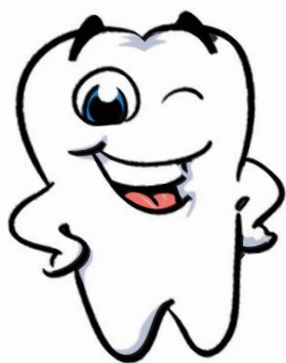




Bazodanowy system obsługi przychodni stomatologicznej

„Dental Clinic Database”



Autorzy: Patrycja Paradowska, Albert Piekielny

Spis treści

1. Wstępne informacje i cele projektu
2. Wymagania funkcjonalne
3. Opis tabel
4. Schemat tabel
5. Prawa dostępu poszczególnych użytkowników
6. Planowane triggery i procedury
7. Diagram przypadków użycia
8. Diagram związków encji
9. Kopia zapasowa oraz archiwizacja

1. Wstępne informacje i cele projektu

Celem jest zaprojektowanie aplikacji, która będzie służyć do przechowywania i przetwarzania danych w oparciu o relacyjną bazę danych. System umożliwi wprowadzanie nowych danych, ale także odpowiednie prezentowanie, segregowanie i modyfikację już istniejących.

W obecnych czasach coraz częściej spotykamy się z trudnościami podczas wizyt w przychodniach lekarskich, a szczególnie stomatologicznych. Długi czas oczekiwania na wolne terminy, brak usprawnionej komunikacji i rejestracji nowych klientów stanowi duży problem, dlatego celem naszej aplikacji jest przeciwdziałanie takim sytuacjom i podniesienie jakości usług poprzez wprowadzenie informatycznego rozwiązania, które zapewni m.in. dużą swobodę w ustalaniu terminów wizyt stomatologicznych. Naszym ograniczeniem jest to, że z aplikacji może korzystać jedynie dyrektor oraz personel (lekarze i recepcjonistki) prywatnej przychodni stomatologicznej, a pacjenci nie mają do niego dostępu. W zależności od zajmowanego stanowiska, system umożliwia bądź blokuje określone operacje i udostępnia użytkownikom odpowiednio przeznaczone dla nich dane.

2. Wymagania funkcjonalne

- Dodawanie nowych pacjentów
- Wyświetlanie lub usuwanie istniejących pacjentów
- Dodawanie, modyfikacja i usuwanie danych o umówionych wizytach w przychodni
- Wyświetlanie i dodawanie dyżurów konkretnych lekarzy
- Wyświetlanie informacji o pacjentach na aktualny dzień
- Dodawanie, usuwanie i wyświetlanie danych lekarzy

- Anulowanie dyżuru poprzez dodanie wiadomości o urlopie
- Anulowanie umówionych wizyt
- Dodawanie, usuwanie dyżurów lekarzy
- Możliwość wypłaty pensji pracownikom przychodni oraz wprowadzanie podwyżek
- Możliwość podsumowania dochodów przychodni w danym miesiącu

3. Opis tabel

Nasza baza danych przechowuje informacje dotyczące pacjentów, personelu przychodni tj. lekarzy i recepcjonistów, oferowanych usług stomatologicznych wraz z ich ceną, dyżurów poszczególnych lekarzy, ale także celu, miejsca i czasu wizyty.

Tabela pracownicy

	Field	Type	Null	Key	Default
1	id_pracownik	int(11)	NO	PRI	<null>
2	Imie	varchar(30)	NO		<null>
3	Nazwisko	varchar(30)	NO		<null>
4	Data_urodzenia	varchar(30)	NO		<null>
5	nr_telefonu	int(9)	NO		<null>
6	Pesel	int(11)	NO	UNI	<null>
7	Pensja	int(11)	NO		<null>
8	Adres	text	NO		<null>

Tabela pracownicy zawiera informacje na temat zatrudnionego w przychodni personelu, czyli dla każdej osoby: imię, nazwisko, datę urodzenia, numer telefonu, pesel, adres. Umieszczona jest także kolumna z informacją o pensji danego pracownika. Każdy z nich posiada identyfikator tzw. numer id, dzięki któremu później może wyszukiwać potrzebne informacje w bazie, np. lekarz o swoich pacjentach, dyżurach i ustalonych wizytach.

Tabela lekarz

	Field	Type	Null	Key	Default	Extra
1	id_lekarz	int(11)	NO	PRI	<null>	auto_increment
2	Specjalizacja	varchar(60)	NO		<null>	

Tabela ta przechowuje informacje o specjalizacjach lekarzy. Id_lekarz jest kluczem obcym odnoszącym się do id_pracownik z tabeli pracownicy.

Tabela dyżury

	Field	Type	Null	Key	Default	Extra
1	id_dyżuru	int(11)	NO	PRI	<null>	auto_increment
2	id_lekarz	int(11)	NO	MUL	<null>	
3	dzien	date	NO		<null>	
4	poczatek	time(2)	NO		<null>	
5	koniec	time(2)	NO		<null>	
6	urlop	enum('tak','nie')	NO		nie	

W tabeli dyżury zawarte są dane na temat dyżurów każdego z lekarzy pracujących w przychodni stomatologicznej m.in. – identyfikator dyżuru, identyfikator danego lekarza, dzień, początek i koniec. Jest także dostępna informacja o urlopie, która przyjmuje wartość domyślną „nie”. Jednak lekarz może otrzymać urlop w danym dniu dyżuru.

Tabela wizyty

	Field	Type	Null	Key	Default	Extra
1	id_wizyty	int(11)	NO	PRI	<null>	auto_increment
2	lekarz	int(11)	NO	MUL	<null>	
3	pacjent	int(11)	NO	MUL	<null>	
4	gabinet	int(11)	NO		<null>	
5	data_wizyty	timestamp	NO		<null>	
6	czas_wizyty	enum('30','60','90','120')	NO		30	
7	cel_wizyty	varchar(40)	NO		<null>	
8	priorytet	enum('1','2','3')	NO		1	
9	status_wizyty	enum('zakończona','oczekująca','do przełożenia')	NO		oczekująca	

Tabela wizyty przechowuje dane na temat umówionych wizyt określonych pacjentów. Każda wizyta ma swój unikalny identyfikator oraz informacje o miejscu (gabinet), czasie oraz celu wizyty, a także o konkretnym lekarzu. Pacjent może być umówiony na kilka wizyt, a lekarz może mieć wiele wizyt jednego dnia, bądź nie mieć żadnej. Domyślny czas jednej wizyty wynosi 30 minut, ale można także umówić się na 60, 90 lub 120 min w przypadku długo trwających zabiegów. Ustalany jest również priorytet wizyty, który domyślnie wynosi 1, ale w bardzo pilnych przypadkach, które muszą być wykonane w trybie natychmiastowym, zwiększamy priorytet na 2 lub 3. Dostajemy także informację o statusie wizyty.

Tabela pacjent

	Field	Type	Null	Key	Default	Extra
1	id_pacjent	int(11)	NO	PRI	<null>	auto_increment
2	Imie	varchar(30)	NO		<null>	
3	Nazwisko	varchar(30)	NO		<null>	
4	Pesel	int(11)	NO	UNI	<null>	
5	nr_telefonu	int(9)	NO		<null>	
6	Data_urodzenia	date	NO		<null>	
7	Adres	varchar(150)	NO		<null>	

Zawiera informacje o zarejestrowanych pacjentach, tzn.: imię, nazwisko, pesel, numer telefonu, data urodzenia oraz adres. Każdy pacjent otrzymuje unikatowy numer identyfikacyjny – id_pacjent.

Tabela usługi

	Field	Type	Null	Key	Default
1	id_wykonawca	int(11)	NO	MUL	<null>
2	cena	int(11)	NO		<null>
3	zabieg	enum('Przegląd','Konsultacja z planem leczenia',...)	NO		Przegląd

Tabela usług zawiera numer identyfikacyjny wykonawcy, czyli stomatologa, który wykonuje daną usługę (zabieg) w określonej cenie. Możliwe usługi to: przegląd, konsultacja z planem leczenia, znieczulenie miejscowe, wypełnienie zęba, leczenie próchnicy, usuwanie kamienia, piaskowanie, wybielanie, usuwanie zęba, założenie aparatu ortodontycznego.

Tabela logi

	Field	Type	Null	Key	Default	Extra
1	id	int(11)	NO	PRI	<null>	auto_increment
2	wizyta_id	int(11)	NO		<null>	
3	Poprzednia_data_wizyty	date	NO		<null>	
4	Nowa_data_wizyty	date	NO		<null>	
5	Poprzedni_lekarz	int(11)	NO		<null>	
6	Nowy_lekarz	int(11)	NO		<null>	
7	Data_zmiany	date	NO		<null>	

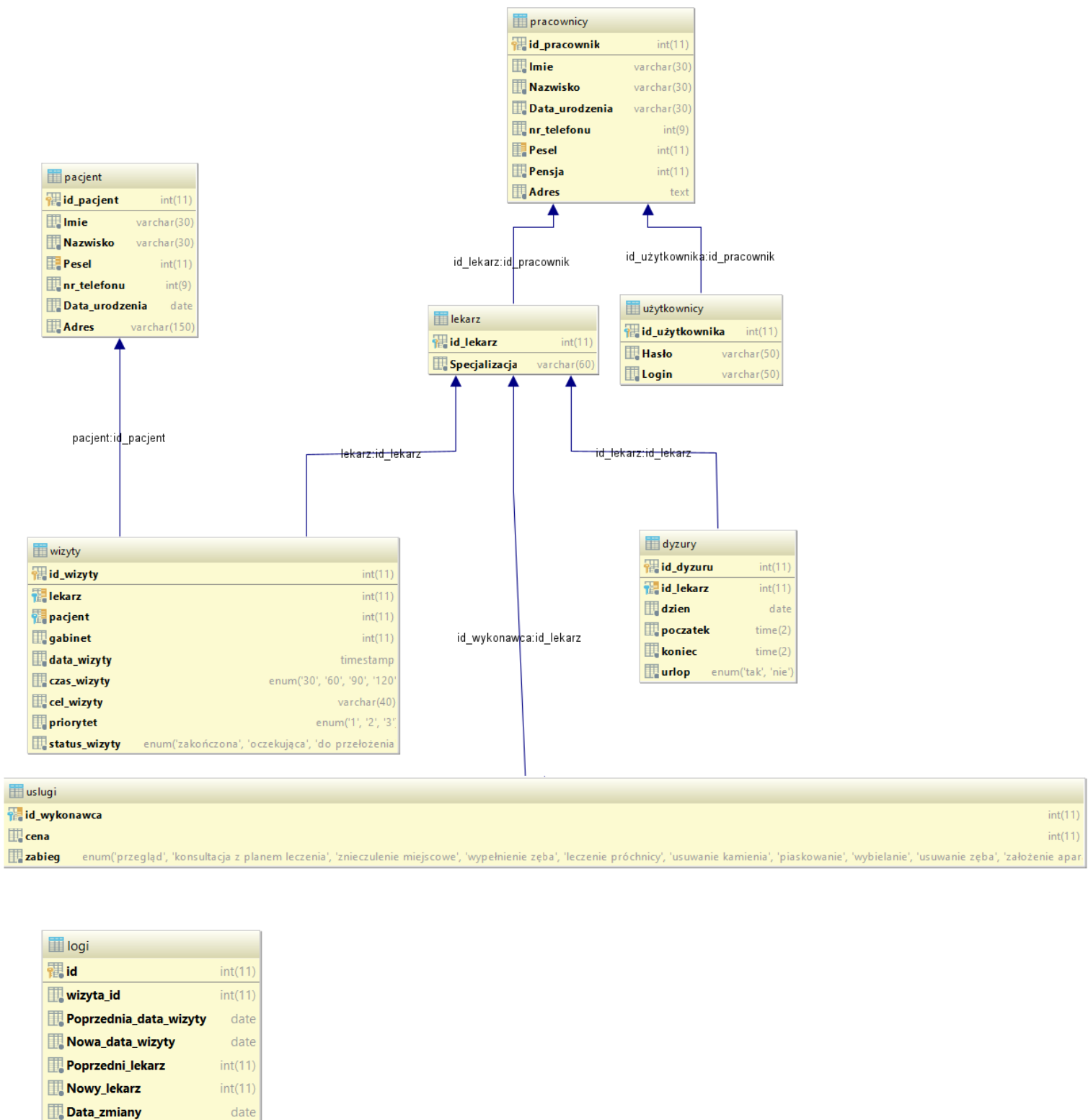
Tabela zawierająca informacje na temat zmian dokonywanych w tabeli „wizyty”, kiedy na przykład na prośbę pacjenta lub lekarza konieczna jest zmiana np. terminu danej wizyty. Wtedy podajemy np. nową datę wizyty lub nowego lekarza, jeśli dany stomatolog ma już wtedy zajęty termin.

Tabela użytkowników

	Field	Type	Null	Key	Default	Extra
1	id_uzytkownika	int(11)	NO	PRI	<null>	auto_increment
2	Hasło	varchar(50)	NO		<null>	
3	Login	varchar(50)	NO		<null>	

Tabela użytkowników zawiera hasła i loginy. Każdy z użytkowników ma swój unikalny identyfikator (id_uzytkownika).

4. Schemat tabel



5. Prawa dostępu

poszczególnych użytkowników

Nasza aplikacja bazodanowa przewiduje następujące poziomy dostępu:

Recepcjonistka

- Posiada prawa do wyświetlania wszystkich danych na temat pacjentów (tabela pacjenci) oraz możliwość rejestracji czyli dodawania nowych pacjentów do bazy danych przychodni, a także modyfikowania ich danych w razie pomyłki itp. Nie może jednak usuwać pacjentów z tabeli.
- Może również usuwać, dodawać nowe wiersze lub modyfikować istniejące w tabeli „wizyty”, ustalając szczegóły dotyczące każdej z nich, np. czas, miejsce, priorytet wizyty, dany stomatolog. Jeśli dane dotyczące prowadzącego lekarza lub terminu jakiejś wizyty zostaną zmienione, to zmiany te są zapisywane w tabeli logi, do której nie ma ona dostępu.
- Recepcjonistka może również jedynie wyświetlać wszystkie dane z tabeli dyżury, ale nie może ich usuwać, dodawać ani modyfikować w żaden sposób (ustalaniem dyżurów zajmuje się dyrektor). Ponadto ma dostęp do jedynie wyświetlania kolumn id_pracownik, imię i nazwisko w tabeli „pracownicy” oraz do wyświetlania całej tabeli „lekarze” i „usługi” (bez modyfikowania ich w żaden sposób), aby móc umówić danego pacjenta do lekarza o określonej specjalizacji i przedstawić potencjalny koszt zabiegu.

Lekarz

- Może wyświetlać wszystkie dane z tabeli „pacjenci”, jednak bez możliwości modyfikowania, usuwania i dodawania.
- Może wyświetlać z tabeli dyżury i wizyty, ale tylko dotyczące jego samego (bez modyfikowania, usuwania, dodawania)

Dyrektor przychodni

Posiada prawa do wszystkich tabel i funkcjonalności. Jako jedyny posiada dostęp do tabeli „użytkownicy”. Zajmuje się np.: ustalaniem urlopów swoich pracowników, wypłacaniem pensji oraz zliczaniem miesięcznego dochodu

przychodni. Może też przeglądać logi zmian wizyt oraz zatrudniać (dodawać) nowych pracowników albo zwalniać (usuwać) istniejących.

6. Planowane triggerzy i procedury

Triggerzy

- Kontrola poprawności numeru PESEL np.: 11 cyfr, zgodność pierwszych sześciu cyfr z datą urodzenia (ale dla osób urodzonych w latach 2000-2099 do numeru miesiąca dodajemy 20) itd.
- Kontrola poprawności numeru telefonu, np.: 9 cyfr, nie może zaczynać się od 0
- Pensja > 0
- Trigger przed dodaniem nowej wizyty, której data nie może być wcześniejsze niż obecna data.
- Trigger kontroli poprawności dyżuru danego lekarza, aby data początku była wcześniejsza niż data końca.
- Trigger przed aktualizowaniem danej wizyty, który doda zmiany (daty wizyty lub lekarza) do tabeli logi.
- Trigger, który przed usunięciem (zwolnieniem) pracownika z tabeli pracownicy, usunie także odpowiedni wiersz z tabeli lekarze.
- Trigger, który przed usunięciem lekarza zmieni status wszystkich jego oczekujących wizyt na „do przełożenia”.
- Trigger, który przed dodaniem nowej wizyty sprawdza najpierw czy dany lekarz ma danego dnia dyżur, następnie czy nie ma wtedy urlopu oraz czy już nie ma o danej godzinie umówionej wizyty.
- Trigger, który po każdej nowej dodanej wizycie zmieni status wszystkich wizyt, które się już odbyły, na „zakończona”.
- Trigger, który po wzięciu urlopu przez lekarza w danych dniach sprawdzi, czy ma on wtedy jakieś umówione wizyty, a jeśli tak to zmieni ich status z „oczekująca” na „do przełożenia”.

Procedury

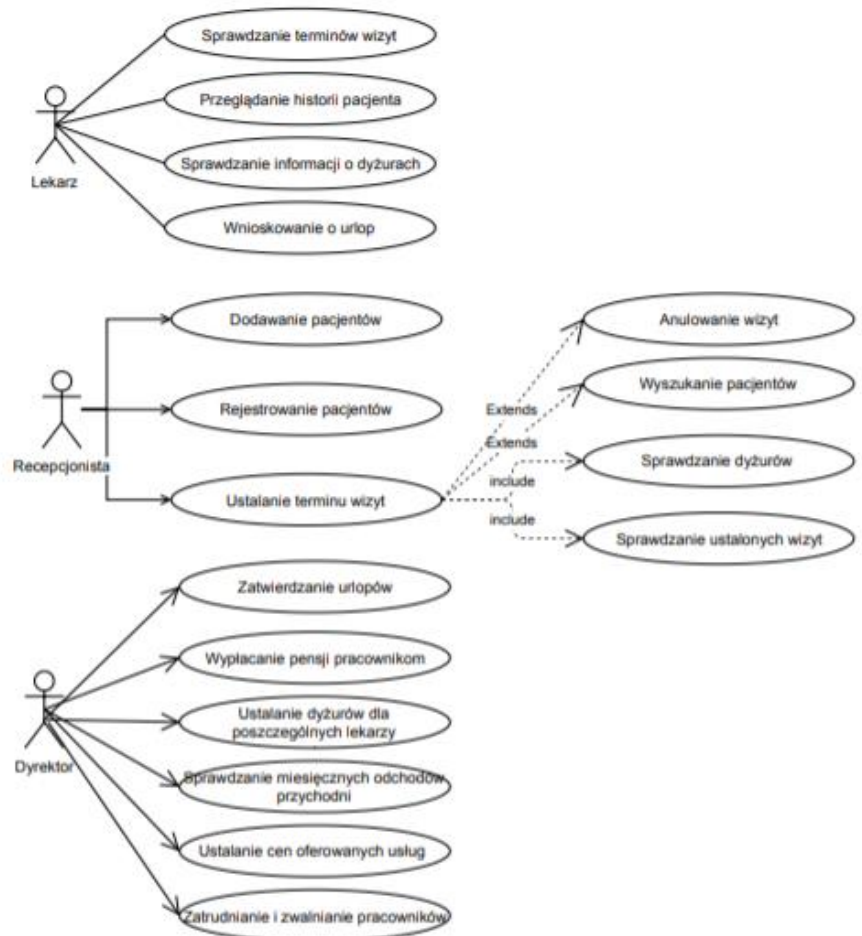
- Procedury tworzące odpowiednie zapytania SQL do wyszukiwania danych w tabelach (np. do sprawdzania czy pacjent, który chce się umówić na wizytę, istnieje już w bazie danych).
- Procedura wypłaty pensji (transakcja) dla personelu przychodni
- Procedura do zliczania miesięcznego dochodu przychodni
- Procedura archiwizacji danych
- Procedury odpowiadające za dodawanie/usuwanie/modyfikację danych (przygotowujące odpowiednie kwerendy SQL z uwzględnieniem kwestii bezpieczeństwa).

7. Diagram przypadków użycia

W naszej aplikacji występują 3 typy aktorów. Są to:

- Lekarze
- Recepcjonistki
- Dyrektor przychodni

Każdy z aktorów ma dostępne inne przypadki użycia w systemie, co świadczy o jego pozycji oraz pozwala nam na skuteczniejszą kontrolę systemu. Dla lekarzy i recepcjonistek dostępna jest węższa wersja systemu, a dyrektor posiada dostęp do wszystkich możliwych uprawnień.



8. Diagram związków encji

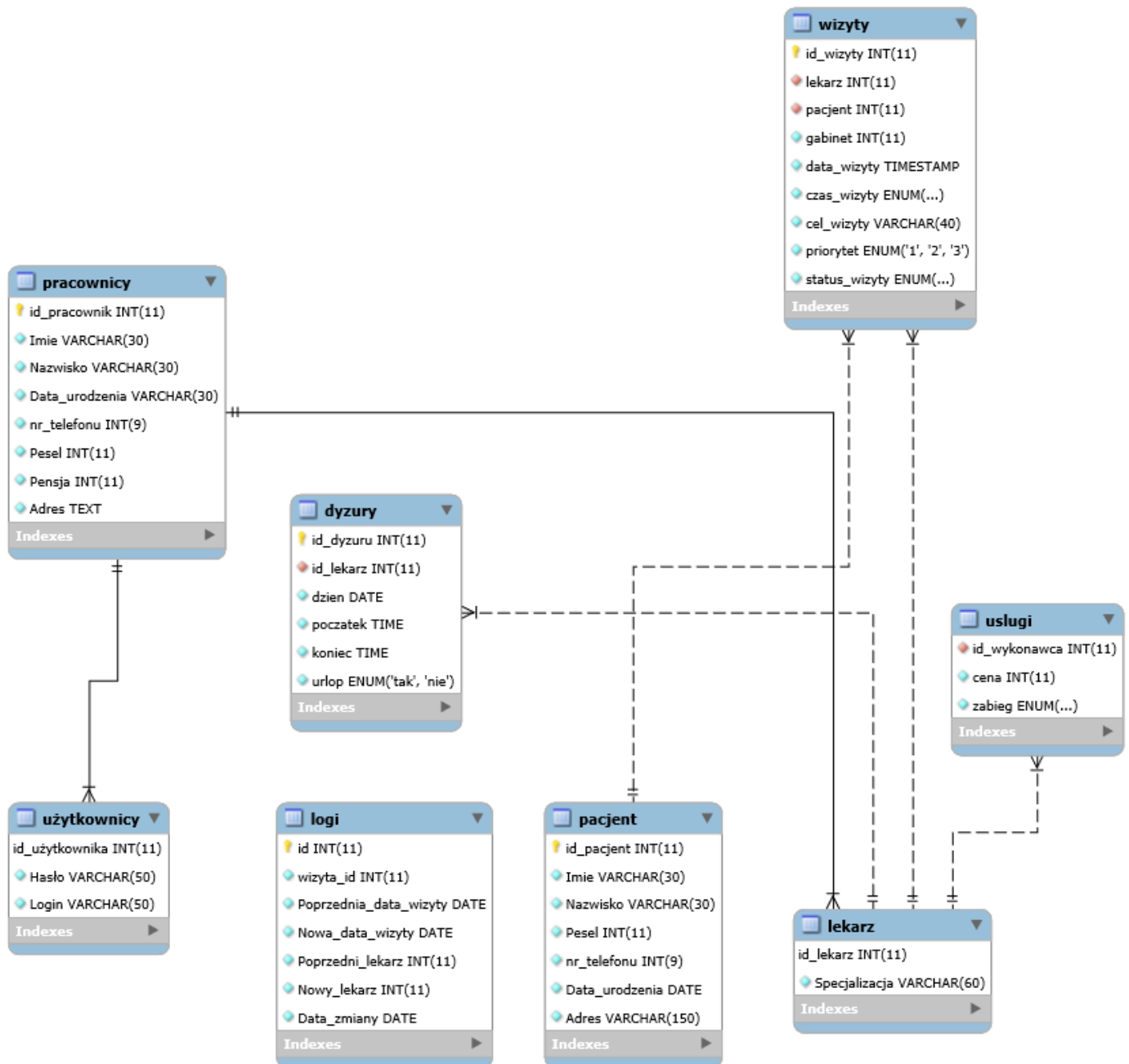


Diagram pokazuje logiczne związki pomiędzy różnymi encjami, związki te mają dwie cechy:

1. **Opcjonalność** – która mówi o tym, czy każda encja **musi**, czy też **może** wystąpić równocześnie z inną. Np. TOWAR **musi** zostać zakupiony przez co najmniej jednego KLIENTA, ale KLIENT **może** być nabywcą TOWARU. W reprezentacji graficznej linia przerywana oznacza opcjonalność związku, natomiast ciągła wymóg związku.

2. **Krotność** – określającą ile encji wchodzi w skład związku:

- a. 1:1 („jeden do jeden”) – encji odpowiada dokładnie jedna encja,
- b. 1:N („jeden do wielu”) – encji odpowiada jedna lub więcej encji,
- c. M:N („wiele do wielu”) – jednej lub więcej encjom odpowiada jedna lub więcej encji.

9. Kopia zapasowa oraz archiwizacja

W naszym projekcie ważnym elementem jest backup&restore, czyli możliwość archiwizacji całej bazy danych lub wybranej części. Przy próbie odtworzenia starej bazy bądź usunięciu wszystkich obecnych danych konieczne jest wykonanie kopii zapasowej usuwanej/podmienianej bazy.