - nazwa firmy,
 - NIP
 - adres firmy,
 - adres e-mail,
 - dane osoby do kontaktu,
 - numer telefonu,
 - proponowany login lub domyślnie adres e-mail klienta,
 - hasłoi zatwierdza je przyciskiem „Załącz konto”.
3. Po zatwierdzeniu wszystkich danych klient jest zalogowany do systemu.

Przypadek: Wysłanie/przegląd zapytań na badania

Aktorzy: Klient

Warunek początkowy: Klient jest zalogowany

Opis:

1. Klient w panelu klienta wybiera przycisk „Nowe zapytanie” który przekierowuje go do formularza zapytania ofertowego.
 2. W formularzu wybiera dla każdej ze swoich próbek następujące informacje:
 - interesujące go badanie z listy dostępnych badań dla danej matrycy,
 - matryca próbki z listy dostępnych matryc,
 - nazwa próbki, jego nazwa nadana próbce
 - może również uzupełnić pole „Komentarz” w którym zawrze dodatkowe informacje dla laboratorium.
- Po wyborze naciska przycisk „Dodaj próbkę do zapytania”. Opis próbki trafia do listy „Próbki do badań”, a on może wprowadzić dane dla następnej próbki.
2. Po uzupełnieniu formularza zatwierdza przyciskiem „Wyślij zapytanie”.
 3. Wysłane zapytanie pojawia się na liście „Zapytania oczekujące” w Panelu klienta.

Przypadek: Przegląd ofert

Aktorzy: Klient

Warunek początkowy: Klient jest zalogowany. Klient wysłał zapytanie na interesujące go badanie, kierownik przygotował ofertę na badanie.

Opis:

1. Po wystawieniu przez kierownika laboratorium oferty na badanie pojawia się ona na liście oferty aktualne na Panelu klienta gdzie może on przeglądać otrzymane oferty.
2. Oferty dzielą się na aktualne i historyczne (których data ważności się skończyła) z podziałem na zrealizowane i nie zrealizowane o czym informuje status.
3. Jeżeli klient zdecyduje się na badanie dla którego jest aktualna oferta naciska przycisk „Generuj zlecenie” (PU Zlecenie na badania)

Przypadek: Zlecenie na badanie

Aktorzy: Klient

Warunek początkowy: Klient jest zalogowany. Klient posiada ofertę na badanie i akceptuje jej warunki.

Opis:

1. Klient wybiera na liście aktualnych ofert dla danej oferty przycisk „GENERUJ ZLECENIE”
2. Zatwierdza przyciskiem „zatwierdź”.
3. Pojawia się podgląd zlecenia w którym automatycznie nadany zostaje numer zlecenia różniący się od numeru oferty tym, że ma na początku dodatkowe trzy litery przypisane przez laboratorium dla klienta (symbol klienta).
4. Klient ma możliwość wydruku zlecenia lub zapisu w formacie pdf.

Przypadek: Przegląd wyników/raportów z badań

Aktorzy: Klient

Warunek początkowy: Klient jest zalogowany. Laboratorium wykonało zlecone badanie.

Opis:

1. Po wykonaniu przez laboratorium zlecenia na badanie na koncie klienta pojawia się raport ze zleconych badań.

Raport zawiera informacje o numerze zlecenia, nazwie próbki, dacie zatwierdzenia raportu, wyniki analiz dla badanych próbek, dane pracownika wykonującego badanie i pracownika zatwierdzającego raport.

Przypadek: Przegląd faktur

Aktorzy: Klient

Warunek początkowy: Klient jest zalogowany. Badania zlecone przez klienta zostały wykonane.

Opis:

1. Klient po otrzymaniu raportu z badań otrzymuje na swoim koncie w wersji elektronicznej fakturę za badanie.

Przypadek: Rejestracja próbki w systemie

Aktorzy: Sekretariat

Warunek początkowy: Klient złożył zlecenie na badanie i wysłał próbkę do laboratorium

Opis:

1. W momencie kiedy do laboratorium zostanie dostarczona próbka sekretariat uzupełnia w systemie dane dotyczące próbki – informacje dodatkowe wynikające z opisu próbki (stan próbki, data dostawy)

Przypadek: Przegląd zapytań ofertowych

Aktorzy: Kierownik laboratorium,

Warunek początkowy: Klient wysłał zapytanie na badanie.

Opis:

1. Kierownik przegląda otrzymane zapytania na badania.
2. W ramach przeglądu ofert laboratorium może wykonać:
 - 2.a. przygotowanie oferty (PU Przygotowanie oferty).

Przypadek: Przygotowanie oferty

Aktorzy: Kierownik laboratorium,

Warunek początkowy: Klient wysłał zapytanie na badanie.

Opis:

1. Kierownik laboratorium przygotowuje ofertę w której podaje m.in.:
 - cenę badania,

- datę ważności oferty
- informacje o warunkach transportu
- prośbę o umieszczenie wraz z przesyłką numeru zlecenia lub jego wydrukowanej wersji.

Przypadek: Przegląd zleceń na badania

Aktorzy: Kierownik laboratorium, pracownik laboratorium.

Warunek początkowy: Próbką została dostarczona do laboratorium i wprowadzona do systemu

Opis:

1. Kierownik ma wyświetloną listę wszystkich zleceń (zleceń aktualnych). Wchodząc w dane zlecenie wyświetlają mu się wszystkie badania z danego zlecenia. Kierownik każdemu badaniu przydziela pracownika z rozwijalnej listy pracowników upoważnionych do wykonywania tego badania (Imię i Nazwisko pracownika). Po zatwierdzeniu tej czynności w kolumnie „czas realizacji” automatycznie pojawi się data do której badania ma zostać wykonane
2. Pracownik laboratorium po zalogowaniu ma możliwość przeglądu wszystkich zleceń gdzie widzi między innymi takie istotne dla niego dane, jak: badanie do wykonania, nazwa klienta, czas realizacji, z możliwością edycji danych jedynie w zakresie wprowadzenia wyniku i numeru sprawozdania.
3. Pracownik po wykonaniu badania wprowadza wynik do systemu (PU Wprowadzenie wyników badań)
4. Kierownik po sprawdzeniu sprawozdań z badań zatwierdza wprowadzone wyniki badań (PU Sprawdzenie i zatwierdzenie wyników)

Przypadek: Wprowadzenie wyników badań

Aktorzy: Pracownik laboratorium

Warunek początkowy: Badanie zostało wykonane

Opis:

1. Pracownik laboratorium po wykonaniu badania wchodzi w odpowiednie zlecenie, wybiera badanie, oznacza jako „wykonane”, wprowadza wynik i numer sprawozdania.

Przypadek: Sprawdzenie i zatwierdzenie wyników

Aktorzy: Kierownik laboratorium

Warunek początkowy: Wynik/wyniki badań zostały wprowadzone przez pracownika laboratorium

Opis:

1. Po pojawieniu się wyniku z badania w systemie kierownik laboratorium sprawdza sprawozdanie/sprawozdania z badań i jeżeli nie znajdzie żadnych błędów zatwierdza wynik/wyniki badań w systemie oznaczając jako „zatwierdzone”.

Przypadek: Wygenerowanie raportu

Aktorzy: Kierownik Laboratorium

Warunek początkowy: Zatwierdzono wyniki wszystkich badań znajdujących się na zleceniu

Opis:

1. Kierownik laboratorium po sprawdzeniu czy wszystkie zlecone przez klienta badania zostały wykonane, generuje raport który zawiera niezbędne dla klienta informacje o wykonanych badaniach oraz ich wyniki.
2. Zlecenie uzyskuje status „Zrealizowane” i zostaje przeniesione do zleceń archiwalnych.

Przypadek: Wygenerowanie faktury

Aktorzy: Sekretariat

Warunek początkowy: W systemie pojawił się wygenerowany raport z badania.

Opis:

Sekretariat po sprawdzeniu że w systemie pojawiła się raport z badania generuje fakturę którą zostaje umieszczona w systemie i widoczna na portalu klienta.

3. MAKIETY INTERFEJSU POTENCJALNEJ APLIKACJI (UX)

Aplikacja umożliwia dostęp do szczegółów zleceń i wyników badań klientowi, kierownikowi laboratorium i pracownikom wykonującym badania w różnym zakresie i jest funkcjonalnością dostosowana do każdego z tych użytkowników indywidualnie.

3.1. Logowanie



Rys. 2 Strona którą klient i pracownicy laboratorium widzą po wejściu na stronę główną laboratorium badawczego.



Rys. 3 Strona do logowania lub rejestracji. Wspólna dla klienta, kierownika laboratorium i pracowników.

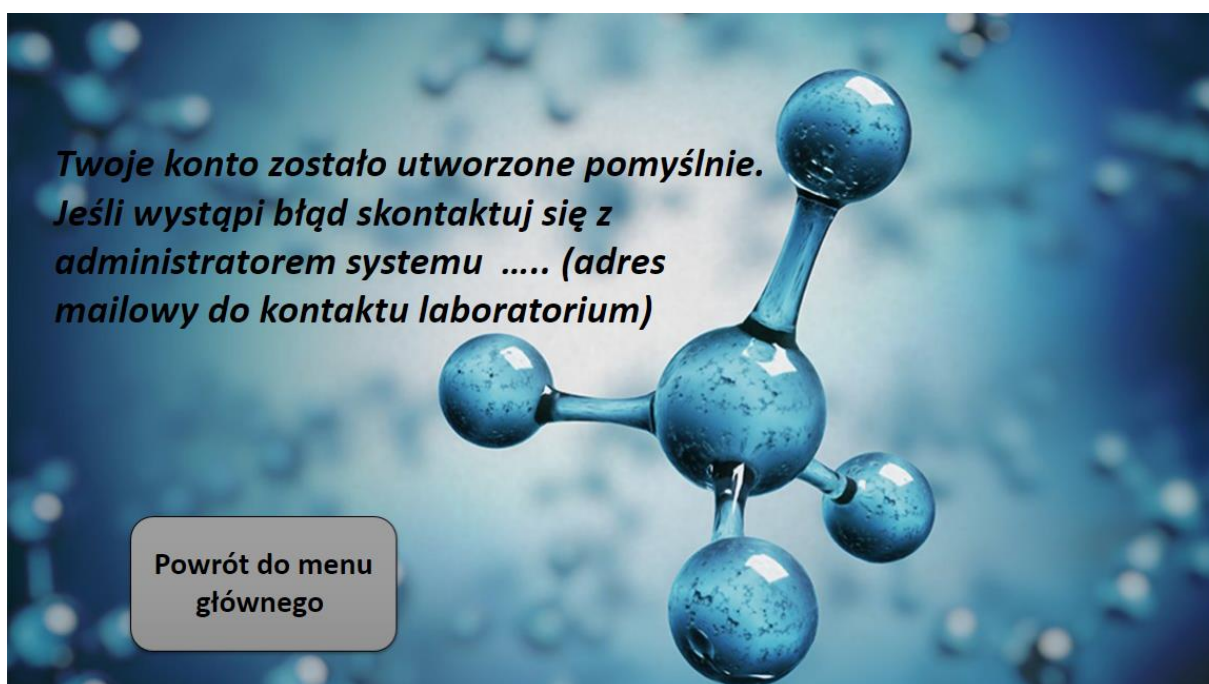
REJESTRACJA

Dane kontaktowe

*pole opcjonalne

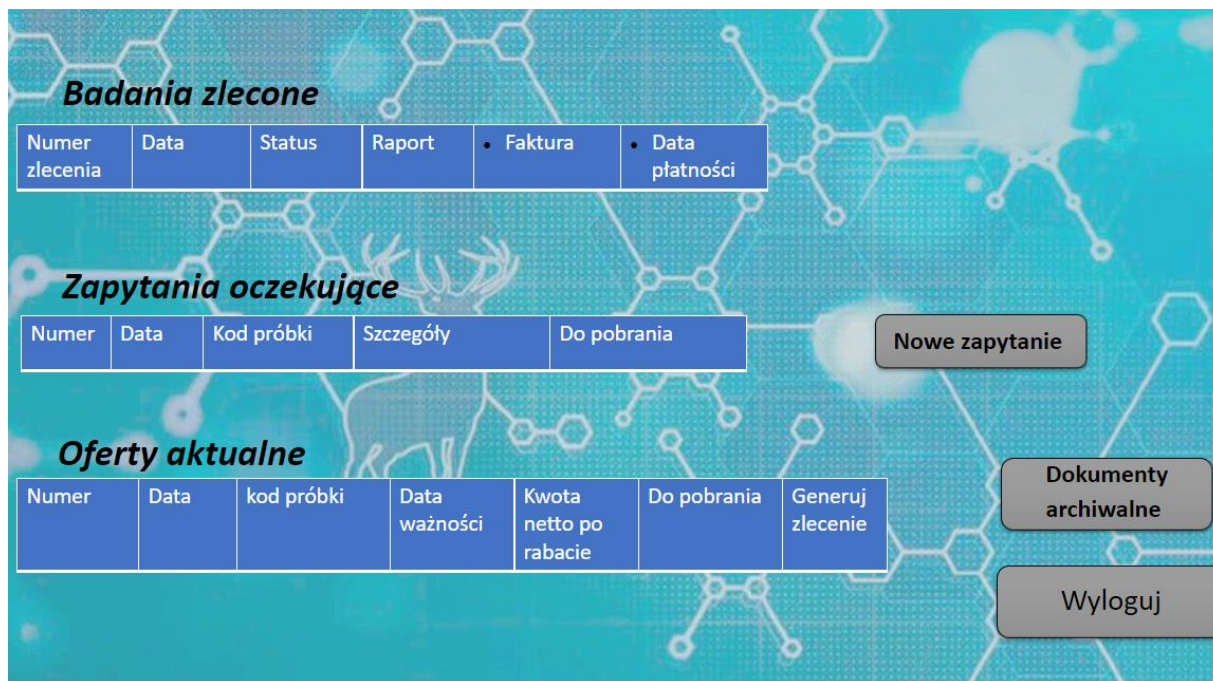
Dane do logowania

Rys. 4 Strona zawierająca dane do rejestracji klienta.



Rys. 5 Strona która pojawi się po założeniu konta.

3.2. Widok panelu Klienta



Rys. 6 Pierwsze okno które pojawia się w aplikacji klienta po zalogowaniu. Klient może przeglądać badania zlecone, zapytania które wysłał, a które oczekują na wycenę oraz oferty aktualne.

Po naciśnięciu przycisku „Nowe zapytanie” klient przeniesiony zostaje do formularza zapytania klienta Rys.7 gdzie generuje zapytanie. Po wysłaniu zapytania trafia ono do tabeli „Zapytania oczekujące”. Po otrzymaniu oferty na badania pole z tabeli „Zapytania oczekujące” znika a pojawia się rekord w tabeli „Oferty aktualne”. Zapytanie zostaje przeniesione do tabeli – „Zapytania archiwalne” Rys.9.

Klient w czasie ważności obowiązywania oferty może dokonać zlecenia na badanie poprzez przyciśnięcie – „Generuj zlecenie”, wtedy rekord zostaje przeniesiony do „Oferty archiwalne” a w tabeli – „Badania zlecone” pojawia się nowy rekord ze statusem – Zlecone. Po okresie ważności oferty, jeżeli klient nie złoży zlecenia na badanie rekord znika a pojawia się w tabeli – „Oferty archiwalne” Rys. 9. Po wykonaniu przez laboratorium badania w kolumnie status – tabela „Badania zlecone” pojawia się status – „Zrealizowane” a klient otrzymuje raport i fakturę do płatności.

Po opłaceniu faktury rekord zostaje przeniesiony do tabeli – „Zlecenia archiwalne” Rys. 9.

Zlecenia archiwalne						
Numer zlecenia	Data	Kod próbki	Status	Raport	Faktura	Data płatności

Zapytania archiwalne				
Numer	Data	Kod próbki	Status	Do pobrania

Oferty archiwalne						
Numer oferty	Data	Kod próbki	Data ważności	Kwota netto	Do pobrania	Status

Powrót

Rys. 9 Okno z tabelami zawierającymi zlecenia, zapytania i oferty archiwalne. Przycisk „Powrót” przenosi klienta do strony głównej.

3.3.Widok panelu kierownika

Przegląd zapytań					
Nazwa klienta	Data zapytania	Numer zapytania	Dane zapytania	Generuj ofertę	Oferta wysłana
Po kliknięciu przekierowuje do danych klienta			Przekierowanie do zapytania klienta	Przycisk/	TAK/NIE

Dokumenty archiwalne

Przegląd zleceń												
Numer klienta	Numer zlecenia	Badanie /matryca	Data realizacji	Data dostawy	Komentarz	Stan próbki	Pracownik	Status	Wynik	Numer sprawozdania	Numer raportu	Numer faktury
					TAK/NIE	Pozytywny/Negatywny						

Wyloguj

Rys. 10 Pierwsza makietą która widzi kierownik po zalogowaniu. W tabeli „Przegląd zapytań” znajdują się wszystkie zapytania jakie zostają przesłane do laboratorium na które kierownik może odpowiedzieć generując ofertę. W tabeli „Przegląd zleceń” kierownik ma wszystkie zlecenia aktualnie realizowane w laboratorium. Po wykonaniu badania, wprowadzeniu wyników

i wygenerowaniu sprawozdania w systemie pojawia się generowana faktura. Po opłaceniu faktury przez klienta rekord zostaje przeniesiony do dokumentów archiwalnych. Dostęp do dokumentów archiwalnych znajduje się pod przyciskiem „Dokumenty archiwalne”.

Przygotowanie oferty

Numer oferty	Data oferty	Data ważności	Data realizacji

Lp	Badanie/matryca	Kwota netto	Rabat	Wat	Kwota brutto

Informacje o warunkach transportu

Generuj ofertę

Powrót

Rys. 11 Na tą stronę kierownik zostaje przeniesiony po naciśnięciu przycisku „generuj ofertę” w tabeli „Przegląd zapytań”. Uzupełnia wszystkie wymagane pola i wysła ofertę do klienta naciskając przycisk „Generuj ofertę”. Kopia oferty trafia do tabeli oferty wysłane w widoku do którego przenosi kierownika przycisk „Dokumenty archiwalne” Rys. 12

Oferty archiwalne

Numer klienta	Numer oferty	Data wysłania	Data ważności	Data realizacji	Badanie/matryca	Kwota netto	Rabat	Kwota brutto	Zlecona	Numer zlecenia
									TAK/NIE	

Zlecenia archiwalne

Numer klienta	Numer zlecenia	Badanie /matryca	Data realizacji	Komentarz	Pracownik	Status	Wynik	Sprawozdanie	Raport	Faktura
				TAK/NIE						

Zapytania archiwalne

Nazwa klienta	Data zapytania	Numer zapytania	Dane zapytania	Status
Po kliknięciu przekierowuje do danych klienta			Przekierowanie do zapytania klienta	TAK/NIE

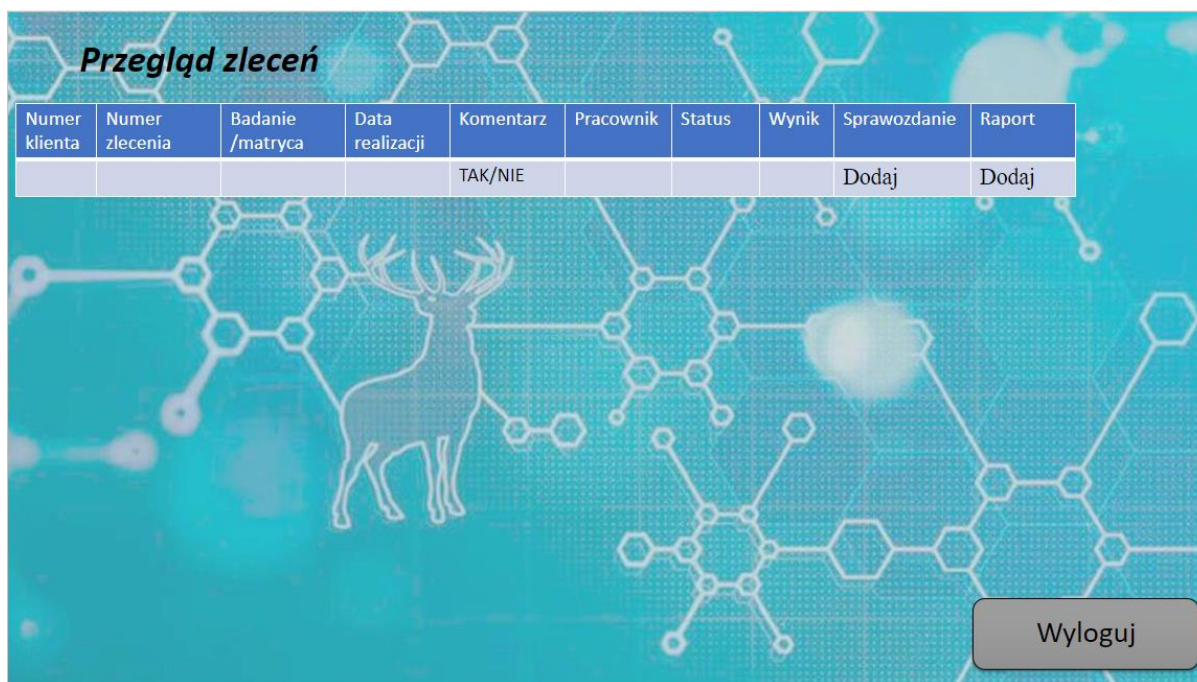
Powrót

Rys. 12 Na tą stronę kierownik zostaje przeniesiony po naciśnięciu przycisku „Dokumenty archiwalne” Rys.10. Zaczynają od góry w tabeli „Wygenerowane oferty” znajdują się wszystkie

oferty wygenerowane przez laboratorium. Jeśli klient zdecyduje się na realizację oferty w kolumnie „Zlecone” zmienia się status z „NIE” na „TAK”. Po opłaceniu przez klienta faktury za badanie, powiązane z nim zlecenie trafia do tabeli „Zlecenia archiwalne”.

„Zapytania archiwalne” znajdują się zapytania na które laboratorium odpowiedziało przygotowując ofertę – status „TAK” i te na które z różnych względów laboratorium nie odpowiedziało status - „NIE”. Przycisk „Powrót” przenosi kierownika na stronę główną Rys. 10.

3.4. Widok panelu pracownika



Przegląd zleceń

Numer klienta	Numer zlecenia	Badanie /matryca	Data realizacji	Komentarz	Pracownik	Status	Wynik	Sprawozdanie	Raport
				TAK/NIE				Dodaj	Dodaj

Wyloguj

Rys. 13 Pierwsza i jedyna strona jaką widzi pracownik laboratorium po zalogowaniu. W tabeli „Przegląd zleceń” pracownik widzi zlecenia do których wykonania został wytypowany. Pracownik aktualizuje status swojego badania, a po wykonaniu badania wpisuje wynik i dołącza sprawozdanie oraz raport z badania.

4. PROJEKT BAZY DANYCH

4.1. Projekt bazy danych

Projekt bazy danych składa się z przedstawiony został na schemacie który zawiera 15 tabel połączonych relacjami jeden do wielu i jeden do jeden (Rys.14)



Rys.

14 Schemat bazy danych i relacje między tabelami.

4.2.Opis tabel

Baza danych zawiera następujące tabele:

- 1) Klienci
- 2) Konta_klient
- 3) Zapytania
- 4) Zlecenia
- 5) Szczegoly_zlecenia
- 6) Wyniki_badania
- 7) Badania_Matryca
- 8) Badania
- 9) Matryca
- 10) Badania_Pracownicy
- 11) Pracownicy
- 12) Konta_pracownicy
- 13) Kraj
- 14) Miasta
- 15) Wojewodztwo

Tabela 1a. Zestawienie atrybutów zawartych w tabeli „Klienci” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_Klienta	nchar(4); NOT NULL; PK	Numer ID Klienta nadawany automatycznie
Nazwa_firmy	nvarchar(40); NULL; UNIQUE	Nazwa firmy
Imie	nvarchar(50); NOT NULL	Imię osoby wyznaczonej do kontaktu
Nazwisko	nvarchar(50); NOT NULL,	Nazwisko osoby wyznaczonej do kontaktu
Adres_ulica	nvarchar(30); NOT NULL	Adres firmy – ulica
Adres_numer_dom u	nvarchar(10); NOT NULL	Adres firmy – numer domu
Adres_numer_loka lu	nvarchar(10); NULL	Adres firmy – numer lokalu
ID_Miejscowosc	int; NOT NULL, FK	ID miejscowości
Kod_pocztowy	char(6); NOT NULL	Kod pocztowy
Telefon	nvarchar(15); NOT NULL; UNIQUE	Numer pocztowy
email	nvarchar(40);	Adres e-mail kontaktowy firmy

	NOT NULL; UNIQUE	
NIP	char(10); NOT NULL; UNIQUE	NIP firmy

```
CREATE TABLE [dbo].[Klienci](
    [ID_Klienta] [nchar](4) NOT NULL,
    [Nazwa_firmy] [nvarchar](40) NULL,
    [Imie] [nvarchar](50) NOT NULL,
    [Nazwisko] [nvarchar](50) NOT NULL,
    [Adres_ulica] [nvarchar](30) NOT NULL,
    [Adres_numer_domu] [nvarchar](10) NOT NULL,
    [Adres_numer_lokalu] [nvarchar](10) NULL,
    [ID_Miejscowosc] [int] NOT NULL,
    [Kod_pocztowy] [char](6) NOT NULL,
    [Telefon] [nvarchar](15) NOT NULL,
    [email] [nvarchar](40) NOT NULL,
    [NIP] [char](10) NULL,
```

```
CONSTRAINT [PK_Klienci] PRIMARY KEY CLUSTERED
```

```
(
```

Tabela 1b. Ograniczenia zawarte w tabeli "Klienci" i ich rola.

Nazwa ograniczenia	kod	Opis
CK_ID_Klienta	((len([ID_Klienta])=8)))	Wymusza długość indeksu klienta na 8
CK_Klienci_email	(([email] like '%_@_%._%'))	Ogranicza możliwość zapisu adresu e-mail do standardowego
CK_Klienci_kod_pocztowy	(([Kod_pocztowy] like '__-__'))	Wymusza zapis kodu w formacie xx-xxx
U_NIP_10	((len([NIP])=10)))	Wymusza wprowadzenie danych o ilości cyfr 10
email_unique		Unikalna nazwa dla adresu email
Nazwa_firmy_unique		Unikalna nazwa dla nazwy firmy
NIP_unique		Unikalny numer dla komórki NIP

100 %

Results Messages

	ID_Klienta	Nazwa_firmy	Imie	Nazwisko	Adres_ulica	Adres_numer_domu	Adres_numer_lokalu	ID_Miejscowosc	Kod_pocztowy	Telefon	email	NIP
1	ASKO	Asza	Jan	Kowalski	Miodowa	1	2	1	30-698	698-879-987	asd@poczta.pl	7895632148
2	BABM	Baba Aga	Joanna	Majer	Radości	1	12	2	00-004	478-852-456	jmajer@interia.pl	4758962106
3	BADN	Bandir i Spółka	Adam	Zapin	Pogodna	2	13	4	53-039	589-897-847	adaza@interi.pl	7859632145
4	BAMA	Baba Jaga	Joanna	Mazur	Jagodowa	13	NULL	1	30-032	658-894-987	jmazur@interia.pl	2354789641
5	COBI	COBRO	Jan	Kaspik	Kłosowa	12	3	1	30-032	658-895-895	jak@poczta.pl	4578968541
6	DOZI	Dom ekologii	Anna	Zima	Długa	1	23	4	53-039	897-989-565	admeko@poczta.pl	8596327482
7	ZANK	NULL	Joanna	Paz	Słoneczna	2	null	2	00-004	458-878-421	paz@interi.pl	4578968574

Rys.15 Zrzut z bazy danych zawartości tabeli „Klienci”.

Tabela 2a. Zestawienie atrybutów zawartych w tabeli „Konta_klient” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_konta	nchar(4); NOT NULL; PK	Numer konta skorelowany relacją 1:1 z ID_Klienta
login	nvarchar(50); NULL; UNIQUE	Login Klienta
haslo	nvarchar(MAX); NOT NULL	Hasło klienta

```
CREATE TABLE [dbo].[Konta_klient](
    [ID_konta] [nchar](4) NOT NULL,
    [login] [nvarchar](max) NOT NULL,
    [haslo] [nvarchar](max) NOT NULL,
    CONSTRAINT [PK_Konta] PRIMARY KEY CLUSTERED
    (
```

	ID_konta	login	haslo
1	ASKO	asko_asz	Yhsye!sdw1
2	BABM	babm_bab	7Gshey2@
3	BADN	badn_dge	89HtwH21!
4	BAMA	bama_dre	90Jehst21!
5	COBI	cobi_get	82Jehatw6!
6	DOZI	dozi_wes	9090JkoO2
7	ZANK	zank_rtsw	HjusGtw23&

Rys.16 Zrzut z bazy danych zawartości tabeli „Konta_klient”.

Tabela 3a. Zestawienie atrybutów zawartych w tabeli „Zapytania” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_Zapytania	int IDENTITY(1,1) NOT NULL; PK	ID zapytania, numer nadawany automatycznie
ID_Klienta	nchar(4); NOT NULL; FK	ID klienta
Data_zapytania	Date; NOT NULL	Data złożenia zapytania
Czas_realizacji_badan	nchar(10); NOT NULL	Czas realizacji badania ustalany przez kierownika na podstawie czasu trwania badania i dostępności laboratorium
Data_przygotowania_oferty	Date; NOT NULL	Data przygotowania oferty
Data_waznosci_oferty	Date; NOT NULL	Data ważności oferty
Status	nchar(10);	Status zapytania. Możliwe wartości: Zlecone,

	NOT NULL	Oczekujące na zlecenie, Wycenione, Złożone
--	----------	--

```
CREATE TABLE [dbo].[Zapytania ](
    [ID_Zapytania] [int] IDENTITY(1,1) NOT NULL,
    [ID_Klienta] [nchar](4) NOT NULL,
    [Data_zapytania] [date] NOT NULL,
    [Czas_realizacji_badan] [nchar](10) NULL,
    [Data_przygotowania_oferty] [date] NULL,
    [Data_waznosci_oferty] [date] NULL,
    [Status] [nchar](12) NULL,
    CONSTRAINT [PK_Zapytania ] PRIMARY KEY CLUSTERED
(
```

Tabela 3b. Ograniczenia zawarte w tabeli “Zapytania” i ich rola.

Nazwa ograniczenia	Kod	Opis
CK_Zapytania_Status	(([Status]='Zlecone' OR [Status] = 'Oczekujące na zlecenia' OR [Status] ='Wycenione' OR [Status]='Złożone'))	Dopuszczalne wartości dla komórki “Status”
CK_Zapytania_data1	([Data_przygotowania_oferty]<[Data_waznosci_oferty] OR [Data_waznosci_oferty] IS NULL)	Data przygotowania oferty musi być wcześniejsza niż data ważności oferty lub przyjąć wartość null.
CK_Zapytania_data2	([Data_waznosci_oferty]>[Data_przygotowania_oferty] OR [Data_waznosci_oferty] IS NULL)	Data ważności oferty musi być późniejsza niż data przygotowania oferty lub przyjąć wartość null.
DF_zapytania_status	DEFAULT ('Złożone') FOR [Status]	Domyślna wartość dla kolumny “Status” - Złożone
DF_Zapytania_Data_zapytania	GETDATE() FOR Data_zapytania	Domyślna data dla kolumny “Data_zapytania” – data bieżąca

ID_Zapytania	ID_Klienta	Data_zapytania	Czas_realizacji_badan	Data_przygotowania_oferty	Data_waznosc_oferty	Status
1	BADN	2023-03-07	5	2023-03-07	2023-03-21	Zrealizowane
2	COBI	2023-03-07	9	2023-03-08	2023-03-22	Zrealizowane
3	ZANK	2023-03-07	2	2023-03-08	2023-03-22	Zrealizowane
4	COBI	2023-03-25	5	2023-03-25	2023-04-10	Zrealizowane
5	ZANK	2023-03-25	9	2023-03-25	2023-04-10	Zrealizowane
6	COBI	2023-08-22	5	2023-08-23	2023-09-06	Zrealizowane
7	ZANK	2023-08-22	11	2023-08-23	2023-09-06	Zrealizowane
8	DOZI	2023-08-22	11	2023-08-23	2023-09-06	Zrealizowane
9	BAMA	2023-08-26	19	2023-08-27	2023-09-10	Oczekujące
10	COBI	2023-08-26	9	2023-08-27	2023-09-10	Oczekujące
11	DOZI	2023-08-27	9	2023-08-27	2023-09-10	Oczekujące
12	COBI	2023-08-28	5	2023-08-28	2023-09-11	Wycenione
14	COBI	2023-08-28	5	2023-08-28	NULL	Wycenione
15	BAMA	2023-08-31	5	NULL	NULL	Złożone

Rys.17 Zrzut z bazy danych zawartości tabeli „Zapytania”.

Tabela 4a. Zestawienie atrybutów zawartych w tabeli „Zlecenia” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_Zlecenia	Int; IDENTITY(1,1) NOT NULL; PK	ID zlecenia
Data_zlecenia	Date; NOT NULL	Data złożenia zlecenia
Stan_probki	nchar(10); NOT NULL	Stan próbki. Możliwe opcje: pozytywny, negatywny.
Data_dostarczenia_probki	Date; NOT NULL	Data dostarczenia próbki
Numer_zlecenia_w_roku	nvarchar(20); NULL	Numer zlecenia w roku. Generowany automatycznie triggerem [tr_kolejny_nr_rok_ZLECENIA]
Data_zatwierdzenia_raportu	Datetime; NOT NULL	Data zatwierdzenia raportu
Data_wystawienia_faktury	Date; NOT NULL	Data wystawienia faktury
Data_platnosci	Date; NOT NULL	Data płatności
Raport_zatwierdzil_Pracownik_ID	Int; NOT NULL	ID pracownika który zatwierdził raport
Status	nchar(10); NOT NULL	Status zlecenia. Możliwe opcje: Zakończzone, Wykonane, W realizacji, Probka dostarczona, Zlecone

```
CREATE TABLE [dbo].[Zlecenia ](
    [ID_Zlecenia] [int] IDENTITY(1,1) NOT NULL,
    [Data_zlecenia] [date] NOT NULL,
    [Stan_probki] [nchar](10) NOT NULL,
    [Data_dostarczenia_probki] [date] NOT NULL,
    [Numer_zlecenia_w_roku] [nvarchar](20) NULL,
```

[Data_zatwierdzenia_raportu] [datetime] NOT NULL,
 [Data_wystawienia_faktury] [date] NOT NULL,
 [Data_platnosci] [date] NOT NULL,
 [Raport_zatwierdzil_Pracownik_ID] [int] NOT NULL,
 [Status] [nchar](10) NOT NULL,

CONSTRAINT [PK_Zlecenia - klient] PRIMARY KEY CLUSTERED

(

Tabela 4b. Ograniczenia zawarte w tabeli "Zlecenia" i ich rola.

Nazwa ograniczenia	kod	Opis
CK_data_dostarczenia	(([Data_dostarczenia_probki]>=[Data_zlecenia]) OR [Data_dostarczenia_probki] IS NULL)	Data dostarczenia próbki musi być późniejsza lub równa dacie zlecenia lub jest NULL
CK_data_faktury	(([data_wystawienia_faktury]>=[Data_zatwierdzenia_raportu]))	Data wystawienia faktury nie może być wcześniejsza niż data zatwierdzenia raportu
CK_data_platności	((datediff(day,[data_wystawienia_faktury],[data_platnosci])>=(14)) OR [Data_platnosci] IS NULL)	Data płatności za badanie nie może być mniejsza niż 14 dni od daty wystawienia faktury lub jest NULL
CK_data_raportu	(([Data_zatwierdzenia_raportu]>=[Data_dostarczenia_probki])OR [Data_zatwierdzenia_raportu] IS NULL))	Data zatwierdzenia raportu nie może być wcześniejsza niż data dostarczenia próbki do laboratorium lub jest równa NULL
CK_data_zlecenia	(([Data_zlecenia]>=getdate()))	Data zlecenia musi być późniejsza lub równa dacie aktualnej
CK_Stan_probki	(([Stan_probki]='negatywny' OR [Stan_probki]='pozytywny') OR [Stan_probki] IS NULL))	Stan próbki może przyjąć wartości „pozytywna”, „negatywna” lub NULL
CK_Zlecenia_Status	(([Status]='Zakończone' OR [Status]='Wykonane' OR [Status]='W realizacji' OR [Status]='Probka dostarczona' OR [Status]='Zlecone'))	Dopuszcza możliwe wartości dla komórki "Status": Zakończone, Wykonane, W realizacji, Próbką dostarczona, Zlecone.
DF_data_dostarczenia	DEFAULT (getdate()) FOR [Data_dostarczenia_probki]	Wartość domyślna dla daty dostarczenia próbki – data bieżąca.
DF_data_faktury	DEFAULT (getdate()) FOR [Data_wystawienia_faktury]	Wartość domyślna dla daty wystawienia faktury – data bieżąca

DF_data_płatności	DEFAULT (getdate()+(14)) FOR [Data_platnosci]	Wartość domyślna dla daty płatności 14 dni od daty bieżącej
DF_data_raportu	DEFAULT (getdate()) FOR [Data_zatwierdzenia_raportu]	Wartość domyślna dla daty raportu – data bieżąca
DF_status	DEFAULT ('Zlecone') FOR [Status]	Wartość domyślna dla statusu zlecenia - Zlecone
DF_zlecenia	DEFAULT (getdate()) FOR [Data_zlecenia]	Wartość domyślna dla daty zlecenia – data bieżąca

	ID_Zlecenia	Data_zlece...	Stan_probki	Data_dosta...	Numer_zle...	Data_zatwi...	Data_wysta...	Data_platn...	Raport_zat...	Status
►	1	2023-03-10	pozytywny	2023-03-13	1/2023	2023-03-17 ...	2023-03-17	2023-04-02	2	Zakończone
	2	2023-03-09	pozytywny	2023-03-14	2/2023	2023-03-24 ...	2023-03-24	2023-04-07	2	Zakończone
	3	2023-03-11	pozytywny	2023-03-15	3/2023	2023-03-25 ...	2023-03-25	2023-04-08	2	Zakończone
	4	2023-04-05	pozytywny	2023-04-07	4/2023	2023-04-12 ...	2023-04-12	2023-04-26	2	Zakończone
	5	2023-04-06	pozytywny	2023-04-07	5/2023	2023-04-12 ...	2023-04-12	2023-04-26	2	Zakończone
	6	2023-08-24	pozytywny	2023-08-25	6/2023	2023-09-04 ...	2023-09-04	2023-09-18	2	Zakończone
	7	2023-08-24	pozytywny	2023-08-25	7/2023	2023-09-04 ...	2023-09-04	2023-09-18	2	Zakończone
	8	2023-08-25	pozytywny	2023-08-26	8/2023	2023-08-29 ...	2023-08-29	2023-09-12	2	Zakończone
	9	2023-08-28	pozytywny	2023-08-29	9/2023	2023-09-04 ...	2023-09-04	2023-09-18	2	Wykonane

Rys.18 Zrzut z bazy danych zawartości tabeli „Zlecenia”.

Tabela 5a. Zestawienie atrybutów zawartych w tabeli „Szczegoly_zlecenia” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_Szczegoly	Int; IDENTITY(1,1); NOT NULL; PK	ID szczegoly
ID_Zapytania	Int; NOT NULL;FK	ID zapytania
ID_Bad_Mat	Int; NOT NULL; FK	ID Bad_Mat
Cena	Money; NULL	Cena badania jaka została wpisana w ofercie na badanie
Rabat	nchar(10); NULL	Rabat nadany klientowi
ID_Pracownika	Int; NOT NULL; FK	ID pracownika wykonującego badanie
Kod_probki	nchar(7); NOT NULL	Kod nadawany próbce przez klienta
Komentarz	Text; NULL	Komentarz klienta

```
CREATE TABLE [dbo].[Szczegoly_zlecenia](
    [ID_Szczegoly] [int] IDENTITY(1,1) NOT NULL,
    [ID_Zapytania] [int] NOT NULL,
```



```

[ID_Bad_Mat] [int] NOT NULL,
[Cena] [money] NULL,
[Rabat] [nchar](10) NULL,
[ID_Pracownika] [int] NULL,
[Kod_probki] [nchar](10) NULL,
[Komentarz] [text] NULL,
CONSTRAINT [PK_Szczegoly_zlecenia] PRIMARY KEY CLUSTERED
(

```

Tabela 5b. Ograniczenia zawarte w tabeli „Szczegoly_zlecenia” i ich rola.

Nazwa ograniczenia	kod	Opis
CK_Jednostka	(([Jednostka]='%' OR [Jednostka]='ng/ml' OR [Jednostka]='mg/ml' OR [Jednostka]='g/kg' OR [Jednostka]='g/L'))	Określa możliwe wartości jednostek otrzymanych wyników

Results Messages		ID_Szczegoly	ID_Zapytania	ID_Bad_Mat	Cena	Rabat	ID_Pracownika	Kod_probki	Komentarz
1		1	1	12	300.00	10	1	GTWE	NULL
2		2	2	8	250.00	5	1	DYRT	NULL
3		3	2	10	250.00	5	3	TRST	NULL
4		4	3	3	170.00	0	4	RTERW	Uwaga na części zielone
5		5	3	6	170.00	0	4	uYTR	Uwaga na opalkowanie
6		6	4	9	250.00	0	3	GrTe	NULL
7		7	5	8	250.00	10	1	hnbgf	NULL
8		8	6	12	300.00	20	1	KLDWM	NULL
9		9	7	7	170.00	0	5	OPVMA	NULL
10		10	7	5	170.00	0	5	CBVZV1	NULL
11		11	8	12	300.00	5	1	KLMR4	NULL
12		12	9	5	170.00	0	4	NMHYB	NULL
13		13	10	8	250.00	5	2	OPLY	NULL
14		14	11	11	300.00	5	4	XCVDS	NULL
15		15	12	1	170.00	5	5	ZZCDE	NULL

Rys.19 Zrzut z bazy danych zawartości tabeli „Szczegoly_zlecenia”.

Tabela 6a. Zestawienie atrybutów zawartych w tabeli „Wyniki_badania” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_Wyniki_badania	Int, PK	ID Wyniki badania
ID_Szczegoly	Int; IDENTITY(1,1); NOT NULL; PK	ID szczegoly
Numer_sprawozdania_w_roku	nvarchar(20); NULL	Kolejny numer sprawozdania w roku nadawany automatycznie triggerem

		tr_kolejny_nr_rok_sprawozdania
Data_wykonania_badiania	Date; NOT NULL	Data wykonania badania
Data_sprawozdania	Date; NOT NULL	Data opracowania sprawozdania
Wynik,	nchar(10); NOT NULL	Wynik badania
Jednostka	nchar(10); NOT NULL	Jednostka badania. Możliwe opcje: g/kg, %, ng/ml, mg/ml, g/L

```
CREATE TABLE [dbo].[Wyniki_badiania](
    [ID_Wyniki_badiania] [int] IDENTITY(1,1) NOT NULL,
    [ID_Szczegoly] [int] NULL,
    [Numer_sprawozdania_w_roku] [nvarchar](20) NULL,
    [Data_wykonania_badiania] [date] NULL,
    [Data_sprawozdania] [date] NULL,
    [Wynik] [nchar](10) NULL,
    [Jednostka] [nchar](10) NULL,
    CONSTRAINT [PK_Wyniki_badiania_1] PRIMARY KEY CLUSTERED
(
```

Tabela 6b. Ograniczenia zawarte w tabeli "Wyniki_badiania" i ich rola.

Nazwa ograniczenia	kod	Opis
CK_Data_sprawozdania	(([Data_sprawozdania]>=[Data_wykonania_badiania]))	Data zatwierdzenia sprawozdanie nie może być wcześniejsza niż data wykonania badania
CK_Jednostka	(([Jednostka]='%' OR [Jednostka]='ng/ml' OR [Jednostka]='mg/ml' OR [Jednostka]='g/kg' OR [Jednostka]='g/L'))	Określa możliwe wartości jednostek otrzymanych wyników

	ID_Wyniki_...	ID_Szczego...	Numer_spr...	Data_wyko...	Data_spra...	Wynik	Jednostka
► 1	1	1	1/2023	2023-03-17	2023-03-17	25	g/kg
2	2	2	2/2023	2023-03-24	2023-03-24	50	g/L
3	3	3	3/2023	2023-03-25	2023-03-25	10	%
4	4	4	4/2023	2023-04-12	2023-04-12	12	%
5	5	5	5/2023	2023-04-12	2023-04-12	50	g/kg
6	6	6	6/2023	2023-09-04	2023-09-04	33	%
7	7	7	7/2023	2023-09-04	2023-09-04	11	mg/ml
8	8	8	8/2023	2023-08-29	2023-08-29	5	g/L
9	9	9	9/2023	2023-09-04	2023-09-04	11	mg/ml

Rys.20 Zrzut z bazy danych zawartości tabeli „Wyniki_badiania”.

Tabela 7. Zestawienie atrybutów zawartych w tabeli „Badania_Matryca” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_Bad_Mat	Int; IDENTITY(1,1); NOT NULL; PK	ID klucz główny tabeli
ID_Matrycy	Int; NOT NULL; FK	ID matrycy
ID_Badania	Int; NOT NULL; FK	ID badania
Warunki_transportu	Text; NOT NULL	Warunki transportu wymagane dla danego rodzaju matrycy

```
CREATE TABLE [dbo].[Badania_Matryca](
    [ID_Bad_Mat] [int] IDENTITY(1,1) NOT NULL,
    [ID_Matrycy] [int] NOT NULL,
    [ID_Badania] [int] NOT NULL,
    [Warunki_transportu] [text] NOT NULL,
    CONSTRAINT [PK_Badania_Matryca] PRIMARY KEY CLUSTERED
    (
```

	ID_Bad_Mat	ID_Matrycy	ID_Badania	Warunki_tr...
► 1	1	1	4	zamrożone
2	1	1	1	zamrożone
3	2	2	2	zamrożone
4	2	2	5	zamrożone
5	2	2	6	zamrożone
6	3	3	3	zamrożone
7	4	5	5	temperatur...
8	5	7	7	temperatur...
9	5	8	8	temperatur...
10	6	8	8	temperatur...
11	7	9	9	lodówka
12	8	9	9	lodówka

Rys.21 Zrzut z bazy danych zawartości tabeli „Badania_Matryca”.

Ad 8a. Zestawienie atrybutów zawartych w tabeli „Badania” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_Badania	Int; IDENTITY(1,1); NOT NULL; PK	Klucz główny tabeli
Nazwa_badania	nvarchar(50); NOT NULL; UNIQUE	Nazwa badania
Cena_badania	Money; NOT NULL	Cena badania. Modyfikowalna w zależności od warunków rynkowych

Czas_realizacji	nvarchar(10); NULL	Minimalny czas niezbędny do realizacji badania przez laboratorium
Ilosc_probki	nchar(10); NULL	Minimalna ilość próbki potrzebna do badań

```
CREATE TABLE [dbo].[Badania](
    [ID_Badania] [int] IDENTITY(1,1) NOT NULL,
    [Nazwa_badania] [nvarchar](50) NOT NULL,
    [Cena_badania] [money] NOT NULL,
    [Czas_realizacji] [nvarchar](10) NULL,
    [Ilosc_probki] [nchar](10) NULL,
    CONSTRAINT [PK_Badania] PRIMARY KEY CLUSTERED
    (
```

Tabela 8b. Ograniczenia zawarte w tabeli "Badania" i ich rola

Nazwa ograniczenia	kod	Opis
CK_Badania_Czas_realizacji	(([Czas_realizacji]>(0)))	Czas realizacji badania musi być większy niż 0 dni
CK_Badnia_Cena	(([Cena_badania]>(0)))	Cena badania musi być większa niż 0
CK_Ilosc_probki	(([Ilosc_probki]>(0)))	Ilość próbki większa niż 0
U_Nazwa_badania	unique	Unikalna nazwa dla badania

	ID_Badania	Nazwa_ba...	Cena_bada...	Czas_realiz...	Ilosc_probki
►	1	Pestycydy G...	170,0000	5	500
	2	Pestycydy G...	170,0000	5	500
	3	Pestycydy G...	170,0000	5	500
	4	Pestycydy L...	170,0000	5	500
	5	Pestycydy L...	170,0000	5	500
	6	Pestycydy L...	170,0000	5	500
	7	Barierowość...	250,0000	9	100
	8	Barierowość...	250,0000	9	100
	9	Zawartość bi...	300,0000	2	200

Rys.22 Zrzut z bazy danych zawartości tabeli „Badania”.

Tabela 9. Zestawienie atrybutów zawartych w tabeli „Matryca” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_Matrycy	Int; IDENTITY(1,1); NOT NULL; PK	Klucz główny tabeli
Nazwa_matrycy	nvarchar(20); NOT NULL; UNIQUE	Nazwa matrycy próbek

```
CREATE TABLE [dbo].[Matryca](
    [ID_Matrycy] [int] IDENTITY(1,1) NOT NULL,
    [Nazwa_matrycy] [nvarchar](20) NOT NULL,
    CONSTRAINT [PK_Matryca] PRIMARY KEY CLUSTERED
```

	ID_Matrycy	Nazwa_ma...
►	5	Folia LDPE
	6	Folia regran...
	7	Mięso
	8	Nabiał
	2	Owoce_duż...
	3	Warzywa_tł...
	1	Wrzywa_du...
	4	Zboża

Rys.23 Zrzut z bazy danych zawartości tabeli „Matryca”.

Tabela 10. Zestawienie atrybutów zawartych w tabeli „Badania_Pracownicy” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_Bad_Prac	Int; IDENTITY(1,1); NOT NULL; PK	Klucz główny tabeli
ID_Badania	Int; NOT NULL; FK	ID badania
ID_Pracownika	Int; NOT NULL; FK	ID Pracownicy

```
CREATE TABLE [dbo].[Badania_Pracownicy](
    [ID_Bad_Prac] [int] IDENTITY(1,1) NOT NULL,
    [ID_Badania] [int] NOT NULL,
    [ID_Pracownika] [int] NOT NULL,
    CONSTRAINT [PK_Badania_Pracownicy] PRIMARY KEY CLUSTERED
(
```

	ID_Bad_Prac	ID_Badania	ID_Pracc
1		7	1
2		9	1
3		1	2
4		2	2
5		3	2
6		4	2
7		5	2
8		6	2
9		7	2
10		8	2
11		9	2
12		7	3
13		8	3
14		1	4
15		2	4
16		3	4
17		4	4
18		9	4
19		1	5
20		5	5
21		6	5

Rys.24 Zrzut z bazy danych zawartości tabeli „Badania_Pracownicy”.

Tabela 11a. Zestawienie atrybutów zawartych w tabeli „Pracownicy” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_Pracownika	int IDENTITY(1,1) NOT NULL; PK	Klucz główny tabeli
Imie	nvarchar(20); NOT NULL	Imię pracownika
Nazwisko	nvarchar(20); NOT NULL	Nazwisko pracownika
Stanowisko	nvarchar(30); NOT NULL	Nazwa stanowiska
Data_zatrudnienia	Date; NOT NULL	Data zatrudnienia
Data_urodzenia	Date; NOT NULL	Data urodzenia
Adres_nazwa_ulicy	nvarchar(20); NOT NULL	Adres zamieszkania pracownika – nazwa ulicy
Adres_numer_domu	nvarchar(10); NOT NULL	Adres zamieszkania pracownika – numer domu

Adres_numer_lokalu	nvarchar(10); NULL	Adres zamieszkania pracownika – numer lokalu
ID_Miejscowosc	Int; NOT NULL	ID miejscowości zamieszkania prcownika
Kod_pocztowy	nvarchar(10); NOT NULL,	Kod pocztowy miejscowości zamieszkania pracownika
Telefon	nvarchar(15); NOT NULL; UNIQUE	Telefon kontaktowy pracownika
email	nvarchar(20); NULL; UNIQUE	Adres e-mail pracownika

```
CREATE TABLE [dbo].[Pracownicy](
    [ID_Pracownika] [int] IDENTITY(1,1) NOT NULL,
    [Imie] [nvarchar](20) NOT NULL,
    [Nazwisko] [nvarchar](20) NOT NULL,
    [Stanowisko] [nvarchar](30) NOT NULL,
    [Data_zatrudnienia] [date] NOT NULL,
    [Data_urodzenia] [date] NOT NULL,
    [Adres_nazwa_ulicy] [nvarchar](20) NOT NULL,
    [Adres_numer_domu] [nvarchar](10) NOT NULL,
    [Adres_numer_lokalu] [nvarchar](10) NULL,
    [ID_Miejscowosc] [int] NOT NULL,
    [Kod_pocztowy] [nvarchar](10) NOT NULL,
    [Telefon] [nvarchar](15) NOT NULL,
    [email] [nvarchar](20) NULL,
    CONSTRAINT [PK_Pracownicy] PRIMARY KEY CLUSTERED
(
```

Tabela 11b. Ograniczenia zawarte w tabeli “Pracownicy” i ich rola

Nazwa ograniczenia	kod	Opis
CK_data_urodzenia	(((datepart(year,getdate()))-datepart(year,[Data_urodzenia]))>=(18)))	Pracownik musi być pełnoletni
CK_Data_zatrudnienia	((([data_zatrudnienia]>[data_urodzenia]))	Data zatrudnienia musi być późniejsza niż data data urodzenia
CK_email	(([email] like '%_@_%._%'))	Ogranicza możliwość zapisu adresu e-mail do standardowego
CK_Kod_pocztowy	(([Kod_pocztowy] like '___-___'))	Wymusza zapis kodu w formacie xx-xxx
CK_Stanowisko	(([Stanowisko]='St_Specjalista' OR [Stanowisko]='Mł_Specjalista' OR	Dopuszcza zapis stanowiska z możliwych dopuszczalnych

	[Stanowisko]='Specjalista' OR [Stanowisko]='Kierownik'))	
U_Tab_Pracownicy	Unique dla kolumn Telefon i email	Unikalne wartości dla atrybutów email i Telefon

	ID_Pracow...	Imie	Nazwisko	Stanowisko	Data_zatru...	Data_urodz...	Adres_naz...	Adres_num...	Adres_num...	ID_Miejsco...	Kod_poczt...	Telefon	email
►	1	Jan	Kowalski	Specjalista	2020-02-20	1978-05-12	Krakowska	7	8	6	37-700	652-895-895	jkow@pod...
	2	Anna	Nowak	Kierownik	2005-02-13	1978-04-22	Wadowicka	10	9	1	30-032	698-784-895	annanow@...
	3	Anna	Jaki	Mł_Specjalis...	2010-01-11	1989-03-22	Łobzowska	1	19	1	30-032	695-895-458	annsw@one...
	4	Karol	Zięba	St_Specjalista	2005-01-20	1981-10-17	Tulipanowa	10	1	4	81-004	124-478-896	krzie@onet....
	5	Joanna	Król	Specjalista	2015-01-30	1991-02-17	Różana	1	15	5	53-039	478-896-985	jkrol@onet.pl
	6	Joanna	Raj	Specjalista	2023-01-01	2000-05-12	Rancka	2	23	4	53-039	458-784-896	jraj@onet.pl

Rys.25 Zrzut z bazy danych zawartości tabeli „Pracownicy”.

Tabela 12. Zestawienie atrybutów zawartych w tabeli „Konta_pracownicy” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_konta	int; NOT NULL; PK	Numer konta skorelowany relacją 1:1 z ID_Pracownika
login	nvarchar(50); UNIQUE	NULL; Login pracownika
hasło	nvarchar(MAX); NOT NULL	Hasło pracownika

```
CREATE TABLE [dbo].[Konta_pracownicy](
    [ID_konta] [int] NOT NULL,
    [login] [nvarchar](50) NOT NULL,
    [haslo] [nvarchar](max) NOT NULL,
    CONSTRAINT [PK_Konta_pracownicy] PRIMARY KEY CLUSTERED
    (
```

	ID_konta	login	haslo
1	1	j_kowalski	Gheri23lj
2	2	a_nowak	HJY23hje10
3	3	a_jaki	Klwkem32!!
4	4	k_zieba	Nhwe34Njwa!
5	5	j_krol	Jdea@#\$S\$jk!
6	6	j_raj	Hjwra1234a

Rys.26 Zrzut z bazy danych zawartości tabeli „Konta_pracownicy”.

Tabela 13a. Zestawienie atrybutów zawartych w tabeli „Kraj” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_Kraj	Int; IDENTITY(1,1);	Klucz główny tabeli

	NOT NULL; PK	
Nazwa_kraju	nvarchar(20); NOT NULL; UNIQUE	Nazwa kraju

```
CREATE TABLE [dbo].[Kraj](
    [ID_Kraj] [int] IDENTITY(1,1) NOT NULL,
    [Nazwa_kraju] [nvarchar](20) NOT NULL,
    CONSTRAINT [PK_Kraj] PRIMARY KEY CLUSTERED
)
```

Tabela 13b. Ograniczenia zawarte w tabeli “ Kraj” i ich rola

Nazwa ograniczenia	kod	Opis
DF_kraj DEFAULT ('Polska') FOR [Nazwa_kraju] GO	DEFAULT ('Polska') FOR [Nazwa_kraju] GO	Domyślna wartość dla kolumny [Nazwa_kraju] - Polska

	ID_Kraj	Nazwa_kraju
►	5	Finlandia
	2	Frakcja
	3	Niemcy
	4	Polska
	6	Szwecja
	1	USA

Rys.27 Zrzut z bazy danych zawartości tabeli „Kraj”.

Tabela 14a. Zestawienie atrybutów zawartych w tabeli „Miasta” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_Miejscowosc	Int; IDENTITY (1,1); NOT NULL	Klucz główny tabeli
Nazwa_miasta	nvarchar(20); NOT NULL; UNIQUE	Nazwa miejscowości
ID_Wojewodztwa	Int; NULL; FK	ID województwa
ID_Kraj	Int; NOT NULL; FK	ID kraju

```
CREATE TABLE [dbo].[Miasta](
    [ID_Miejscowosc] [int] IDENTITY(1,1) NOT NULL,
    [Nazwa_miasta] [nvarchar](20) NOT NULL,
```

```

[ID_Wojewodztwa] [int] NULL,
[ID_Kraj] [int] NOT NULL,
CONSTRAINT [PK_Miasta_1] PRIMARY KEY CLUSTERED
(

```

Tabela 14b. Ograniczenia zawarte w tabeli "Miasta" i ich rola

Nazwa ograniczenia	kod	Opis
U_miasta	Unique	Unikalna wartość dla kolumny [Nazwa_miasta]

	ID_Miejsco...	Nazwa_mia...	ID_Wojewo...	ID_Kraj
►	1	Kraków	1	4
	2	Warszawa	2	4
	3	Gdańsk	3	4
	4	Gdynia	3	4
	5	Wrocław	4	4
	6	Przemyśl	5	4

Rys.28 Zrzut z bazy danych zawartości tabeli „Miasta”.

Tabela 15a. Zestawienie atrybutów zawartych w tabeli „Województwa” oraz ich właściwości i opis.

Atrybut	Właściwości	Opis
ID_Wojewodztwa	Int; IDENTITY(1,1); NOT NULL	Klucz główny tabeli
Nazwa_wojewodztwa	nvarchar(20); NOT NULL; UNIQUE	Nazwa województwa

```

CREATE TABLE [dbo].[Wojewodztwa](
    [ID_Wojewodztwa] [int] IDENTITY(1,1) NOT NULL,
    [Nazwa_wojewodztwa] [nvarchar](20) NOT NULL,
    CONSTRAINT [PK_Województwa_1] PRIMARY KEY CLUSTERED
(

```

Tabela 15b. Ograniczenia zawarte w tabeli "Województwa" i ich rola

Nazwa ograniczenia	kod	Opis
U_województwa	Unique	Unikalna wartość dla kolumny [Nazwa_wojewodztwa]

	ID_Wojewo...	Nazwa_woj...
►	4	Dolnośląskie
	1	Małopolskie
	2	Mazowieckie
	5	Podkarpacki
	3	Pomorskie

Rys.29 Zrzut z bazy danych zawartości tabeli „Województwa”.

5. OPIS WARSTWY DOSTĘPU DO DANYCH

5.1. Widoki

Widoki są wirtualnymi tabelami, które umożliwiają dostęp do wybranych danych z jednej lub wielu tabel bazy danych. Są używane do uproszczenia złożonych zapytań, powtarzanych wielokrotnie zapytań, zapewniają spójność danych i ułatwiają zarządzanie bazą danych.

W pracy widoki zostały skonstruowane w taki sposób, aby zestawiać dane z wielu tabel w widok występujący w aplikacji bądź po stronie klienta bądź pracownika laboratorium.

W pracy zastosowano następujące widoki:

1. PK_Zapytania_oczekujace
2. PK__Zapytania_oczekujace_szczegóły
3. Kwota_netto_po_rabacie
4. PK_Zapytania_archiwum
5. Szczegóły_oferty
6. Zestawienie_oferty_Aktualne
7. Zestawienie_oferty_Archiwalne
8. PK_Badania_Zlecone_Aktualne
9. PK_Badania_Zlecone_Archiwalne
10. Wzór_raportu

Ad 1. PK_Zapytania_oczekujace

Widok pobiera dane do panelu klienta, tabeli zapytania oczekujące

```
CREATE view [dbo].[PK_Zapytania_oczekujace] as
select k.ID_Klienta, ID_Zapytania as 'Numer Zapytania', Data_zapytania, Kod_probki, Status
from [Zapytania ] as z
join Klienci as k on k.ID_Klienta = z.ID_Klienta
where Status = 'Złożone'
```

100 %

Results

Messages

	ID_Klienta	Numer Zapytania	Data_zapytania	Kod_probki	Status
1	BAMA	15	2023-08-31	DSrt4	Złożone

Rys.30 Zrzut generowany przez widok „PK_Zapytania_oczekujace”.

Ad 2. PK__Zapytania_oczekujace_szczegóły

Widok pobiera dane dotyczące rodzaju zleconych badań na wybranych matrycach dla danego zlecenia.

```
Create view [dbo].[PK__Zapytania_oczekujace_szczegóły] as
select sz.ID_Zapytania ,b.Nazwa_badania, m.Nazwa_matrycy,za.Kod_probki ,za.Komentarz
from [Szczegoly_zlecenia] as sz
join Badania_Matryca as bm on bm.ID_Bad_Mat=sz.ID_Bad_Mat
join Badania as b on b.ID_Badania=bm.ID_Badania
join Matryca as m on m.ID_Matrycy=bm.ID_Matrycy
join [Zapytania ] as za on za.ID_Zapytania=sz.ID_Zapytania
```

100 %

Results

Messages

	ID_Zapytania	Nazwa_badania	Nazwa_matrycy	Kod_probki	Komentarz
1	1	Zawartość białka	Nabiał	GHTR	NULL
2	2	Barierowość para	Folia LDPE	gtRWy10	NULL
3	2	Barierowość CO2	Folia regranulat	gtRWy10	NULL
4	3	Pestycydy GC-MS/MS Grupa2	Owoce_duża_zaw_wody	jufrd23	NULL
5	3	Pestycydy GC-MS/MS Grupa3	Warzywa_tłuste	jufrd23	NULL
6	4	Barierowość CO2	Folia LDPE	GHYTR	null
7	5	Barierowość para	Folia LDPE	GHYTR	null
8	6	Zawartość białka	Nabiał	ABC123	null
9	7	Pestycydy LC-MS/MS Grupa2	Zboża	BAR12	null
10	7	Pestycydy LC-MS/MS Grupa3	Owoce_duża_zaw_wody	BAR12	null
11	8	Zawartość białka	Nabiał	Pes34	null
12	9	Pestycydy LC-MS/MS Grupa3	Owoce_duża_zaw_wody	LAB231	null
13	10	Barierowość para	Folia LDPE	ghytt	NULL
14	11	Zawartość białka	Mięso	hgte45	NULL
15	12	Pestycydy LC-MS/MS Grupa1	Wrzywa_duża_zaw_wody	jab234	NULL

Rys.31 Zrzut generowany przez widok „PK_Zapytania_oczekujace_szczegóły”.

Ad 3. Kwota_netto_po_rabacie

Widok generuje wartość zlecenia

```
CREATE view [dbo].[Kwota_netto_po_rabacie] as
select ID_Zapytania, sum(Cena*(100-Rabat)/100) as 'Kwota netto po rabacie'
from [Szczegoly_zlecenia]
group by ID_Zapytania
```

	ID_Zapytania	Kwota netto po rabacie
1	1	270,00
2	2	475,00
3	3	340,00
4	4	250,00
5	5	225,00
6	6	240,00
7	7	340,00
8	8	285,00
9	9	170,00
10	10	237,50
11	11	285,00
12	12	161,50

Rys.32 Zrzut generowany przez widok „Kwota_netto_po_rabacie”.

Ad 4. PK_Zapytania_archiwum

Widok pobiera dane do panelu klienta, tabeli zapytania archiwalne

```
CREATE view [dbo].[PK_Zapytania_archiwum] as
select k.ID_Klienta, ID_Zapytania as 'Numer Zapytania', Data_zapytania, Kod_probki, Status
from [Zapytania ] as z
join Klienci as k on k.ID_Klienta = z.ID_Klienta
where Status <> 'Złożone'
```

100 %

Results Messages					
	ID_Klienta	Numer Zapytania	Data_zapytania	Kod_probki	Status
1	BADN	1	2023-03-07	GHTR	Zrealizowane
2	COBI	2	2023-03-07	gtRWy10	Zrealizowane
3	ZANK	3	2023-03-07	jufrd23	Zrealizowane
4	COBI	4	2023-03-25	GHYTR	Zrealizowane
5	ZANK	5	2023-03-25	GHYTR	Zrealizowane
6	COBI	6	2023-08-22	ABC123	Zrealizowane
7	ZANK	7	2023-08-22	BAR12	Zrealizowane
8	DOZI	8	2023-08-22	Pes34	Zrealizowane
9	BAMA	9	2023-08-26	LAB231	Oczekujące
10	COBI	10	2023-08-26	ghytt	Oczekujące
11	DOZI	11	2023-08-27	hgte45	Oczekujące
12	COBI	12	2023-08-28	jab234	Wycenione
13	COBI	14	2023-08-28	gru234	Wycenione

Rys.33 Zrzut generowany przez widok „PK_Zapytania_archiwum”.

Ad 5. Szczegóły_oferty

Widok pobiera dane widoczne na ofercie

```
CREATE view [dbo].[Szczegóły_oferty] as
select sz.ID_Zapytania as 'Numer oferty', za.Data_przygotowania_oferty, za.Kod_probki,
za.Data_waznosci_oferty, b.Nazwa_badiania, m.Nazwa_matrycy,sz.Cena as 'Cena jednostkowa za
badanie', Rabat, round((Cena*(100-sz.Rabat)/100), 1) as 'Kwota netto po rabacie',
cast(round((Cena*(100-sz.Rabat)/100)* 0.22, 1)AS decimal(10,2)) as 'Wat', cast(round((Cena*(100-
sz.Rabat)/100)* 1.22, 1) AS decimal(10,2)) as 'Kwota brutto'
from [Szczegoly_zlecenia] as sz
join Badania_Matryca as bm on bm.ID_Bad_Mat=sz.ID_Bad_Mat
join Badania as b on b.ID_Badiania = bm.ID_Badiania
join Matryca as m on m.ID_Matrycy=bm.ID_Matrycy
join [Zapytania ] as za on za.ID_Zapytania = sz.ID_Zapytania
```

	Numer oferty	Data_przygotowania_oferty	Kod_probki	Data_waznosci_oferty	Nazwa_badiania	Nazwa_matrycy	Cena jednostkowa za badanie	Rabat	Kwota netto po rabacie	Vat	Kwota brutto
1	1	2023-03-07	GHTR	2023-03-21	Zawartość białka	Nabiał	300,00	10	270,00	59,40	329,40
2	2	2023-03-08	gtRWy10	2023-03-22	Barierowość para	Folia LDPE	250,00	5	237,50	52,30	289,80
3	2	2023-03-08	gtRWy10	2023-03-22	Barierowość CO2	Folia regenerulat	250,00	5	237,50	52,30	289,80
4	3	2023-03-08	jufid23	2023-03-22	Pestycydy GC-MS/MS Grupa2	Owoce_duza_zaw_wody	170,00	0	170,00	37,40	207,40
5	3	2023-03-08	jufid23	2023-03-22	Pestycydy GC-MS/MS Grupa3	Warzywa_tluste	170,00	0	170,00	37,40	207,40
6	4	2023-03-25	GHYTR	2023-04-10	Barierowość CO2	Folia LDPE	250,00	0	250,00	55,00	305,00
7	5	2023-03-25	GHYTR	2023-04-10	Barierowość para	Folia LDPE	250,00	10	225,00	49,50	274,50
8	6	2023-08-23	ABC123	2023-09-06	Zawartość białka	Nabiał	300,00	20	240,00	52,80	292,80
9	7	2023-08-23	BAR12	2023-09-06	Pestycydy LC-MS/MS Grupa2	Zboża	170,00	0	170,00	37,40	207,40
10	7	2023-08-23	BAR12	2023-09-06	Pestycydy LC-MS/MS Grupa3	Owoce_duza_zaw_wody	170,00	0	170,00	37,40	207,40
11	8	2023-08-23	Pes34	2023-09-06	Zawartość białka	Nabiał	300,00	5	285,00	62,70	347,70
12	9	2023-08-27	LAB231	2023-09-10	Pestycydy LC-MS/MS Grupa3	Owoce_duza_zaw_wody	170,00	0	170,00	37,40	207,40
13	10	2023-08-27	ghytt	2023-09-10	Barierowość para	Folia LDPE	250,00	5	237,50	52,30	289,80
14	11	2023-08-27	hgte45	2023-09-10	Zawartość białka	Mięso	300,00	5	285,00	62,70	347,70
15	12	2023-08-28	jab234	2023-09-11	Pestycydy LC-MS/MS Grupa1	Wzrywa_duza_zaw_wody	170,00	5	161,50	35,50	197,00

Rys.34 Zrzut generowany przez widok „Szczegóły_oferty”.

Ad 6. Zestawienie_oferty_Aktualne

Widok pobiera dane widoczne na panelu klienta w tabeli oferty aktualnej

```
CREATE view [dbo].[Zestawienie_oferty_Aktualne] as
select ID_Klienta ,ID_Zapytania, Data_przygotowania_oferty, Kod_probki, Data_waznosci_oferty,
(select [Kwota netto po rabacie] from [dbo].[Kwota_netto_po_rabacie] as kn
where kn.ID_Zapytania = za.Id_Zapytania) as 'Kwota netto po rabacie'
from [Zapytania ] as za
where za.Data_waznosci_oferty >= GETDATE()
```

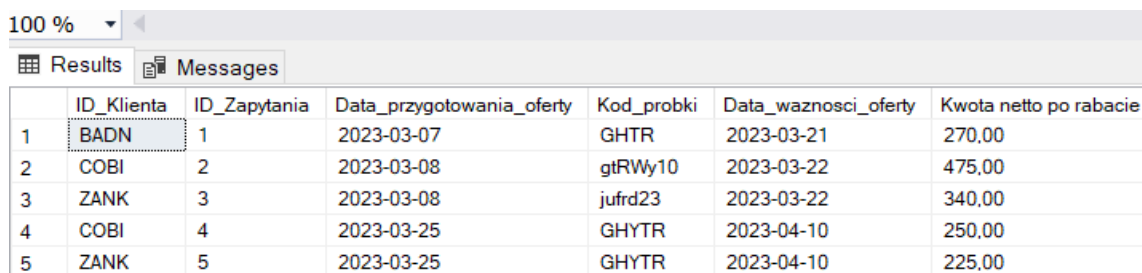
	ID_Klienta	ID_Zapytania	Data_przygotowania_oferty	Kod_probki	Data_waznosci_oferty	Kwota netto po rabacie
1	COBI	6	2023-08-23	ABC123	2023-09-06	240,00
2	ZANK	7	2023-08-23	BAR12	2023-09-06	340,00
3	DOZI	8	2023-08-23	Pes34	2023-09-06	285,00
4	BAMA	9	2023-08-27	LAB231	2023-09-10	170,00
5	COBI	10	2023-08-27	ghytt	2023-09-10	237,50
6	DOZI	11	2023-08-27	hgte45	2023-09-10	285,00
7	COBI	12	2023-08-28	jab234	2023-09-11	161,50

Rys.35 Zrzut generowany przez widok „Zestawienie_oferty_Aktualne”.

Ad 7. Zestawienie_oferty_Archiwalne

Widok pobiera dane widoczne na panelu klienta w tabeli oferty archiwalne

```
create view [dbo].[Zestawienie_oferty_Archiwalne] as
select ID_Klienta ,ID_Zapytania, Data_przygotowania_oferty, Kod_probki, Data_waznosci_oferty,
      (select [Kwota netto po rabacie] from [dbo].[Kwota_netto_po_rabacie] as kn
       where kn.ID_Zapytania = za.Id_Zapytania) as 'Kwota netto po rabacie'
from [Zapytania] as za
where za.Data_waznosci_oferty < GETDATE()
```



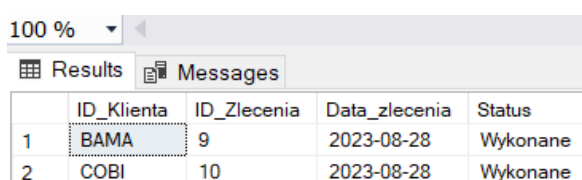
	ID_Klienta	ID_Zapytania	Data_przygotowania_oferty	Kod_probki	Data_waznosci_oferty	Kwota netto po rabacie
1	BADN	1	2023-03-07	GHTR	2023-03-21	270,00
2	COBI	2	2023-03-08	gtRWy10	2023-03-22	475,00
3	ZANK	3	2023-03-08	jufrd23	2023-03-22	340,00
4	COBI	4	2023-03-25	GHYTR	2023-04-10	250,00
5	ZANK	5	2023-03-25	GHYTR	2023-04-10	225,00

Rys.36 Zrzut generowany przez widok „Zestawienie_oferty_Archiwalne”.

Ad 8. PK_Badania_Zlecone_Aktualne

Widok pobiera dane widoczne w tabeli Badania zlecone w panelu klienta

```
CREATE view [dbo].[PK_Badania_Zlecone_Aktualne] as
select za.ID_Klienta,ID_Zlecenia, Data_zlecenia, zl.Status
from [Zlecenia] as zl
join [Zapytania] as za on za.ID_Zapytania=zl.ID_Zlecenia
where zl.Status not in ('Zakończone')
```



	ID_Klienta	ID_Zlecenia	Data_zlecenia	Status
1	BAMA	9	2023-08-28	Wykonane
2	COBI	10	2023-08-28	Wykonane

Rys.37 Zrzut generowany przez widok „Zestawienie_oferty_Archiwalne”.

Ad 9. PK_Badania_Zlecone_Archiwalne

Widok uzupełnia dane w tabeli Badania zlecone archiwalne w panelu klienta

```
CREATE view [dbo].[PK_Badania_Zlecone_Archiwalne] as
select za.ID_Klienta,ID_Zlecenia, Data_zlecenia, zl.Status
from [Zlecenia] as zl
join [Zapytania] as za on za.ID_Zapytania=zl.ID_Zlecenia
where zl.Status in ('Zakończone')
```

100 %

	ID_Klienta	ID_Zlecenia	Data_zlecenia	Status
1	BADN	1	2023-03-10	Zakończone
2	COBI	2	2023-03-09	Zakończone
3	ZANK	3	2023-03-11	Zakończone
4	COBI	4	2023-04-05	Zakończone
5	ZANK	5	2023-04-06	Zakończone
6	COBI	6	2023-08-24	Zakończone
7	ZANK	7	2023-08-24	Zakończone
8	DOZI	8	2023-08-25	Zakończone

Rys.38 Zrzut generowany przez widok „PK_Badania_Zlecone_Archiwalne”.

Ad 10. Wzór_raportu

Widok pobiera dane widoczne w raporcie który otrzymuje klient.

```
CREATE view [dbo].[Wzór_raportu] as
select zl.ID_Zlecenia, concat(zl.ID_Zlecenia,'/',kl.ID_Klienta,'/', year(getdate())) as 'Numer raportu',
za.Kod_probki, b.Nazwa_badania, m.Nazwa_matrycy, concat(w.Wynik,' ',w. Jednostka) as 'Wynik',
w.Data_wykonania_badania as 'Data badania'
from [Zlecenia ] as zl
join [Zapytania ] as za on zl.ID_Zlecenia=za.ID_Zapytania
join Klienci as kl on kl.ID_Klienta=za.ID_Klienta
join [Szczegoly_zlecenia] as sz on sz.ID_Zapytania=za.ID_Zapytania
join Wyniki_badania as w on w.ID_Szczegoly = sz.ID_Szczegoly
join Badania_Matryca as bm on bm.ID_Bad_Mat = sz.ID_Bad_Mat
join Badania as b on b.ID_Badania= bm.ID_Badania
join Matryca as m on m.ID_Matrycy= bm.ID_Matrycy
```

100 %

	ID_Zlecenia	Numer raportu	Kod_probki	Nazwa_badania	Nazwa_matrycy	Wynik	Data badania
1	1	1/BADN/2023	GHTR	Zawartość białka	Nabiał	25 g/kg	2023-03-17
2	2	2/COBI/2023	gtRWy10	Barierowość para	Folia LDPE	50 g/L	2023-03-24
3	2	2/COBI/2023	gtRWy10	Barierowość CO2	Folia regranulat	10 %	2023-03-25
4	3	3/ZANK/2023	jufd23	Pestycydy GC-MS/MS Grupa2	Owoce_duża_zaw_wody	12 %	2023-04-12
5	3	3/ZANK/2023	jufd23	Pestycydy GC-MS/MS Grupa3	Warzywa_tłuste	50 g/kg	2023-04-12
6	4	4/COBI/2023	GHYTR	Barierowość CO2	Folia LDPE	33 %	2023-09-04
7	5	5/ZANK/2023	GHYTR	Barierowość para	Folia LDPE	11 mg/ml	2023-09-04
8	6	6/COBI/2023	ABC123	Zawartość białka	Nabiał	5 g/L	2023-08-29
9	7	7/ZANK/2023	BAR12	Pestycydy LC-MS/MS Grupa2	Zboża	11 mg/ml	2023-09-04

Rys.39 Zrzut generowany przez widok „Wzór_raportu”.

5.2.Triggery

Triggery w bazach danych to specjalne procedury lub reguły, które są aktywowane automatycznie w odpowiedzi na określone operacje na danych, takie jak dodawanie, aktualizacja lub usuwanie rekordów. Służą one do automatyzacji zadań, zapewnienia spójności danych i weryfikacji integralności bazy danych. Triggery pozwalają na wykonywanie określonych działań, jak wysyłanie powiadomień, generowanie logów lub wprowadzanie zmian w innych tabelach, w zależności od warunków określonych przez użytkownika. Są kluczowym narzędziem w utrzymaniu danych w bazie w spójnym stanie i reagowaniu na zdarzenia w czasie rzeczywistym.

W pracy zastosowano następujące triggery:

- 1) tr_kolejny_nr_rok_sprawozdania
- 2) tr_kolejny_nr_rok_ZLECENIA
- 3) trigger_szczegoly_zlecenia

Ad 1. TRIGGE tr_kolejny_nr_rok_sprawozdania

Trigger tworzy w tabeli „Wyniki_badania” kolejny w roku numer sprawozdania w formacie numer sprawozdania/rok.

```
CREATE TRIGGER [dbo].[tr_kolejny_nr_rok_sprawozdania]
ON [dbo].[Wyniki_badania]
AFTER INSERT, DELETE AS
BEGIN
DECLARE @Rok AS INT
SELECT @Rok = YEAR(GETDATE())
DECLARE @Numer AS VARCHAR(10)
SELECT @Numer = CAST(ISNULL(MAX(SUBSTRING(Numer_sprawozdania_w_roku, 1, CHARINDEX('/',
Numer_sprawozdania_w_roku) - 1)), 0) AS INT) + 1
FROM [Wyniki_badania]
WHERE YEAR(Data_sprawozdania) = @Rok
UPDATE [Wyniki_badania]
SET Numer_sprawozdania_w_roku = @Numer + '/' + CAST(@Rok AS VARCHAR(4))
FROM inserted
WHERE [Wyniki_badania].ID_Wyniki_badania = inserted.ID_Wyniki_badania
END
```

Ad 2. TRIGGE tr_kolejny_nr_rok_ZLECENIA

Trigger tworzy w tabeli „Zlecenia” kolejny w roku numer zlecenia w formacie numer zlecenia/rok.

```
CREATE TRIGGER [dbo].[tr_kolejny_nr_rok_ZLECENIA]
ON [dbo].[Zlecenia]
AFTER INSERT, DELETE AS
BEGIN
```

```

DECLARE @Rok AS INT
SELECT @Rok = YEAR(GETDATE())
DECLARE @Numer AS VARCHAR(10)
SELECT @Numer = CAST(ISNULL(MAX(SUBSTRING(Numer_zlecenia_w_roku, 1, CHARINDEX('/',
Numer_zlecenia_w_roku) - 1)), 0) AS INT) + 1
FROM Zlecenia
WHERE YEAR(Data_zlecenia) = @Rok
UPDATE Zlecenia
SET Numer_zlecenia_w_roku = @Numer + '/' + CAST(@Rok AS VARCHAR(4))
FROM inserted
WHERE Zlecenia.ID_Zlecenia = inserted.ID_Zlecenia
END

```

Ad 3. trigger_szczegoly_zlecenia

Trigger pobiera z tabeli „Badania” aktualną cenę za badanie dla zleceń które zostały dopiero złożone i jeszcze nie zostały wycenione.

```

CREATE TRIGGER [dbo].[trigger_szczegoly_zlecenia]
ON [dbo].[Szczegoly_zlecenia]
FOR INSERT AS
BEGIN
    UPDATE [Szczegoly_zlecenia]
    SET Cena = b.Cena_badania
    FROM inserted as a
    JOIN Badania_Matryca as bm ON a.ID_Bad_Mat = bm.ID_Bad_Mat
    JOIN Badania as b ON bm.ID_Badania = b.ID_Badania
    JOIN Zapytania as z ON a.ID_Zapytania = z.ID_Zapytania
    WHERE z.Status = 'Złożone' AND [Szczegoly_zlecenia].ID_Szczegoly = a.ID_Szczegoly;
END

```

5.3.Procedury

Procedury w SQL to zbiory instrukcji i operacji, które można wielokrotnie wywoływać w celu przetwarzania danych w bazie danych. Główne znaczenie procedur polega na umożliwieniu organizacji i uporządkowania operacji na danych, co prowadzi do zwiększenia czytelności i zarządzalności kodu SQL. Procedury mogą przyjmować argumenty, wykonywać operacje na danych, a także zwracać wyniki, choć ich głównym celem jest wykonywanie określonych zadań lub operacji na bazie danych. Różnica między funkcjami a procedurami polega na tym, że procedury zazwyczaj nie zwracają wyników, a jedynie wykonują określone operacje na danych. Procedury są przydatne w sytuacjach, gdy chcemy zachować jednolitość i powtarzalność określonych operacji w bazie danych.

W pracy zastosowano następujące procedury:

1. RejestracjaKlienta,
2. Zapytanie_dane_tymczasowe,
3. Zapytanie

Ad. 1) Procedura RejestracjaKlienta

Procedura wprowadza dane nowego klienta podane podczas rejestracji do tabel „Klienci” i „Konta_klient”

```
CREATE PROCEDURE [dbo].[RejestracjaKlienta]
    @ID_Klienta NCHAR(4),
    @Nazwa_firmy nvarchar(40),
    @Imie NVARCHAR(50),
    @Nazwisko NVARCHAR(50),
    @Adres_ulica NVARCHAR(30),
    @Adres_numer_domu NVARCHAR(10),
    @Adres_numer_lokalu NVARCHAR(10),
    @ID_Miejscowosc int,
    @Kod_pocztowy char(6),
    @Telefon NVARCHAR(15),
    @email NVARCHAR(40),
    @NIP char(10),
    @Login NVARCHAR(50),
    @Haslo NVARCHAR(50)
AS
BEGIN
    INSERT INTO Klienci (ID_Klienta, Nazwa_firmy, Imie, Nazwisko, Adres_ulica, Adres_numer_domu,
    Adres_numer_lokalu, ID_Miejscowosc, Kod_pocztowy, Tlefon, email, NIP)
    VALUES (@ID_Klienta, @Nazwa_firmy, @Imie, @Nazwisko, @Adres_ulica, @Adres_numer_domu,
    @Adres_numer_lokalu, @ID_Miejscowosc, @Kod_pocztowy, @Telefon, @email, @NIP)
    INSERT INTO Konta_klient (ID_konta, login, haslo)
    VALUES (@ID_Klienta, @Login, @Haslo)
END
```

Ad. 2 Procedura Zapytanie_dane_tymczasowe

Procedura tworzy tabelę tymczasową w czasie tworzenia zapytania, aby zapisywać w niej dane dotyczące badania takie jak: rodzaj badania, matryca, kod próbki i komentarz. Powyższe dane wpisywane są w formularz zlecenia w aplikacji widocznej po stronie klienta. Każdy rekord w tabeli tymczasowej to pytanie klienta o jedno badanie.

```
CREATE PROCEDURE [dbo].[Zapytanie_dane_tymczasowe]
    @ID_Klienta nchar(4),
    @Badanie NVARCHAR(50),
```

```

@Matryca NVARCHAR(20),
@KodProbki NCHAR(7),
@Komentarz TEXT
AS
BEGIN
IF OBJECT_ID(N'dbo.TabelaTymczasowa', N'U') IS NULL
    BEGIN
    CREATE TABLE TabelaTymczasowa (
    Numer INT IDENTITY(1, 1) PRIMARY KEY,
    Badanie NVARCHAR(50),
    Matryca NVARCHAR(20),
    KodProbki NCHAR(7),
    Komentarz TEXT
    )
    END
INSERT INTO TabelaTymczasowa (Badanie, Matryca, KodProbki, Komentarz)
VALUES (@Badanie, @Matryca, @KodProbki, @Komentarz)
END

```

Ad. 3) Procedura Zapytanie

Procedura „Zapytanie” jest kontynuacją wcześniej opisanej procedury „Zapytanie_dane_tymczasowe”. Ponieważ w jednym zapytaniu klient może uwzględnić kilka różnych badań powyższa procedura najpierw wprowadza rekord w tabeli „Zapytania” odpowiadający całemu zapytaniu klienta i pobiera ID_Zapytania które następnie będzie wprowadzane do wszystkich rekordów tabeli „Szczegoly_zlecenia” odpowiadających temu zapytaniu. Następnie dla każdego wiersza z tabeli tymczasowej przeszukuje tabele Badania, Matryca i Badania_Matryca aby uzyskać numer ID_Bad_Mat odpowiedni dla wybranej przez klienta kombinacji badanie – matryca. Numer ID_Bad_Mat zostaje wprowadzony do tabeli „Szczegoly_zlecenia” razem z podanym przez klienta kodem próbki i komentarzem oraz odczytanym ID_Zlecenia. Pozostałe dane z tabel „Zapytanie” i „Szczegoly_zlecenia” dostają wartość null i zostaną uzupełnione przez kierownika podczas tworzenia oferty. Po wprowadzeniu danych ten rekord z tabeli tymczasowej zostaje osunięty. Iteracja po tabeli tymczasowej odbywa się do czasu kiedy wszystkie rekordy tabeli zostaną usunięte. Następnie tabela tymczasowa zostaje usunięta. Status zlecenie a w tabeli „Zapytania” zostaje uzupełniony jako „Zlozone” i procedura zostaje zakończona.

```

CREATE PROCEDURE [dbo].[Zapytanie]
@ID_Klienta nchar(4)
AS
BEGIN
    DECLARE @ID_Numer INT
    DECLARE @ID_Bad_Mat INT
    DECLARE @ID_Zapytania INT
    DECLARE @Matryca NVARCHAR(20)
    DECLARE @Badanie NVARCHAR(50)
    DECLARE @KodProbki NCHAR(7)

```

```

    DECLARE @Komentarz NVARCHAR(MAX)
BEGIN TRANSACTION
IF EXISTS (SELECT TOP 1 * FROM TabelaTymczasowa)
BEGIN
    INSERT INTO Zapytania (ID_Klienta, Data_zapytania)
    VALUES (@ID_Klienta, GETDATE())
    SELECT @ID_Zapytania = SCOPE_IDENTITY()
    BEGIN
        WHILE EXISTS (SELECT TOP 1 * FROM TabelaTymczasowa)
        SELECT TOP 1 @ID_Numer = Numer, @Badanie = Badanie, @Matryca = Matryca,
        @KodProbki = KodProbki, @Komentarz = Komentarz
        FROM TabelaTymczasowa
        ORDER BY Numer
        SELECT @ID_Bad_Mat = bm.ID_Bad_Mat
        FROM Badania_Matryca bm
        INNER JOIN Badania b ON bm.ID_Badania = b.ID_Badania
        INNER JOIN Matryca m ON bm.ID_Matrycy = m.ID_Matrycy
        WHERE b.Nazwa_badania = @Badanie AND m.Nazwa_matrycy = @Matryca

        INSERT INTO Szczegoly_zlecenia (ID_Zapytania, ID_Bad_Mat, Cena, Rabat,
        Kod_probki, Komentarz)
        VALUES (@ID_Zapytania, @ID_Bad_Mat, NULL, NULL, @KodProbki, @Komentarz )
        DELETE FROM TabelaTymczasowa WHERE Numer = @ID_Numer
    END
END
COMMIT TRANSACTION
DROP TABLE TabelaTymczasowa
UPDATE Zapytania SET Status = 'Złożone' WHERE ID_Zapytania = @ID_Zapytania
END

```

5.4.Funkcje

Funkcje w SQL to bloki kodu, które wykonują określone operacje na danych lub zwracają konkretne wyniki. Ich główne znaczenie polega na umożliwieniu bardziej zaawansowanych i złożonych operacji na bazach danych. Funkcje mogą przyjmować argumenty, przetwarzać dane oraz zwracać wyniki, co pozwala na tworzenie bardziej elastycznych i wydajnych zapytań SQL.

W pracy zastosowano funkcję „AutoryzacjaUzytkownika”, której kod podano poniżej.

```

CREATE FUNCTION [dbo].[AutoryzacjaUzytkownika]
    (@Login NVARCHAR(50),
    @Haslo NVARCHAR(50))
RETURNS NVARCHAR(50)
AS
BEGIN

```

```

DECLARE @Rola NVARCHAR(50)
SELECT @Rola = 'Kierownik'
FROM Pracownicy p
INNER JOIN Konta_pracownicy kp ON p.ID_Pracownika = kp.ID_konta
WHERE p.Stanowisko = 'Kierownik' AND kp.login = @Login AND kp.haslo = @Haslo
IF @Rola IS NOT NULL
BEGIN
    RETURN @Rola
END
SELECT @Rola = 'Pracownik'
FROM Pracownicy p
INNER JOIN Konta_pracownicy kp ON p.ID_Pracownika = kp.ID_Konta
WHERE p.Stanowisko <> 'Kierownik' AND kp.login = @Login AND kp.haslo = @Haslo
IF @Rola IS NOT NULL
BEGIN
    RETURN @Rola
END

SELECT @Rola = 'Klient'
FROM Konta_klient kk
WHERE kk.login = @Login AND kk.haslo = @Haslo
IF @Rola IS NOT NULL
BEGIN
    RETURN @Rola
END
RETURN 'Brak konta'
END

```

6. OPIS ZAIMPLEMENTOWANYCH ELEMENTÓW APLIKACJI

W niniejszym rozdziale przedstawiono szczegółowy opis elementów zaimplementowanych w aplikacji będącej tematem pracy dyplomowej tj. języka HTML – języka znaczników używanego do tworzenia struktury i zawartości stron internetowych oraz Pythona - wszechstronnego języka programowania wykorzystywanego do obsługi logiki biznesowej aplikacji i interakcji z użytkownikami.

W powyższym projekcie HTML pełni rolę struktury, dzięki której definiuje się jak aplikacja będzie wyglądała w przeglądarkach internetowych. Python z kolei jest odpowiedzialny za obsługę żądań użytkowników, przetwarzanie danych i generowanie dynamicznej zawartości na podstawie logiki biznesowej.

6.1.Zaimplementowane elementy HTML

W projekcie zdefiniowano następujące style używane następnie w kodzie tworzącym kolejne strony aplikacji.

```
.btn {  
  color: black;  
  text-align: center;  
  background-color: rgb(173, 125, 125);  
  padding: 20px;  
  border-radius: 5px;  
  text-decoration: none ;  
}  
.left-column {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  align-items: flex-start;  
  width: 70%;  
  padding: 20px;  
  margin-left: 100px;  
}  
.left-column1 {  
  display: flex;  
  flex-direction: column;  
  justify-content: flex-start;  
  align-items: flex-start;  
  width: 100%; /* Zmieniamy szerokość na 100% */  
  padding: 20px;  
  margin-left: 10px;  
  margin-top: 0;  
}  
.left-column2 {  
  display: flex;  
  flex-direction: column;  
  justify-content: flex-start;  
  align-items: flex-start;  
  width: 50%; /* Zmieniamy szerokość na 100% */  
  padding: 20px;  
  margin-left: 10px;  
  margin-top: 0;  
}  
.left-column3 {  
  display: flex;  
  flex-direction: column;  
  justify-content: flex-start;  
  align-items: flex-start;  
  width: 50%; /* Zmieniamy szerokość na 100% */
```

```
padding: 20px;
margin-left: 10px;
margin-top: 0;
}
.right-column {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: flex-start;
  width: 30%;
  padding: 20px;
  margin-right: 20px;
}
.right-column1 {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: flex-start;
  width: 50%;
  padding: 20px;
  margin-right: 20px;
}
.right-column2 {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: flex-start;
  width: 50%;
  padding: 20px;
  margin-right: 20px;
  margin-top: 20px;
}
h1 {
  font-size: 36px;
  color: black;
}
h2 {
  font-size: 32px;
  color: black;
}
h3 {
  font-size: 28px;
  color: black;
}
.btn1 {
  display: block;
  font-size: 24px;
  color: black;
```



```

    text-align: left;
    margin-top: 40px;
    text-decoration: none;
    background-color: gray;
    padding: 10px 20px;
    border-radius: 5px;
}
.btn2 {
    display: block;
    font-size: 30px;
    color: black;
    text-align: left;
    margin-top: 20px;
    text-decoration: none;
    background-color: gray;
    padding: 20px 30px;
    border-radius: 5px;
}
.btn3 {
    display: block;
    font-size: 20px;
    color: black;
    text-align: left;
    margin-top: 20px;
    text-decoration: none;
    background-color: gray;
    padding: 20px 30px;
    border-radius: 5px;
font-weight: bold;
}
form {
    display: flex;
    flex-direction: column;
    align-items: center;
}
label {
    font-size: 20px;
}
input[type="text"],
input[type="password"] {
    width: 300px;
    padding: 10px;
    margin-bottom: 15px;
    font-size: 16px;
}
.registration-form input[type="text"],
.registration-form input[type="password"] {
    width: 300px;

```

```

padding: 5px;
margin-bottom: 2px;
font-size: 12px;
}
.registration-form label {
font-size: 16px;
font-weight: bold;
color: white
}
input[type="submit"] {
width: 150px;
padding: 10px;
background-color: gray;
color: white;
font-size: 18px;
border: none;
cursor: pointer;
}
input[type="submit"]:hover {
background-color: darkgray;
}
font-weight: bold;
}
.tabela-container {
display: flex;
justify-content: flex-start;
align-items: flex-start;
flex-wrap: wrap;
}
.tabela {
width: 30%;
margin-right: 20px;
}
table {
border-collapse: collapse;
width: 100%;
}
.custom-table, .custom-table th, .custom-table td {
border: 1px solid black;
font-size: 18px;
padding: 20px;
background-color: lightgray;
}

```

Ogólny szablon stron internetowych do którego odwołuje się każda strona:

```
<html lang="en">
<head>
<title>{{title}}</title>
<link rel="stylesheet" href="/static/style.css">
</head>
<body>

    {% block content %}

    {% endblock %}

</body>
</html>
```

Pierwsza strona aplikacji

```
{% extends "szablon.html" %}
{% block content %}
<style>
body {
    background-image: url("/static/tlo.jpg");
    background-size: cover;
    display: flex;
    flex-direction: row;
    justify-content: space-between;
    align-items: center;
    height: 100vh;
    margin: 0;
    padding: 0;
}
</style>
<div class="left-column">
    <h1>Laboratorium Badawczo - Rozwojowe</h1>
    <br><br><br><br>
    <h1>A&M jakość i doświadczenie</h1>
</div>
<div class="right-column">
    <a href="/o_nas" class="btn1">O nas</a>
    <a href="/oferta_badan" class="btn1">Oferta badań</a>
    <a href="/kontakt" class="btn1">Kontakt</a>
    <a href="/rejestracja" class="btn1">Zarejestruj się</a>
    <a href="/logowanie" class="btn1">Zaloguj się</a>
</div>
{% endblock %}
```



Rys.39 Zrzut strony głównej aplikacji

Strona rejestracji nowego klienta:

```
{% extends "szablon.html" %}
{% block content %}
<style>
  body {
    background-image: url("/static/tlo.jpg");
    background-size: cover;
    display: flex;
    flex-direction: row;
    justify-content: space-between;
    align-items: flex-start; /* Zmieniono na flex-start, aby elementy były u góry */
    height: 100vh;
    margin: 0;
    padding: 0;
  }
</style>
<div class="left-column2">
  <h3> Dane kontaktowe </h3>
  <form action="/uwaga" target="_blank" class="registration-form">
    <label for="imie"> Imię </label>
    <input type="text" id="imie" name="imie">
    <label for="nazwisko"> Nazwisko </label>
    <input type="text" id="nazwisko" name="nazwisko">
    <label for="ulica"> Ulica </label>
    <input type="text" id="ulica" name="ulica">
    <label for="numer domu"> Numer domu </label>
    <input type="text" id="numer domu" name="numer domu">
    <label for="numer lokalu"> Numer lokalu * </label>
    <input type="text" id="numer lokalu" name="numer lokalu">
    <label for="kod pocztowy"> Kod pocztowy </label>
    <input type="text" id="kod pocztowy" name="kod pocztowy">
    <label for="miejscowosc"> Miejscowość </label>
    <input type="text" id="miejscowosc" name="miejscowosc">
```

```

<label for="wojewodztwo">Województwo</label>
<input type="text" id="wojewodztwo" name="wojewodztwo">
<label for="kraj">Kraj</label>
<input type="text" id="kraj" name="kraj">
<label for="adres e-mail">Adres e-mail</label>
<input type="text" id="adres e-mail" name="adres e-mail">
<label for="telefon">Telefon</label>
<input type="text" id="telefon" name="telefon">
<label for="nazwa firmy">Nazwa firmy*</label>
<input type="text" id="nazwa firmy" name="nazwa firmy">
<label for="NIP">NIP*</label>
<input type="text" id="NIP" name="NIP">
<h5>*pole opcjonalne</h5>
</form>
</div>
<div class="right-column2">
  <h2>REJESTRACJA</h2>
  <h3>Dane do logowania</h3>
  <form action="/uwaga" target="_blank">
    <label for="login">Login</label>
    <input type="text" id="login" name="login"><br>
    <label for="hasło">Hasło</label>
    <input type="password" id="hasło" name="hasło">
    <label for="powtórz hasło">POWTÓRZ HASŁO</label>
    <input type="password" id="powtórz hasło" name="powtórz hasło">
    <input type="submit" value="Załącz konto">
  </form>
  <a href="/" class="btn3">ZREZYGNUJ</a>
  <a href="/uwaga" class="btn3">ZAŁÓŻ KONTO</a>
</div>
{# Strona wprowadzająca dane nowego klienta #}
{% endblock %}

```

Rys.40 Zrzut strony rejestracji nowego użytkownika

Strona która pojawi się po naciśnięciu przycisku „ZAŁÓŻ KONTO”:

```
{% extends "szablon.html" %}
{% block content %}
<style>
  body {
    background-image: url("/static/tlo2.jpg");
    background-size: cover;
    display: flex;
    flex-direction: row;
    justify-content: space-between;
    align-items: center;
    height: 100vh;
    margin: 0;
    padding: 0;
  }
</style>
<div class="left-column">
  <h4>Twoje konto zostało utworzone pomyślnie.<br>Jeśli wystąpi błąd
    <br>skontaktuj się z<br>administratoerm systemu panixinski@labMA.com.pl</h4>
  <br><br><br><br><br><br>
  <a href="/panel_klienta" class="btn2">Przejdź do konta</a>
</div>
  {# Strona która pojawi się po założeniu konta #}
{% endblock %}
```



Rys.41 Zrzut strony zawierającej komunikat o utworzeniu konta

Strona logowania:

```
{% extends "szablon.html" %}
{% block content %}

<style>
  body {
    background-image: url("/static/tlo1.jpg");
    background-size: cover;
    display: flex;
```



```

flex-direction: row;
justify-content: space-between;
align-items: center;
height: 100vh;
margin: 0;
padding: 0;
}
</style>
<div class="left-column">
<form action="/logowanie" method = "post" target="_blank" >
  <label for="fname">Login:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Hasło:</label><br>
  <input type="password" id="lname" name="lname">
  <input type="submit" value="ZALOGUJ SIĘ" >
</form>
<br><br><br><br>
<a href="/" class="btn1">POWRÓT</a>
</div>

```

{# Strona do logowania. Wspólna dla klienta, kierownika laboratorium i pracowników. W zależności od roli danego użytkownika – sprawdzenie funkcją „AutoryzacjaUzytkownika” podczas logowania użytkownik zostaje przekierowany na strony z odpowiednią funkcjonalnością #}

{% endblock %}



Rys.42 Zrzut strony logowania.

PANEL KLIENTA

Strona główna panelu klienta:

```

{% extends "szablon1.html" %}
{% block content %}

```

```

<style>
  body {
    background-image: url("/static/tlo.jpg");
    background-size: cover;

```

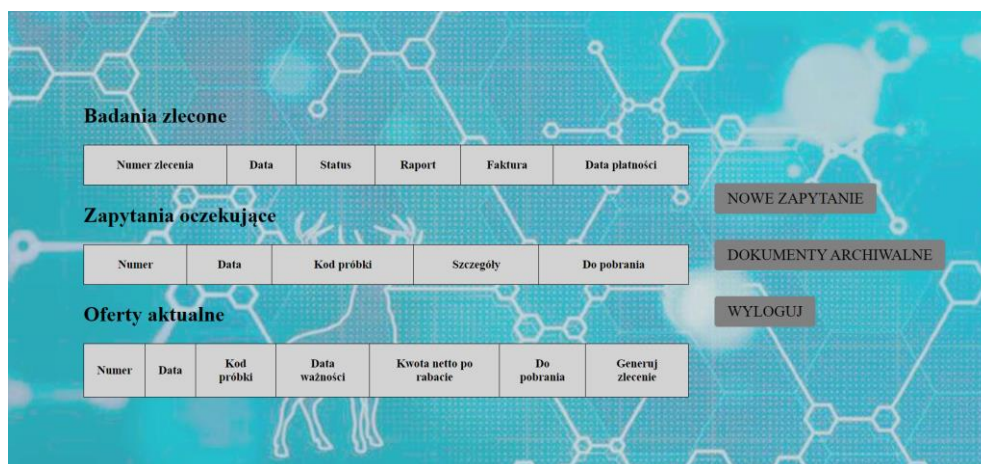
```

display: flex;
flex-direction: row;
justify-content: space-between;
align-items: center;
height: 100vh;
margin: 0;
padding: 0;
}
</style>
<div class="left-column">
  <h2>Badania zlecone</h2>
  <table class="custom-table">
    <tr>
      <th>Numer zlecenia</th>
      <th>Data</th>
      <th>Status</th>
      <th>Raport</th>
      <th>Faktura</th>
      <th>Data płatności</th>
    </tr>
  </table>
  <h2>Zapytania oczekujące</h2>
  <table class="custom-table">
    <tr>
      <th>Numer </th>
      <th>Data</th>
      <th>Kod próbki</th>
      <th>Szczegóły</th>
      <th>Do pobrania</th>
    </tr>
  </table>
  <h2>Oferty aktualne</h2>
  <table class="custom-table">
    <tr>
      <th>Numer</th>
      <th>Data</th>
      <th>Kod próbki</th>
      <th>Data ważności</th>
      <th>Kwota netto po rabacie</th>
      <th>Do pobrania</th>
      <th>Generuj zlecenie</th>
    </tr>
  </table>
</div>
<div class="right-column">
  <a href="/formularz_zapytania" class="btn1">NOWE ZAPYTANIE</a>
  <a href="/dokumenty_archiwalne" class="btn1">DOKUMENTY ARCHIWALNE</a>
  <a href="/" class="btn1">WYLOGUJ</a>

```


</div>

{# Pierwsze okno które pojawia się w aplikacji klienta po zalogowaniu. Klienta ma możliwość przeglądania wyników/postępów zleconych badań, zapytań oczekujących i ofert aktualnych. Ponadto klient może przejść na kolejną stronę aby złożyć nowe zapytanie, przeglądnąć dokumenty archiwalne bądź się wylogować #}
{% endblock %}



Rys.43 Zrzut pierwszej strony panelu klienta.

Strona która pojawia się po naciśnięciu przycisku „NOWE ZAPYTANIE”:

```
{% extends "szablon.html" %}
```

```
{% block content %}
```

```
<style>
```

```
body {
```

```
background-image: url("/static/tlo.jpg");
```

```
background-size: cover;
```

```
display: flex;
```

```
flex-direction: row;
```

```
justify-content: space-between;
```

```
align-items: center;
```

```
height: 100vh;
```

```
margin: 0;
```

```
padding: 0;
```

```
}
```

```
</style>
```

```
<div class="left-column2">
```

```
<h2>Nowe zapytanie ofertowe</h2>
```

```
<form action="/panel_klienta" target="_blank" >
```

```
<form action="/uwaga" target="_blank" >
```

```
<label for="badanie">Badanie</label>
```

```
<input type="text" id="badanie" name="imie">
```

```
<label for="matryca">Matryca</label><br>
```

```
<input type="text" id="matryca" name="nazwisko">
```

```
<label for="kod_probki">Kod próbki</label><br>
```

```

        <input type="text" id="kod probki" name="kod probki">
        <label for="komentarz">Komentarz</label><br>
        <input type="text" id="komentarz" name="komentarz"><br>
        <a href="/" class="btn1">Dodaj próbke do zapytania</a>
    </div>
    <div class="right-column1">
        <h2>Próbki do badań</h2>
        <table class="custom-table">
            <tr>
                <th>Numer</th>
                <th>Badanie</th>
                <th>Matryca</th>
                <th>Nazwa próbki</th>
                <th>Komentarz</th>
            </tr>
        </table>
        <br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
        <input type="submit" value="Wyślij zapytanie">
        <br>
        <a href="/uwaga_2" class="btn1">POWRÓT</a>
    </div>

```

{# Okno zapytania ofertowego. Klient może dodać większą ilość kombinacji badanie - matryca do zapytanie, które pojawią się w tabeli "Próbki do badań". Następnie może wysłać zapytanie bądź nacisnąć "POWRÓT", wtedy jest przekierowany na stronę informującą o konsekwencjach rezygnacji" #}

{% endblock %}

Rys.44 Zrzut strony do składania zapytania ofertowego.

Strona która pojawia się po naciśnięciu przycisku „POWRÓT”

```

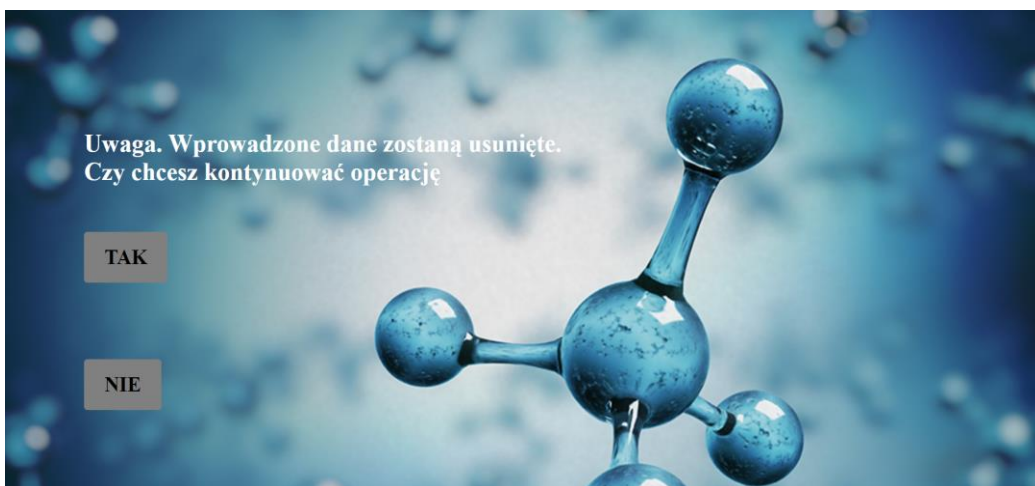
{% extends "szablon.html" %}
{% block content %}
<style>
    body {

```

```

background-image: url("/static/tlo2.jpg");
background-size: cover;
display: flex;
flex-direction: row;
justify-content: space-between;
align-items: center;
height: 100vh;
margin: 0;
padding: 0;
}
</style>
<div class="left-column">
  <h4>Uwaga. Wprowadzone dane zostaną usunięte.<br>Czy chcesz kontynuować operację</h4>
  <a href="/panel_klienta" class="btn2">TAK</a><br><br><br><br><br>
  <a href="/formularz_zapytania" class="btn2">NIE</a>
</div>
{# Informacja że wprowadzone dane zamówienia zostaną usunięte po opuszczeniu strony
NIE - powrót do strony zapytania TAK - przeniesienie do strony głównej panelu klienta #}
{% endblock %}

```



Rys.45 Zrzut strony informującej o konsekwencjach wyjścia ze strony zapytania.

Strona która pojawia się po naciśnięciu przycisku „DOKUMENTY ARCHIWALNE”:

```

{% extends "szablon.html" %}
{% block content %}
<style>
  body {
    background-image: url("/static/tlo.jpg");
    background-size: cover;
    display: flex;
    flex-direction: row;
    justify-content: space-between;
    align-items: center;

```

```

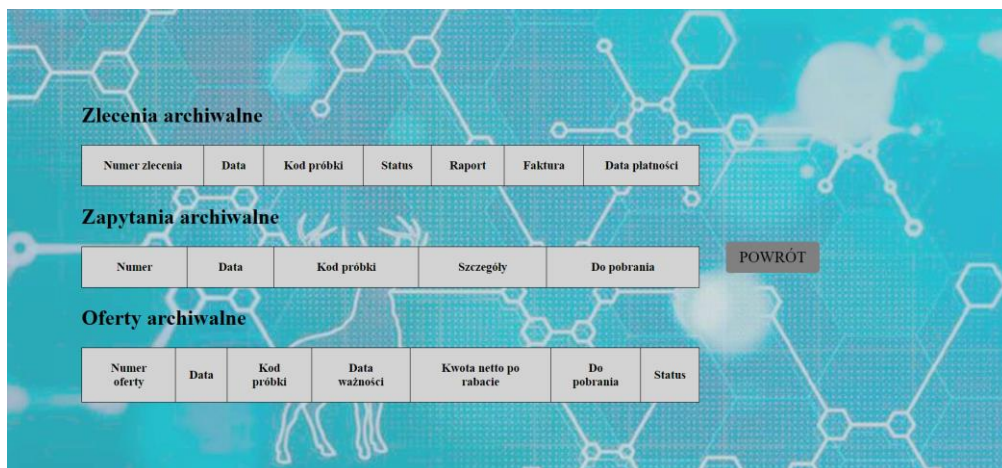
    height: 100vh;
    margin: 0;
    padding: 0;
}
</style>
<div class="left-column">
  <h2>Zlecenia archiwalne</h2>
  <table class="custom-table">
    <tr>
      <th>Numer zlecenia</th>
      <th>Data</th>
      <th>Kod próbki</th>
      <th>Status</th>
      <th>Raport</th>
      <th>Faktura</th>
      <th>Data płatności</th>
    </tr>
  </table>
  <h2>Zapytania archiwalne</h2>
  <table class="custom-table">
    <tr>
      <th>Numer </th>
      <th>Data</th>
      <th>Kod próbki</th>
      <th>Szczegóły</th>
      <th>Do pobrania</th>
    </tr>
  </table>
  <h2>Oferty archiwalne</h2>
  <table class="custom-table">
    <tr>
      <th>Numer oferty</th>
      <th>Data</th>
      <th>Kod próbki</th>
      <th>Data ważności</th>
      <th>Kwota netto po rabacie</th>
      <th>Do pobrania</th>
      <th>Status</th>
    </tr>
  </table>
</div>
<div class="right-column">
  <a href="/panel_klienta" class="btn1">POWRÓT</a>
</div>

```

```

{# Okno z tabelami zawierającymi zlecenia , zapytania i oferty archiwalne. Przycisk „Powrót”
przenosi klienta do strony głównej. #}
{% endblock %}

```



Rys.46 Zrzut strony dokumenty archiwalne.

PANEL KIERWONIKA

Strona główna panelu kierownika:

```
{% extends "szablon.html" %}
```

```
{% block content %}
```

```
<style>
```

```
body {
```

```
background-image: url("/static/tlo.jpg");
```

```
background-size: cover;
```

```
display: flex;
```

```
flex-direction: row;
```

```
justify-content: space-between;
```

```
align-items: center;
```

```
height: 100vh;
```

```
margin: 0;
```

```
padding: 0;
```

```
}
```

```
</style>
```

```
<div class="left-column2">
```

```
<h2>Przegląd zapytań</h2>
```

```
<table class="custom-table">
```

```
<tr>
```

```
<th>Nazwa klienta</th>
```

```
<th>Data zapytania</th>
```

```
<th>Numer zapytania</th>
```

```
<th>Dane zapytania</th>
```

```
<th>Generuj ofertę</th>
```

```
<th>Oferta wysłana</th>
```

```
</tr>
```

```
</table>
```

```
<h2>Przegląd zleceń</h2>
```

```
<table class="custom-table">
```

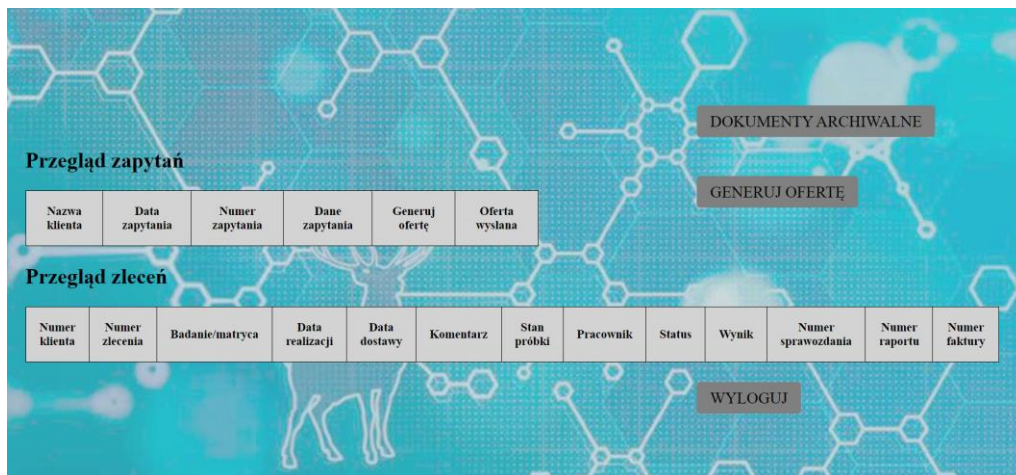


```

<tr>
  <th>Numer klienta</th>
  <th>Numer zlecenia</th>
  <th>Badanie/matryca</th>
  <th>Data realizacji</th>
  <th>Data dostawy</th>
  <th>Komentarz</th>
  <th>Stan próbki</th>
  <th>Pracownik</th>
  <th>Status</th>
  <th>Wynik</th>
  <th>Numer sprawozdania</th>
  <th>Numer raportu</th>
  <th>Numer faktury</th>
</tr>
</table>
</div>
<div class="right-column">
  <a href="/dokumenty_archiwalne_kierownik" class="btn1">DOKUMENTY ARCHIWALNE</a><br>
  <a href="/generowanie_oferty" class="btn1">GENERUJ
OFERTĘ</a><br><br><br><br><br><br><br><br><br><br><br>
  <a href="/" class="btn1">WYLOGUJ</a>
</div>

```

{# Pierwsza makietta która widzi kierownik po zalogowaniu. Możliwość przeglądania zapytań i zleceń. Przycisk "DOKUMENTY ARCHIWALNE" przenosi użytkownika do strony z dokumentami archiwalnymi a przycisk "GENERUJ OFERTĘ" do strony generowania oferty #}
{% endblock %}



Rys.47 Zrzut pierwszej strony panelu kierownika.

Strona która pojawia się po naciśnięciu przycisku „DOKUMENTY ARCHIWALNE”:

```

{% extends "szablon.html" %}
{% block content %}
<style>

```

```

body {
  background-image: url("/static/tlo.jpg");
  background-size: cover;
  display: flex;
  flex-direction: row;
  justify-content: space-between;
  align-items: center;
  height: 100vh;
  margin: 0;
  padding: 0;
}

```

```

</style>

```

```

<div class="left-column">

```

```

  <h2>Oferty archiwalne</h2>

```

```

    <table class="custom-table">

```

```

      <tr>

```

```

        <th>Numer klienta</th>

```

```

        <th>Numer oferty</th>

```

```

        <th>Data wystawienia</th>

```

```

        <th>Data ważności</th>

```

```

        <th>Data realizacji</th>

```

```

        <th>Badanie/matryca</th>

```

```

        <th>Kwota netto</th>

```

```

        <th>Rabat</th>

```

```

        <th>Kwota brutto</th>

```

```

        <th>Zlecona</th>

```

```

        <th>Numer zlecenia</th>

```

```

      </tr>

```

```

    </table>

```

```

    <h2>Zlecenia archiwalne</h2>

```

```

    <table class="custom-table">

```

```

      <tr>

```

```

        <th>Numer klienta </th>

```

```

        <th>Numer zlecenia</th>

```

```

        <th>Badanie/matryca</th>

```

```

        <th>Data realizacji</th>

```

```

        <th>Komentarz</th>

```

```

        <th>Pracownik</th>

```

```

        <th>Status</th>

```

```

        <th>Wyniki</th>

```

```

        <th>Sprawozdanie</th>

```

```

        <th>Raport</th>

```

```

        <th>Faktura</th>

```

```

      </tr>

```

```

    </table>

```

```

    <h2>Zapytania archiwalne</h2>

```

```

    <table class="custom-table">

```

```

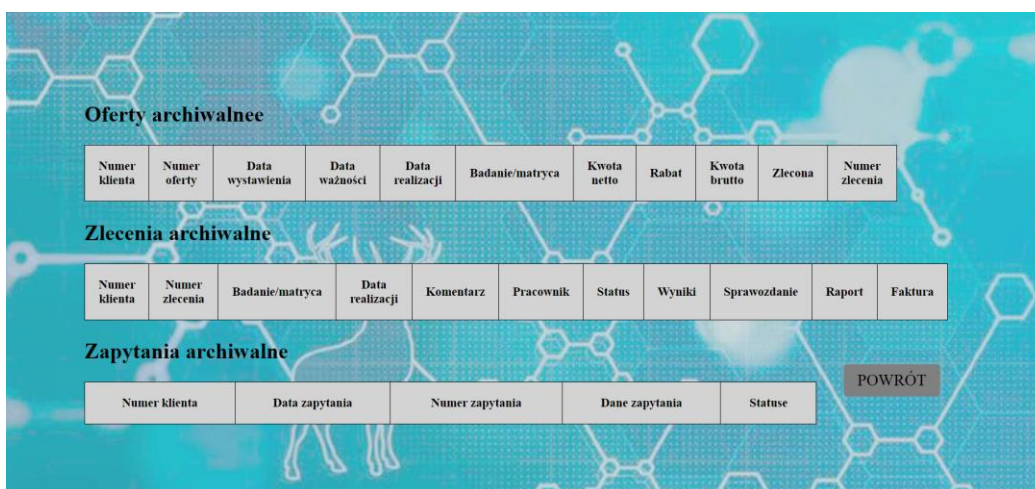
      <tr>

```

```

        <th>Numer klienta</th>
        <th>Data zapytania</th>
        <th>Numer zapytania</th>
        <th>Dane zapytania</th>
        <th>Statuse</th>
    </tr>
</table>
</div>
<div class="right-column">
    <br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
    <a href="/panel_kierownika" class="btn1">POWRÓT</a>
</div>
    {# Dokumenty archiwalne kierownika. #}
{% endblock %}

```



Rys.48 Zrzut strony dokumenty archiwalne w panelu kierownika.

PANEL PRACOWNIKA

Strona główna panelu pracownika:

```

{% extends "szablon.html" %}
{% block content %}
<style>
    body {
        background-image: url("/static/tlo.jpg");
        background-size: cover;
        display: flex;
        flex-direction: row;
        justify-content: space-between;
        align-items: center;
        height: 100vh;
        margin: 0;
        padding: 0;
    }
</style>

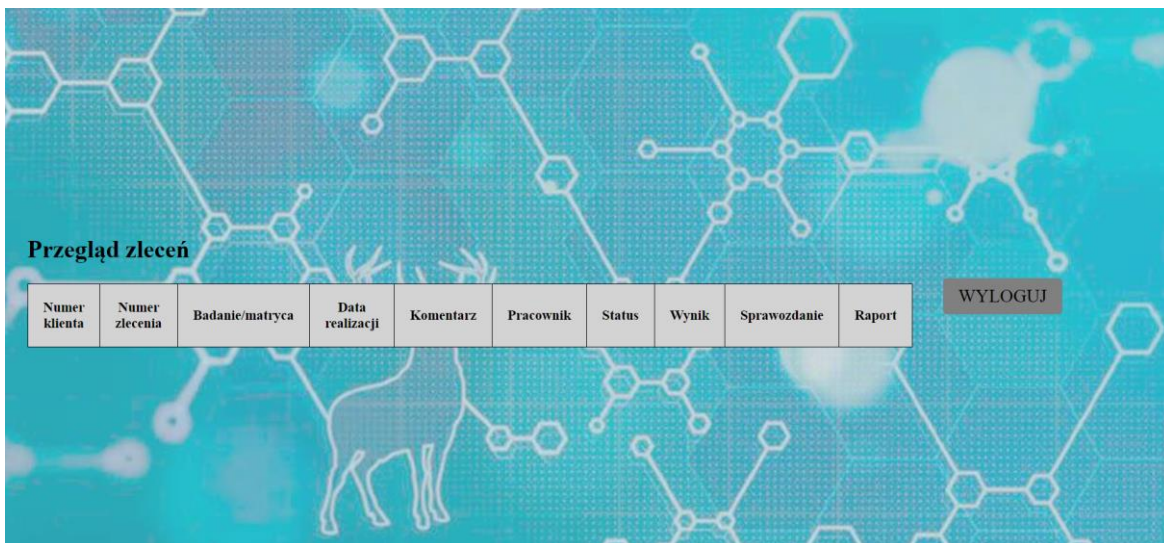
```



```

<div class="left-column1">
  <h2>Przegląd zleceń</h2>
  <table class="custom-table">
    <tr>
      <th>Numer klienta</th>
      <th>Numer zlecenia</th>
      <th>Badanie/matryca</th>
      <th>Data realizacji</th>
      <th>Komentarz</th>
      <th>Pracownik</th>
      <th>Status</th>
      <th>Wynik</th>
      <th>Sprawozdanie</th>
      <th>Raport</th>
    </tr>
  </table>
</div>
<div class="right-column">
  <a href="/" class="btn1">WYLOGUJ</a>
</div>
{% endblock %}

```



Rys.49 Zrzut pierwszej strony panelu pracownika.

6.2.Zaimplementowane elementy kodu Python.

Poniżej przedstawiono fragment kodu stanowiącego podstawę aplikacji internetowej napisanej w języku Python, przy użyciu frameworka Flask, która umożliwia użytkownikom logowanie i dostęp do różnych paneli w zależności od ich ról. Aplikacja korzysta z bazy danych SQL Server do autoryzacji użytkowników.

Najważniejsze elementy zamieszczonego kodu:

- Importowane moduły i funkcje:
 - Flask - importuje klasę Flask, która jest podstawą tworzenia aplikacji internetowej,
 - render_template- funkcja służąca do renderowania szablonów HTML,
 - request- obiekt umożliwiający dostęp do danych przesłanych przez klienta, takich jak formularz,
 - redirect- funkcja do przekierowywania użytkownika na inne strony.

- Inicjalizacja aplikacji:

`App = Flask(__name__,static_folder='static')` - tworzy instancję aplikacji Flask o nazwie App.

 Dodatkowo, określony jest folder dla plików statycznych (static_folder)

- Trasy (Endpoints) i Obsługa Zapytań HTTP:
 - @App.route(...) – definiuje trasy dostępne w aplikacji
 - Funkcje obsługujące trasy (np. get_tytulowa(), post_logowanie()) renderują lub przekierowują użytkownika w zależności od żądania http.

- Logowanie i Autoryzacja:

W funkcji post_logowanie() obsługiwane jest logowanie użytkowników. Dane przesyłane przez formularz są weryfikowane za pomocą funkcji autuzyt(). Na podstawie roli użytkownik przekierowywany jest na odpowiedni panel (Pracownik, Kierownik, Klient) lub strona błędu

- Połączenie z Bazą Danych:
 - tworzona jest instancja engine służąca do połączenia z bazą danych SQL Server,
 - w funkcji test() wykonuje się prosty test połączenia i wyświetla wynik,
 - funkcja autuzyt() wykonuje procedurę przechowywaną na bazie danych, która autoryzuje użytkownika na podstawie loginu i hasła

Kod aplikacji:

```
from flask import Flask, render_template, request, redirect
from services.db import autuzyt
```

```
App = Flask(__name__,static_folder='static')
```

```
@App.route('/')
def get_tytulowa():
    return render_template('tytulowa.html')
```

```
@App.route('/logowanie')
def get_logowanie():
    return render_template('logowanie.html')
```

```

@app.route('/logowanie', methods=['POST'])
def post_logowanie():
    print(request.form)
    a = autuzyt(request.form['fname'], request.form['lname'])
    match a:
        case 'Pracownik':
            return redirect('/panel_pracownika')
        case 'Kierownik':
            return redirect('/panel_kierownika')
        case 'Klient':
            return redirect('/panel_klienta')
        case _:
            return redirect('/blad')

@app.route('/rejestracja')
def get_rejestracja():
    return render_template('rejestracja.html')

@app.route('/panel_klienta')
def get_panel_klienta_1():
    return render_template('panel_klienta.html')

@app.route('/dokumenty_archiwalne')
def get_dokumenty_archiwalne():
    return render_template('dokumenty_archiwalne.html')

@app.route('/panel_kierownika')
def get_panel_kierownika():
    return render_template('panel_kierownika.html')

@app.route('/dokumenty_archiwalne_kierownik')
def get_dokumenty_archiwalne_kierownik():
    return render_template('dokumenty_archiwalne_kierownik.html')

@app.route('/panel_pracownika')
def get_panel_pracownika():
    return render_template('panel_pracownika.html')

@app.route('/uwaga')
def get_uwaga():
    return render_template('uwaga.html')

@app.route('/formularz_zapytania')
def get_formularz_zapytania():
    return render_template('formularz_zapytania.html')

```

```

@app.route('/uwaga_2')
def get_uwaga2():
    return render_template('uwaga_2.html')

@app.route('/generowanie_oferty')
def get_generowanie_oferty():
    return render_template('generowanie_oferty.html')

@app.route('/blad')
def blad():
    return render_template('blad.html')

from sqlalchemy import text

from sqlalchemy import create_engine

engine = create_engine("mssql+pyodbc://@DESKTOP-
HROC29U/u_agomulcz?trusted_connection=yes&driver=ODBC+Driver+17+for+SQL+Server",
echo=True)

def autuzyt(login, haslo):
    with engine.connect() as conn:
        result = conn.execute(text("select dbo.AutoryzacjaUzytkownika ('"+login+"', '"+haslo+"') as
'Rola' "))
    return result.mappings().all()[0]['Rola']

```

7. PODSUMOWANIA, WNIOSKI

Celem wykonanej pracy było opracowanie aplikacji bazodanowej do zarządzania zleceniami i wynikami badań w laboratorium chemicznym. Aplikacja miała zapewnić różny poziom dostępu klientom zlecającym badania, pracownikom laboratorium wykonującym badania oraz kierownikowi nadzorującemu badania i przygotowującemu oferty.

W pracy wykonano analizę wymagań użytkownika której efektem był diagram przypadków uwzględniający wszystkie wymagane funkcjonalności i aktywności zarówno po stronie pracowników laboratorium jak i klientów oraz szczegółowo opisane przypadki użycia zawierające sekwencje kroków lub akcji przedstawiających, jak użytkownik i system współdziałają w konkretnym przypadku użycia.

Przedstawiono makiety interfejsu użytkownika wykonane zgodnie z wytycznymi UX, zapewniające łatwe i przyjemne przemieszczanie się użytkownika po aplikacji.

Sercem aplikacji jest baza danych zawierająca 15 tabel połączonych ze sobą głównie relacjami jeden do wielu i jeden do jeden. Zastosowano szereg ograniczeń zarówno wpływających na kształt i format wprowadzanych danych jak i unikalność danych oraz defaultowość, które ostatecznie miały

na celu zapewnić spójność i integralność bazy danych. W przypadku miejscowości zastosowano technikę słownika.

W pracy zastosowano warstwy dostępu w postaci widoków i triggerów. Widoki zostały skonstruowane w taki sposób aby były spójne z tabelami i zestawieniami widzianymi przez użytkowników aplikacji. Triggery natomiast miały ułatwić pracę pracownikom poprzez automatyzację nadawania numerów sprawozdań oraz aktualizację cen.

W pracy wykonano pełny interfejs użytkownika w kodzie HTML. Utworzono szkielet w kodzie Python z przechodzeniem pomiędzy stronami. Komunikacja z bazą danych ogranicza się do wywołania procedury logowania. Dalsze prace nad aplikacją będą prowadzone.

7.1. Dalsze prace

Aplikacja będąca celem tej pracy, kiedy będzie skończona zdecydowanie ułatwi przepływ informacji zarówno pomiędzy samymi pracownikami, jak i pracownikami i klientem. Automatyzacja zapytań zastępująca jeszcze bardzo często stosowaną drogę mailową zdecydowanie przyspieszy otrzymanie przez klienta informacji zwrotnej, natomiast dla samego kierownika ułatwi przygotowanie oferty i nadzór nad ewentualnym złożeniem zlecenia. Klient będzie czerpał szereg korzyści z aplikacji, ponieważ na bieżąco będzie mógł śledzić status zleconych badań, a po ich wykonaniu i rozliczeniu finansowym będzie miał ciągły wgląd we wszystkie dokumenty związane ze zleceniem. Aplikacja zapewni płynną komunikację pomiędzy pracownikami laboratorium w zakresie przydzielenia pracowników do wykonania badań, sprawdzenia i autoryzacji wyników oraz przygotowania finalnego raportu dla klienta. Wszyscy zaangażowani w dane badanie pracownicy będą mieli możliwość nadzorowania statusu realizacji zadań. Prezentowana aplikacja po ukończeniu będzie użytecznym narzędziem w pracy każdego laboratorium.

8. LITERATURA

<https://www.w3schools.com/html/default.asp>

https://docs.sqlalchemy.org/en/20/core/engines.html#sqlalchemy.create_engine

<https://www.connectionstrings.com/sql-server/>

https://docs.sqlalchemy.org/en/20/tutorial/dbapi_transactions.html

<https://docs.python.org/pl/3.8/tutorial/index.html>

<https://learn.microsoft.com/pl-pl/training/modules/introduction-to-transact-sql/>

<https://learn.microsoft.com/pl-pl/training/paths/program-transact-sql/>

<https://www.w3schools.com/python/default.asp>

9. SKRYPT BAZY DANYCH

```
USE [master]
GO
/***** Object: Database [u_agomulcz]  Script Date: 26.09.2023 21:10:40 *****/
CREATE DATABASE [u_agomulcz]
    CONTAINMENT = NONE
    ON PRIMARY
( NAME = N'Laboratorium', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\Laboratorium.mdf' , SIZE = 8192KB , MAXSIZE =
UNLIMITED, FILEGROWTH = 65536KB )
    LOG ON
( NAME = N'Laboratorium_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\Laboratorium_log.ldf' , SIZE = 8192KB , MAXSIZE =
2048GB , FILEGROWTH = 65536KB )
    WITH CATALOG_COLLATION = DATABASE_DEFAULT
GO
ALTER DATABASE [u_agomulcz] SET COMPATIBILITY_LEVEL = 150
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [u_agomulcz].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [u_agomulcz] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [u_agomulcz] SET ANSI_NULLS OFF
GO
ALTER DATABASE [u_agomulcz] SET ANSI_PADDING OFF
GO
ALTER DATABASE [u_agomulcz] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [u_agomulcz] SET ARITHABORT OFF
GO
ALTER DATABASE [u_agomulcz] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [u_agomulcz] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [u_agomulcz] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [u_agomulcz] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [u_agomulcz] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [u_agomulcz] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [u_agomulcz] SET NUMERIC_ROUNDABORT OFF
```

```

GO
ALTER DATABASE [u_agomulcz] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [u_agomulcz] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [u_agomulcz] SET DISABLE_BROKER
GO
ALTER DATABASE [u_agomulcz] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [u_agomulcz] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [u_agomulcz] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [u_agomulcz] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [u_agomulcz] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [u_agomulcz] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [u_agomulcz] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [u_agomulcz] SET RECOVERY FULL
GO
ALTER DATABASE [u_agomulcz] SET MULTI_USER
GO
ALTER DATABASE [u_agomulcz] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [u_agomulcz] SET DB_CHAINING OFF
GO
ALTER DATABASE [u_agomulcz] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [u_agomulcz] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
ALTER DATABASE [u_agomulcz] SET DELAYED_DURABILITY = DISABLED
GO
ALTER DATABASE [u_agomulcz] SET ACCELERATED_DATABASE_RECOVERY = OFF
GO
EXEC sys.sp_db_vardecimal_storage_format N'u_agomulcz', N'ON'
GO
ALTER DATABASE [u_agomulcz] SET QUERY_STORE = OFF
GO
USE [u_agomulcz]
GO
/***** Object: UserDefinedFunction [dbo].[AutoryzacjaUzytkownika]    Script Date: 26.09.2023
21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```



```

GO
CREATE FUNCTION [dbo].[AutoryzacjaUzytkownika]
    (@Login NVARCHAR(50),
     @Haslo NVARCHAR(50))
RETURNS NVARCHAR(50)
AS
BEGIN
    DECLARE @Rola NVARCHAR(50)

    -- Sprawdzenie, czy uzytkownik to kierownik
    SELECT @Rola = 'Kierownik'
    FROM Pracownicy p
    INNER JOIN Konta_pracownicy kp ON p.ID_Pracownika = kp.ID_konta
    WHERE p.Stanowisko = 'Kierownik' AND kp.login = @Login AND kp.haslo = @Haslo

    IF @Rola IS NOT NULL
    BEGIN
        -- Jeśli to kierownik, zwróć jego rolę
        RETURN @Rola
    END

    -- Jeśli nie jest kierownikiem, sprawdź, czy to pracownik
    SELECT @Rola = 'Pracownik'
    FROM Pracownicy p
    INNER JOIN Konta_pracownicy kp ON p.ID_Pracownika = kp.ID_Konta
    WHERE p.Stanowisko <> 'Kierownik' AND kp.login = @Login AND kp.haslo = @Haslo

    IF @Rola IS NOT NULL
    BEGIN
        -- Jeśli to pracownik, zwróć jego rolę
        RETURN @Rola
    END

    -- Jeśli nie pasuje do żadnej z tych ról, sprawdź, czy to klient
    SELECT @Rola = 'Klient'
    FROM Konta_klient kk
    WHERE kk.login = @Login AND kk.haslo = @Haslo

    IF @Rola IS NOT NULL
    BEGIN
        -- Jeśli to klient, zwróć jego rolę
        RETURN @Rola
    END

    -- Jeśli nie pasuje do żadnej z tych kategorii, zwróć komunikat o braku konta
    RETURN 'Brak konta'
END
GO

```

```

/***** Object: Table [dbo].[Matryca]  Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Matryca](
    [ID_Matrycy] [int] IDENTITY(1,1) NOT NULL,
    [Nazwa_matrycy] [nvarchar](20) NOT NULL,
    CONSTRAINT [PK_Matryca] PRIMARY KEY CLUSTERED
(
    [ID_Matrycy] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY],
    CONSTRAINT [U_Nazwa_matrycy] UNIQUE NONCLUSTERED
(
    [Nazwa_matrycy] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

/***** Object: Table [dbo].[Szczegoly_zlecenia]  Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Szczegoly_zlecenia](
    [ID_Szczegoly] [int] IDENTITY(1,1) NOT NULL,
    [ID_Zapytania] [int] NOT NULL,
    [ID_Bad_Mat] [int] NOT NULL,
    [Cena] [money] NULL,
    [Rabat] [nchar](10) NULL,
    [ID_Pracownika] [int] NULL,
    [Kod_probki] [nchar](10) NULL,
    [Komentarz] [text] NULL,
    CONSTRAINT [PK_Szczegoly_zlecenia] PRIMARY KEY CLUSTERED
(
    [ID_Szczegoly] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

```

```

/***** Object: Table [dbo].[Zapytania ]  Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
CREATE TABLE [dbo].[Zapytania ](
    [ID_Zapytania] [int] IDENTITY(1,1) NOT NULL,
    [ID_Klienta] [nchar](4) NOT NULL,
    [Data_zapytania] [date] NOT NULL,
    [Czas_realizacji_badan] [nchar](10) NULL,
    [Data_przygotowania_oferty] [date] NULL,
    [Data_waznosci_oferty] [date] NULL,
    [Status] [nchar](12) NULL,
    CONSTRAINT [PK_Zapytania ] PRIMARY KEY CLUSTERED
(
    [ID_Zapytania] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Badania]  Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Badania](
    [ID_Badania] [int] IDENTITY(1,1) NOT NULL,
    [Nazwa_badania] [nvarchar](50) NOT NULL,
    [Cena_badania] [money] NOT NULL,
    [Czas_realizacji] [nvarchar](10) NULL,
    [Ilosc_probki] [nchar](10) NULL,
    CONSTRAINT [PK_Badania] PRIMARY KEY CLUSTERED
(
    [ID_Badania] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY],
    CONSTRAINT [U_Nazwa_badania] UNIQUE NONCLUSTERED
(
    [Nazwa_badania] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Badania_Matryca]  Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Badania_Matryca](

```

```

[ID_Bad_Mat] [int] IDENTITY(1,1) NOT NULL,
[ID_Matrycy] [int] NOT NULL,
[ID_Badania] [int] NOT NULL,
[Warunki_transportu] [text] NOT NULL,
CONSTRAINT [PK_Badania_Matryca] PRIMARY KEY CLUSTERED
(
    [ID_Bad_Mat] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/***** Object: View [dbo].[PK__Zapytania_oczekujace_szczegoly]    Script Date: 26.09.2023
21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE view [dbo].[PK__Zapytania_oczekujace_szczegoly]
as
    select sz.ID_Zapytania ,b.Nazwa_badania, m.Nazwa_matrycy,za.Kod_probki ,za.Komentarz
    from [Szczegoly_zlecenia] as sz
    join Badania_Matryca as bm on bm.ID_Bad_Mat=sz.ID_Bad_Mat
    join Badania as b on b.ID_Badania=bm.ID_Badania
    join Matryca as m on m.ID_Matrycy=bm.ID_Matrycy
    join [Zapytania ] as za on za.ID_Zapytania=sz.ID_Zapytania
GO
/***** Object: View [dbo].[Kwota_netto_po_rabacie]    Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE view [dbo].[Kwota_netto_po_rabacie]
as
select ID_Zapytania, sum(Cena*(100-Rabat)/100) as'Kwota netto po rabacie'
from [Szczegoly_zlecenia]
group by ID_Zapytania
GO
/***** Object: Table [dbo].[Zlecenia ]    Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Zlecenia ](
    [ID_Zlecenia] [int] IDENTITY(1,1) NOT NULL,
    [Data_zlecenia] [date] NULL,
    [Stan_probki] [nchar](10) NULL,
    [Data_dostarczenia_probki] [date] NULL,

```

```

[Numer_zlecenia_w_roku] [nvarchar](20) NULL,
[Data_zatwierdzenia_raportu] [datetime] NULL,
[Data_wystawienia_faktury] [date] NULL,
[Data_platnosci] [date] NULL,
[Raport_zatwierdzil_Pracownik_ID] [int] NULL,
[Status] [nchar](10) NULL,
CONSTRAINT [PK_Zlecenia - klient] PRIMARY KEY CLUSTERED
(
    [ID_Zlecenia] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY],
CONSTRAINT [U_nr_zlec_roku] UNIQUE NONCLUSTERED
(
    [Numer_zlecenia_w_roku] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: View [dbo].[PK_Badania_Zlecone_Archiwalne]    Script Date: 26.09.2023 21:10:40
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE view [dbo].[PK_Badania_Zlecone_Archiwalne]
as
    select za.ID_Klienta, ID_Zlecenia, Data_zlecenia, zl.Status
    from [Zlecenia ] as zl
    join [Zapytania ] as za on za.ID_Zapytania=zl.ID_Zlecenia
    where zl.Status in ('Zakończzone')

GO
/***** Object: View [dbo].[Zestawienie_oferty_Archiwalne]    Script Date: 26.09.2023 21:10:40
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE view [dbo].[Zestawienie_oferty_Archiwalne]
as
    select      ID_Klienta      ,ID_Zapytania,      Data_przygotowania_oferty,      Kod_probki,
    Data_waznosci_oferty,
        (select [Kwota netto po rabacie] from [dbo].[Kwota_netto_po_rabacie] as kn
        where kn.ID_Zapytania = za.Id_Zapytania) as 'Kwota netto po rabacie'
    from [Zapytania ] as za

```

```

        where za.Data_waznosci_oferty < GETDATE()
GO
/***** Object: View [dbo].[Szczegóły_oferty]  Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE view [dbo].[Szczegóły_oferty]
as
select      sz.ID_Zapytania      as 'Numer      oferty',      za.Data_przygotowania_oferty,
za.Kod_probki,za.Data_waznosci_oferty,b.Nazwa_badania,
m.Nazwa_matrycy,sz.Cena as 'Cena jednostkowa za badanie', Rabat, round((Cena*(100-
sz.Rabat)/100), 1) as 'Kwota netto po rabacie',
cast(round((Cena*(100-sz.Rabat)/100)* 0.22, 1)AS decimal(10,2)) as 'Vat', cast(round((Cena*(100-
sz.Rabat)/100)* 1.22, 1) AS decimal(10,2)) as 'Kwota brutto'
from [Szczegoly_zlecenia] as sz
join Badania_Matryca as bm on bm.ID_Bad_Mat=sz.ID_Bad_Mat
join Badania as b on b.ID_Badania = bm.ID_Badania
join Matryca as m on m.ID_Matrycy=bm.ID_Matrycy
join [Zapytania ] as za on za.ID_Zapytania = sz.ID_Zapytania
GO
/***** Object: View [dbo].[Zestawienie_oferty_Aktualne]  Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE view [dbo].[Zestawienie_oferty_Aktualne]
as
        select      ID_Klienta      ,ID_Zapytania,      Data_przygotowania_oferty,      Kod_probki,
Data_waznosci_oferty,
        (select [Kwota netto po rabacie] from [dbo].[Kwota_netto_po_rabacie] as kn
        where kn.ID_Zapytania = za.Id_Zapytania) as 'Kwota netto po rabacie'
        from [Zapytania ] as za
        where za.Data_waznosci_oferty >= GETDATE()
GO
/***** Object: View [dbo].[PK_Badania_Zlecone_Aktualne]  Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
Create view [dbo].[PK_Badania_Zlecone_Aktualne]
as
        select za.ID_Klienta,ID_Zlecenia, Data_zlecenia, zl.Status, zl.Data_platnosci
        from [Zlecenia ] as zl
        join [Zapytania ] as za on za.ID_Zapytania=zl.ID_Zlecenia
        where zl.Status not in ('Zakończone')

```

```

GO
/***** Object: Table [dbo].[Klienci]  Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Klienci](
    [ID_Klienta] [nchar](4) NOT NULL,
    [Nazwa_firmy] [nvarchar](40) NULL,
    [Imie] [nvarchar](50) NOT NULL,
    [Nazwisko] [nvarchar](50) NOT NULL,
    [Adres_ulica] [nvarchar](30) NOT NULL,
    [Adres_numer_domu] [nvarchar](10) NOT NULL,
    [Adres_numer_lokalu] [nvarchar](10) NULL,
    [ID_Miejscowosc] [int] NOT NULL,
    [Kod_pocztowy] [char](6) NOT NULL,
    [Telefon] [nvarchar](15) NOT NULL,
    [email] [nvarchar](40) NOT NULL,
    [NIP] [char](10) NULL,
    CONSTRAINT [PK_Klienci] PRIMARY KEY CLUSTERED
(
    [ID_Klienta] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY],
    CONSTRAINT [email_unique] UNIQUE NONCLUSTERED
(
    [email] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY],
    CONSTRAINT [Nazwa_firmy_unique] UNIQUE NONCLUSTERED
(
    [Nazwa_firmy] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY],
    CONSTRAINT [Nazwa_Klienta_unique] UNIQUE NONCLUSTERED
(
    [Nazwa_firmy] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY],
    CONSTRAINT [NIP_unique] UNIQUE NONCLUSTERED
(
    [NIP] ASC

```



```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY],
CONSTRAINT [U_Tab_klienci] UNIQUE NONCLUSTERED
(
    [Nazwa_firmy] ASC,
    [Telefon] ASC,
    [email] ASC,
    [NIP] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: View [dbo].[PK_Zapytania_oczekujace] Script Date: 26.09.2023 21:10:40
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE view [dbo].[PK_Zapytania_oczekujace]
as
    select k.ID_Klienta, ID_Zapytania as 'Numer Zapytania', Data_zapytania, Kod_probki, Status
    from [Zapytania ] as z
    join Klienci as k on k.ID_Klienta = z.ID_Klienta
    where Status = 'Złożone'
GO
/***** Object: View [dbo].[PK_Zapytania_archiwum] Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE view [dbo].[PK_Zapytania_archiwum]
as
    select k.ID_Klienta, ID_Zapytania as 'Numer Zapytania', Data_zapytania, Kod_probki, Status
    from [Zapytania ] as z
    join Klienci as k on k.ID_Klienta = z.ID_Klienta
    where Status <> 'Złożone'
GO
/***** Object: Table [dbo].[Wyniki_badania] Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Wyniki_badania](
    [ID_Wyniki_badania] [int] IDENTITY(1,1) NOT NULL,
    [ID_Szczegoly] [int] NULL,
    [Numer_sprawozdania_w_roku] [nvarchar](20) NULL,

```

```

        [Data_wykonania_badania] [date] NULL,
        [Data_sprawozdania] [date] NULL,
        [Wynik] [nchar](10) NULL,
        [Jednostka] [nchar](10) NULL,
CONSTRAINT [PK_Wyniki_badania_1] PRIMARY KEY CLUSTERED
(
        [ID_Wyniki_badania] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: View [dbo].[Wzór_raportu]  Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE view [dbo].[Wzór_raportu]
as
        select  zl.ID_Zlecenia,concat(zl.ID_Zlecenia,'/',kl.ID_Klienta,'/', year(getdate())) as 'Numer
raportu',  za.Kod_probki,  b.Nazwa_badania,  m.Nazwa_matrycy,  concat(w.Wynik,' ',w.
Jednostka) as 'Wynik', w.Data_wykonania_badania as'Data badania'
        from [Zlecenia ] as zl
        join [Zapytania ] as za on zl.ID_Zlecenia=za.ID_Zapytania
        join Klienci as kl on kl.ID_Klienta=za.ID_Klienta
        join [Szczegoly_zlecenia] as sz on  sz.ID_Zapytania=za.ID_Zapytania
        join Wyniki_badania as w on w.ID_Szczegoly = sz.ID_Szczegoly
        join Badania_Matryca as bm on bm.ID_Bad_Mat = sz.ID_Bad_Mat
        join Badania as b on b.ID_Badania= bm.ID_Badania
        join Matryca as m on m.ID_Matrycy= bm.ID_Matrycy

GO
/***** Object: Table [dbo].[Badania_Pracownicy]  Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Badania_Pracownicy](
        [ID_Bad_Prac] [int] IDENTITY(1,1) NOT NULL,
        [ID_Badania] [int] NOT NULL,
        [ID_Pracownika] [int] NOT NULL,
CONSTRAINT [PK_Badania_Pracownicy] PRIMARY KEY CLUSTERED
(
        [ID_Bad_Prac] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
/***** Object: Table [dbo].[Konta_klient]  Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Konta_klient](
    [ID_konta] [nchar](4) NOT NULL,
    [login] [nvarchar](50) NOT NULL,
    [haslo] [nvarchar](max) NOT NULL,
    CONSTRAINT [PK_Konta] PRIMARY KEY CLUSTERED
(
    [ID_konta] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/***** Object: Table [dbo].[Konta_pracownicy]  Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Konta_pracownicy](
    [ID_konta] [int] NOT NULL,
    [login] [nvarchar](50) NOT NULL,
    [haslo] [nvarchar](max) NOT NULL,
    CONSTRAINT [PK_Konta_pracownicy] PRIMARY KEY CLUSTERED
(
    [ID_konta] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/***** Object: Table [dbo].[Kraj]  Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Kraj](
    [ID_Kraj] [int] IDENTITY(1,1) NOT NULL,
    [Nazwa_kraju] [nvarchar](20) NOT NULL,
    CONSTRAINT [PK_Kraj] PRIMARY KEY CLUSTERED
(
    [ID_Kraj] ASC

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY],
CONSTRAINT [U_kraj] UNIQUE NONCLUSTERED
(
    [Nazwa_kraju] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Miasta] Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Miasta](
    [ID_Miejscowosc] [int] IDENTITY(1,1) NOT NULL,
    [Nazwa_miasta] [nvarchar](20) NOT NULL,
    [ID_Wojewodztwa] [int] NULL,
    [ID_Kraj] [int] NOT NULL,
    CONSTRAINT [PK_Miasta_1] PRIMARY KEY CLUSTERED
(
    [ID_Miejscowosc] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY],
CONSTRAINT [U_miasta] UNIQUE NONCLUSTERED
(
    [Nazwa_miasta] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Pracownicy] Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Pracownicy](
    [ID_Pracownika] [int] IDENTITY(1,1) NOT NULL,
    [Imie] [nvarchar](20) NOT NULL,
    [Nazwisko] [nvarchar](20) NOT NULL,
    [Stanowisko] [nvarchar](30) NOT NULL,
    [Data_zatrudnienia] [date] NOT NULL,
    [Data_urodzenia] [date] NOT NULL,
    [Adres_nazwa_ulicy] [nvarchar](20) NOT NULL,

```

```

        [Adres_numer_domu] [nvarchar](10) NOT NULL,
        [Adres_numer_lokalu] [nvarchar](10) NULL,
        [ID_Miejscowosc] [int] NOT NULL,
        [Kod_pocztowy] [nvarchar](10) NOT NULL,
        [Telefon] [nvarchar](15) NOT NULL,
        [email] [nvarchar](20) NULL,
CONSTRAINT [PK_Pracownicy] PRIMARY KEY CLUSTERED
(
        [ID_Pracownika] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY],
CONSTRAINT [U_Tab_Pracownicy] UNIQUE NONCLUSTERED
(
        [Telefon] ASC,
        [email] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[TabelaTymczasowa] Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TabelaTymczasowa](
        [Numer] [int] IDENTITY(1,1) NOT NULL,
        [Badanie] [nvarchar](50) NULL,
        [Matryca] [nvarchar](20) NULL,
        [KodProbki] [nchar](7) NULL,
        [Komentarz] [text] NULL,
PRIMARY KEY CLUSTERED
(
        [Numer] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/***** Object: Table [dbo].[Wojewodztwa] Script Date: 26.09.2023 21:10:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Wojewodztwa](
        [ID_Wojewodztwa] [int] IDENTITY(1,1) NOT NULL,
        [Nazwa_wojewodztwa] [nvarchar](20) NOT NULL,

```

```

CONSTRAINT [PK_Województwa_1] PRIMARY KEY CLUSTERED
(
    [ID_Wojewodztwa] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY],
CONSTRAINT [U_województwa] UNIQUE NONCLUSTERED
(
    [Nazwa_województwa] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Kraj] ADD CONSTRAINT [DF_kraj] DEFAULT ('Polska') FOR [Nazwa_kraju]
GO
ALTER TABLE [dbo].[Zapytania ] ADD CONSTRAINT [DF_Zapytania_Data_zapytania] DEFAULT
(getdate()) FOR [Data_zapytania]
GO
ALTER TABLE [dbo].[Zapytania ] ADD CONSTRAINT [DF_zapytania_status] DEFAULT ('Złożone') FOR
[Status]
GO
ALTER TABLE [dbo].[Zlecenia ] ADD CONSTRAINT [DF_zlecenia] DEFAULT (getdate()) FOR
[Data_zlecenia]
GO
ALTER TABLE [dbo].[Zlecenia ] ADD CONSTRAINT [DF_data_dostarczenia] DEFAULT (getdate()) FOR
[Data_dostarczenia_probki]
GO
ALTER TABLE [dbo].[Zlecenia ] ADD CONSTRAINT [DF_data_raportu] DEFAULT (getdate()) FOR
[Data_zatwierdzenia_raportu]
GO
ALTER TABLE [dbo].[Zlecenia ] ADD CONSTRAINT [DF_data_faktury] DEFAULT (getdate()) FOR
[Data_wystawienia_faktury]
GO
ALTER TABLE [dbo].[Zlecenia ] ADD CONSTRAINT [DF_data_płatności] DEFAULT (getdate()+(14))
FOR [Data_platnosci]
GO
ALTER TABLE [dbo].[Zlecenia ] ADD CONSTRAINT [DF_status] DEFAULT ('Zlecone') FOR [Status]
GO
ALTER TABLE [dbo].[Badania_Matryca] WITH CHECK ADD CONSTRAINT [FK_łącznikowa
Bad_Matr._Badania] FOREIGN KEY([ID_Matrycy])
REFERENCES [dbo].[Badania] ([ID_Badania])
GO
ALTER TABLE [dbo].[Badania_Matryca] CHECK CONSTRAINT [FK_łącznikowa Bad_Matr._Badania]
GO
ALTER TABLE [dbo].[Badania_Matryca] WITH CHECK ADD CONSTRAINT [FK_łącznikowa
Bad_Matr._Matryca] FOREIGN KEY([ID_Matrycy])
REFERENCES [dbo].[Matryca] ([ID_Matrycy])

```

```

GO
ALTER TABLE [dbo].[Badania_Matryca] CHECK CONSTRAINT [FK_łącznikowa Bad_Matr._Matryca]
GO
ALTER TABLE [dbo].[Badania_Pracownicy] WITH CHECK ADD CONSTRAINT [FK_łącznikowa
Bad_Prac._Badania] FOREIGN KEY([ID_Badania])
REFERENCES [dbo].[Badania] ([ID_Badania])
GO
ALTER TABLE [dbo].[Badania_Pracownicy] CHECK CONSTRAINT [FK_łącznikowa Bad_Prac._Badania]
GO
ALTER TABLE [dbo].[Badania_Pracownicy] WITH CHECK ADD CONSTRAINT [FK_łącznikowa
Bad_Prac._Pracownicy] FOREIGN KEY([ID_Pracownika])
REFERENCES [dbo].[Pracownicy] ([ID_Pracownika])
GO
ALTER TABLE [dbo].[Badania_Pracownicy] CHECK CONSTRAINT [FK_łącznikowa
Bad_Prac._Pracownicy]
GO
ALTER TABLE [dbo].[Klienci] WITH CHECK ADD CONSTRAINT [FK_Klienci_Miasta] FOREIGN
KEY([ID_Miejscowosc])
REFERENCES [dbo].[Miasta] ([ID_Miejscowosc])
GO
ALTER TABLE [dbo].[Klienci] CHECK CONSTRAINT [FK_Klienci_Miasta]
GO
ALTER TABLE [dbo].[Konta_klient] WITH CHECK ADD CONSTRAINT [FK_Konta_klient_Klienci]
FOREIGN KEY([ID_konta])
REFERENCES [dbo].[Klienci] ([ID_Klienta])
GO
ALTER TABLE [dbo].[Konta_klient] CHECK CONSTRAINT [FK_Konta_klient_Klienci]
GO
ALTER TABLE [dbo].[Konta_pracownicy] WITH CHECK ADD CONSTRAINT
[FK_Konta_pracownicy_Pracownicy] FOREIGN KEY([ID_konta])
REFERENCES [dbo].[Pracownicy] ([ID_Pracownika])
GO
ALTER TABLE [dbo].[Konta_pracownicy] CHECK CONSTRAINT [FK_Konta_pracownicy_Pracownicy]
GO
ALTER TABLE [dbo].[Miasta] WITH CHECK ADD CONSTRAINT [FK_Miasta_Kraj] FOREIGN
KEY([ID_Kraj])
REFERENCES [dbo].[Kraj] ([ID_Kraj])
GO
ALTER TABLE [dbo].[Miasta] CHECK CONSTRAINT [FK_Miasta_Kraj]
GO
ALTER TABLE [dbo].[Miasta] WITH CHECK ADD CONSTRAINT [FK_Miasta_Województwa] FOREIGN
KEY([ID_Wojewodztwa])
REFERENCES [dbo].[Wojewodztwa] ([ID_Wojewodztwa])
GO
ALTER TABLE [dbo].[Miasta] CHECK CONSTRAINT [FK_Miasta_Województwa]
GO
ALTER TABLE [dbo].[Pracownicy] WITH CHECK ADD CONSTRAINT [FK_Pracownicy_Miasta]
FOREIGN KEY([ID_Miejscowosc])

```



```

REFERENCES [dbo].[Miasta] ([ID_Miejscowosc])
GO
ALTER TABLE [dbo].[Pracownicy] CHECK CONSTRAINT [FK_Pracownicy_Miasta]
GO
ALTER TABLE [dbo].[Szczegoly_zlecenia] WITH CHECK ADD CONSTRAINT
[FK_Szczegoly_zlecenia_Badania_Matryca] FOREIGN KEY([ID_Bad_Mat])
REFERENCES [dbo].[Badania_Matryca] ([ID_Bad_Mat])
GO
ALTER TABLE [dbo].[Szczegoly_zlecenia] CHECK CONSTRAINT
[FK_Szczegoly_zlecenia_Badania_Matryca]
GO
ALTER TABLE [dbo].[Szczegoly_zlecenia] WITH CHECK ADD CONSTRAINT
[FK_Szczegoly_zlecenia_Zapytania ] FOREIGN KEY([ID_Zapytania])
REFERENCES [dbo].[Zapytania ] ([ID_Zapytania])
GO
ALTER TABLE [dbo].[Szczegoly_zlecenia] CHECK CONSTRAINT [FK_Szczegoly_zlecenia_Zapytania ]
GO
ALTER TABLE [dbo].[Szczegoly_zlecenia] WITH CHECK ADD CONSTRAINT [FK_Szczegóły
zlecenia_Pracownicy] FOREIGN KEY([ID_Pracownika])
REFERENCES [dbo].[Pracownicy] ([ID_Pracownika])
GO
ALTER TABLE [dbo].[Szczegoly_zlecenia] CHECK CONSTRAINT [FK_Szczegóły zlecenia_Pracownicy]
GO
ALTER TABLE [dbo].[Wyniki_badania] WITH CHECK ADD CONSTRAINT
[FK_Wyniki_badania_Szczegoly_zlecenia1] FOREIGN KEY([ID_Szczegoly])
REFERENCES [dbo].[Szczegoly_zlecenia] ([ID_Szczegoly])
GO
ALTER TABLE [dbo].[Wyniki_badania] CHECK CONSTRAINT
[FK_Wyniki_badania_Szczegoly_zlecenia1]
GO
ALTER TABLE [dbo].[Zapytania ] WITH CHECK ADD CONSTRAINT [FK_Zapytania _Klienci] FOREIGN
KEY([ID_Klienta])
REFERENCES [dbo].[Klienci] ([ID_Klienta])
GO
ALTER TABLE [dbo].[Zapytania ] CHECK CONSTRAINT [FK_Zapytania _Klienci]
GO
ALTER TABLE [dbo].[Zlecenia ] WITH CHECK ADD CONSTRAINT [FK_Zlecenia _Zapytania ] FOREIGN
KEY([ID_Zlecenia])
REFERENCES [dbo].[Zapytania ] ([ID_Zapytania])
GO
ALTER TABLE [dbo].[Zlecenia ] CHECK CONSTRAINT [FK_Zlecenia _Zapytania ]
GO
ALTER TABLE [dbo].[Badania] WITH CHECK ADD CONSTRAINT [CK_Badania_Czas_realizacji] CHECK
(((Czas_realizacji)>(0)))
GO
ALTER TABLE [dbo].[Badania] CHECK CONSTRAINT [CK_Badania_Czas_realizacji]
GO

```

```

ALTER TABLE [dbo].[Badania] WITH CHECK ADD CONSTRAINT [CK_Badnia_Cena] CHECK
(((Cena_badania)>(0)))
GO
ALTER TABLE [dbo].[Badania] CHECK CONSTRAINT [CK_Badnia_Cena]
GO
ALTER TABLE [dbo].[Badania] WITH CHECK ADD CONSTRAINT [CK_Ilosc_probki] CHECK
(((Ilosc_probki)>(0)))
GO
ALTER TABLE [dbo].[Badania] CHECK CONSTRAINT [CK_Ilosc_probki]
GO
ALTER TABLE [dbo].[Klienci] WITH CHECK ADD CONSTRAINT [CK_ID_Klienta] CHECK
(((len([ID_Klienta])=(4))))
GO
ALTER TABLE [dbo].[Klienci] CHECK CONSTRAINT [CK_ID_Klienta]
GO
ALTER TABLE [dbo].[Klienci] WITH CHECK ADD CONSTRAINT [CK_Klienci_email] CHECK (((email]
like '%_@__%._%'))
GO
ALTER TABLE [dbo].[Klienci] CHECK CONSTRAINT [CK_Klienci_email]
GO
ALTER TABLE [dbo].[Klienci] WITH CHECK ADD CONSTRAINT [CK_Klienci_kod_pocztowy] CHECK
(((Kod_pocztowy] like '___-___'))
GO
ALTER TABLE [dbo].[Klienci] CHECK CONSTRAINT [CK_Klienci_kod_pocztowy]
GO
ALTER TABLE [dbo].[Klienci] WITH CHECK ADD CONSTRAINT [U_NIP_10] CHECK (((len([NIP])=(10)))
GO
ALTER TABLE [dbo].[Klienci] CHECK CONSTRAINT [U_NIP_10]
GO
ALTER TABLE [dbo].[Pracownicy] WITH CHECK ADD CONSTRAINT [CK_data_urodzenia] CHECK
(((datepart(year,getdate())-datepart(year,[Data_urodzenia]))>=(18)))
GO
ALTER TABLE [dbo].[Pracownicy] CHECK CONSTRAINT [CK_data_urodzenia]
GO
ALTER TABLE [dbo].[Pracownicy] WITH CHECK ADD CONSTRAINT [CK_Data_zatrudnienia] CHECK
(((data_zatrudnienia]>[data_urodzenia]))
GO
ALTER TABLE [dbo].[Pracownicy] CHECK CONSTRAINT [CK_Data_zatrudnienia]
GO
ALTER TABLE [dbo].[Pracownicy] WITH CHECK ADD CONSTRAINT [CK_email] CHECK (((email] like
'%_@__%._%'))
GO
ALTER TABLE [dbo].[Pracownicy] CHECK CONSTRAINT [CK_email]
GO
ALTER TABLE [dbo].[Pracownicy] WITH CHECK ADD CONSTRAINT [CK_Kod_pocztowy] CHECK
(((Kod_pocztowy] like '___-___'))
GO
ALTER TABLE [dbo].[Pracownicy] CHECK CONSTRAINT [CK_Kod_pocztowy]

```

```

GO
ALTER TABLE [dbo].[Pracownicy] WITH CHECK ADD CONSTRAINT [CK_Stanowisko] CHECK
(((Stanowisko]='St_Specjalista' OR [Stanowisko]='Mł_Specjalista' OR [Stanowisko]='Specjalista' OR
[Stanowisko]='Kierownik'))
GO
ALTER TABLE [dbo].[Pracownicy] CHECK CONSTRAINT [CK_Stanowisko]
GO
ALTER TABLE [dbo].[Szczegoly_zlecenia] WITH CHECK ADD CONSTRAINT [CK_Badania_Cena]
CHECK (((Cena]>(0)))
GO
ALTER TABLE [dbo].[Szczegoly_zlecenia] CHECK CONSTRAINT [CK_Badania_Cena]
GO
ALTER TABLE [dbo].[Wyniki_badania] WITH CHECK ADD CONSTRAINT [CK_Wyniki_badania__data]
CHECK (((Data_sprawozdania]>=[Data_wykonania_badania]))
GO
ALTER TABLE [dbo].[Wyniki_badania] CHECK CONSTRAINT [CK_Wyniki_badania__data]
GO
ALTER TABLE [dbo].[Wyniki_badania] WITH CHECK ADD CONSTRAINT
[CK_Wyniki_badania_jednostka] CHECK (((Jednostka]='%' OR [Jednostka]='ng/ml' OR
[Jednostka]='mg/ml' OR [Jednostka]='g/kg' OR [Jednostka]='g/L'))
GO
ALTER TABLE [dbo].[Wyniki_badania] CHECK CONSTRAINT [CK_Wyniki_badania_jednostka]
GO
ALTER TABLE [dbo].[Zapytania ] WITH CHECK ADD CONSTRAINT [CK_Zapytania ] CHECK
(((Data_waznosci_oferty]>[Data_przygotowania_oferty] OR [Data_waznosci_oferty] IS NULL))
GO
ALTER TABLE [dbo].[Zapytania ] CHECK CONSTRAINT [CK_Zapytania ]
GO
ALTER TABLE [dbo].[Zapytania ] WITH CHECK ADD CONSTRAINT [CK_Zapytania _datat1] CHECK
(((Data_przygotowania_oferty]<[Data_waznosci_oferty] OR [Data_przygotowania_oferty] IS NULL))
GO
ALTER TABLE [dbo].[Zapytania ] CHECK CONSTRAINT [CK_Zapytania _datat1]
GO
ALTER TABLE [dbo].[Zapytania ] WITH CHECK ADD CONSTRAINT [CK_Zapytania_Status ] CHECK
(((Status]='Zlecone' OR [Status]='Oczekujące' OR [Status]='Wycenione' OR [Status]='Złożone' OR
[Status]='Zrealizowane'))
GO
ALTER TABLE [dbo].[Zapytania ] CHECK CONSTRAINT [CK_Zapytania_Status ]
GO
ALTER TABLE [dbo].[Zlecenia ] WITH CHECK ADD CONSTRAINT [CK_data_dostarczenia] CHECK
(((Data_zlecenia]<=[Data_dostarczenia_probki] OR [Data_dostarczenia_probki] IS NULL))
GO
ALTER TABLE [dbo].[Zlecenia ] CHECK CONSTRAINT [CK_data_dostarczenia]
GO
ALTER TABLE [dbo].[Zlecenia ] WITH CHECK ADD CONSTRAINT [CK_data_faktury] CHECK
(((data_wystawienia_faktury]>=[Data_zatwierdzenia_raportu]))
GO
ALTER TABLE [dbo].[Zlecenia ] CHECK CONSTRAINT [CK_data_faktury]

```

```

GO
ALTER TABLE [dbo].[Zlecenia ] WITH CHECK ADD CONSTRAINT [CK_data_platności] CHECK
((datediff(day,[data_wystawienia_faktury],[data_platnosci])>=(14) OR [Data_platnosci] IS NULL))
GO
ALTER TABLE [dbo].[Zlecenia ] CHECK CONSTRAINT [CK_data_platności]
GO
ALTER TABLE [dbo].[Zlecenia ] WITH CHECK ADD CONSTRAINT [CK_data_raportu] CHECK
(((Data_zatwierdzenia_raportu)>=[Data_dostarczenia_probki] OR [Data_zatwierdzenia_raportu] IS
NULL))
GO
ALTER TABLE [dbo].[Zlecenia ] CHECK CONSTRAINT [CK_data_raportu]
GO
ALTER TABLE [dbo].[Zlecenia ] WITH CHECK ADD CONSTRAINT [CK_Stan_probki] CHECK
((([Stan_probki]='negatywny' OR [Stan_probki]='pozytywny' OR [Stan_probki] IS NULL))
GO
ALTER TABLE [dbo].[Zlecenia ] CHECK CONSTRAINT [CK_Stan_probki]
GO
ALTER TABLE [dbo].[Zlecenia ] WITH CHECK ADD CONSTRAINT [CK_Zlecenia_Status] CHECK
(((Status]='Zakończone' OR [Status]='Wykonane' OR [Status]='W realizacji' OR [Status]='Probka
dostarczona' OR [Status]='Zlecone'))
GO
ALTER TABLE [dbo].[Zlecenia ] CHECK CONSTRAINT [CK_Zlecenia_Status]
GO
/***** Object: StoredProcedure [dbo].[RejestracjaKlienta] Script Date: 26.09.2023 21:10:40
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[RejestracjaKlienta]
    @ID_Klienta NCHAR(4),
    @Nazwa_firmy nvarchar(40),
    @Imie NVARCHAR(50),
    @Nazwisko NVARCHAR(50),
    @Adres_ulica NVARCHAR(30),
    @Adres_numer_domu NVARCHAR(10),
    @Adres_numer_lokalu NVARCHAR(10),
    @ID_Miejscowosc int,
    @Kod_pocztowy char(6),
    @Telefon NVARCHAR(15),
    @email NVARCHAR(40),
    @NIP char(10),
    @Login NVARCHAR(50),
    @Haslo NVARCHAR(50)
AS
BEGIN

```

```
INSERT INTO Klienci (ID_Klienta, Nazwa_firmy, Imie, Nazwisko, Adres_ulica, Adres_numer_domu,
Adres_numer_lokalu, ID_Miejscowosc, Kod_pocztowy, Telefon, email, NIP)
VALUES (@ID_Klienta, @Nazwa_firmy, @Imie, @Nazwisko, @Adres_ulica,
@Adres_numer_domu, @Adres_numer_lokalu, @ID_Miejscowosc, @Kod_pocztowy, @Telefon,
@email, @NIP)
```

```
INSERT INTO Konta_klient (ID_konta, login, haslo)
VALUES (@ID_Klienta, @Login, @Haslo)
```

END

GO

/***** Object: StoredProcedure [dbo].[Zapytanie] Script Date: 26.09.2023 21:10:40 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[Zapytanie]

@ID_Klienta nchar(4)

AS

BEGIN

```
DECLARE @ID_Numer INT
DECLARE @ID_Bad_Mat INT
DECLARE @ID_Zapytania INT
DECLARE @Matryca NVARCHAR(20)
DECLARE @Badanie NVARCHAR(50)
DECLARE @KodProbki NCHAR(7)
DECLARE @Komentarz NVARCHAR(MAX)
```

BEGIN TRANSACTION

IF EXISTS (SELECT TOP 1 * FROM TabelaTymczasowa)

BEGIN

```
INSERT INTO Zapytania (ID_Klienta, Data_zapytania)
VALUES (@ID_Klienta, GETDATE())
```

```
SELECT @ID_Zapytania = SCOPE_IDENTITY()
```

BEGIN

```
WHILE EXISTS (SELECT TOP 1 * FROM TabelaTymczasowa)
SELECT TOP 1 @ID_Numer = Numer, @Badanie = Badanie, @Matryca =
Matryca, @KodProbki = KodProbki, @Komentarz = Komentarz
FROM TabelaTymczasowa
```

ORDER BY Numer

```
SELECT @ID_Bad_Mat = bm.ID_Bad_Mat
FROM Badania_Matryca bm
INNER JOIN Badania b ON bm.ID_Badania = b.ID_Badania
INNER JOIN Matryca m ON bm.ID_Matrycy = m.ID_Matrycy
WHERE b.Nazwa_badania = @Badanie AND m.Nazwa_matrycy = @Matryca
```

```
INSERT INTO Szczegoly_zlecenia (ID_Zapytania, ID_Bad_Mat, Cena, Rabat,
Kod_probki, Komentarz)
VALUES (@ID_Zapytania, @ID_Bad_Mat, NULL, NULL, @KodProbki,
@Komentarz )
```

```
DELETE FROM TabelaTymczasowa WHERE Numer = @ID_Numer
```

END

END

COMMIT TRANSACTION

DROP TABLE TabelaTymczasowa

```
UPDATE Zapytania SET Status = 'Złożone' WHERE ID_Zapytania = @ID_Zapytania
```

END

GO

/***** Object: StoredProcedure [dbo].[Zapytanie_dane_tymczasowe] Script Date: 26.09.2023 21:10:40 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[Zapytanie_dane_tymczasowe]

```
@ID_Klienta nchar(4),
@Badanie NVARCHAR(50),
@Matryca NVARCHAR(20),
@KodProbki NCHAR(7),
@Komentarz TEXT
```

AS

BEGIN

```
IF OBJECT_ID(N'dbo.TabelaTymczasowa', N'U') IS NULL
```

BEGIN

```
CREATE TABLE TabelaTymczasowa ( Numer INT IDENTITY(1, 1) PRIMARY KEY,
Badanie NVARCHAR(50),
```

```
Matryca NVARCHAR(20),
KodProbki NCHAR(7),
Komentarz TEXT
)
    END
    INSERT INTO TabelaTymczasowa (Badanie, Matryca, KodProbki, Komentarz)
    VALUES (@Badanie, @Matryca, @KodProbki, @Komentarz)

END
GO
USE [master]
GO
ALTER DATABASE [u_agomulcz] SET READ_WRITE
```