Klinika weterynaryjna

Projekt bazy danych

Dominika Wojtaszewska

SPIS TREŚCI

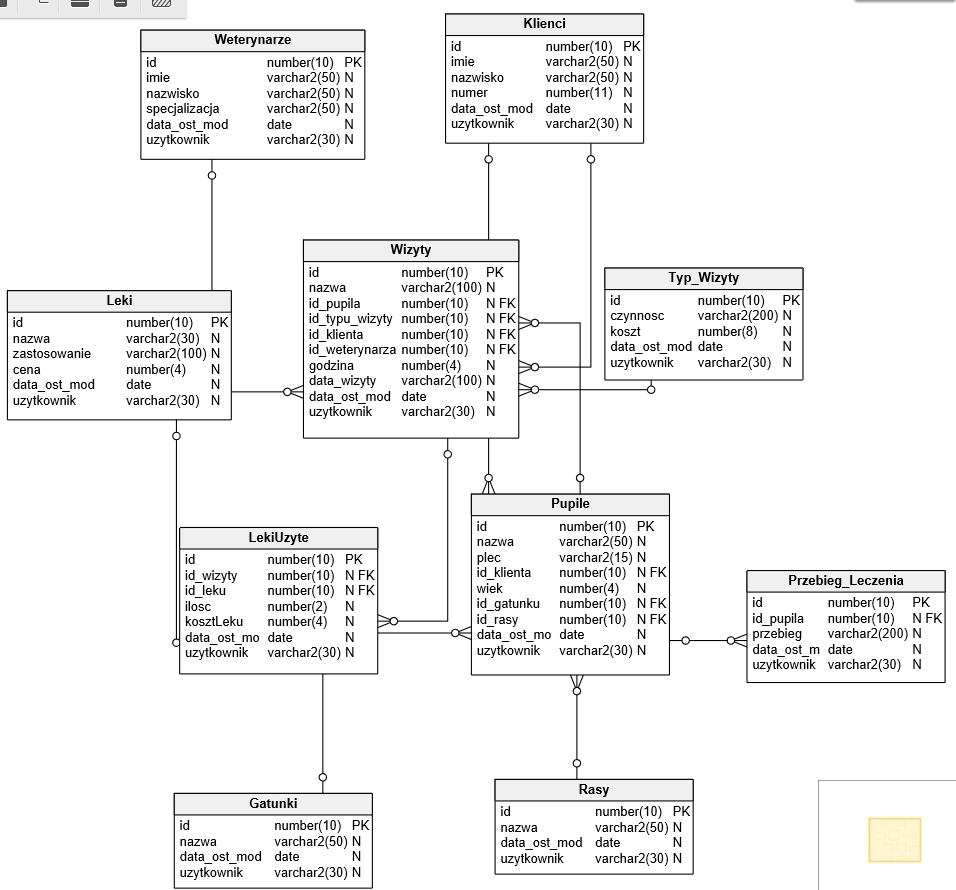
1. Opis problemu biznesowego…………….2
2. Założenia projektowe……………………2
3. Schemat bazy danych……………………3
4. Wymagania funkcjonalne……………..4-5
5. Wymagania niefunkcjonalne……………5
6. Opis działania systemu………………….5
7. Opis zarządzania systemem……………..6
8. Skrypt PL/SQL……………………....6-10
9. Plik eksportu bazy i komponentów w załączniku.
10. **Opis problemu biznesowego.**

Zarządzanie kliniką weterynaryjną może być niezwykle trudne dla jednej osoby i wymaga zatrudnienia większej ilości pracowników/recepcjonistów. Dzięki elektronicznemu systemu zarządzania taką klientą weterynarz jest w stanie wprowadzać sam odpowiednie dane. Taki system musi umożliwiać między innymi dodawanie klientów, ich pupili, przebiegu wizyt, użytych lekarstw, rejestracji wizyt.  
  
**2. Założenia projektowe.**

Projekt będzie miał w sobie bazę danych, odpowiednie procedury, pakiety, ciała pakietów, funkcje oraz interfejs graficzny dla użytkownika systemu. Użytkownik logować się będzie przed załadowaniem strony głównej. Użytkownik systemu będzie tworzony w SQL Plus, a reszta operacji w środowisku Oracle SQL Developer. Baza będzie zawierać 10 tabel : Klienci, Gatunki, Rasy, Weterynarze, Pupile, Typ\_Wizyty (rodzaje wykonów zabiegów), Wizyty, Leki, LekiUzyte (leki użyte podczas jednej wizyty), Przebieg\_Leczenia (opis leczenia pupila).

Dalej - > Schemat bazy danych.

1. **Schemat bazy danych.**



**4. Wymagania funkcjonalne:**

- możliwość dodawania nowych klientów

- możliwość wyświetlania szczegółowych informacji o klientach

- możliwość edycji klientów

- możliwość usuwania klientów

- możliwość wyświetlania wszystkich klientów

- możliwość dodawania nowych gatunków zwierząt

- możliwość wyświetlania szczegółowych informacji o gatunkach

- możliwość edycji gatunków

- możliwość usuwania gatunków

- możliwość dodawania nowych ras zwierząt

- możliwość wyświetlania szczegółowych informacji o rasach

- możliwość edycji ras

- możliwość usuwania ras

- możliwość dodawania nowych weterynarzy

- możliwość wyświetlania szczegółowych informacji o weterynarzach

- możliwość edycji weterynarzy

- możliwość usuwania weterynarzy

- możliwość dodawania nowych zwierząt/pupili

- możliwość wyświetlania szczegółowych informacji o pupilach

- możliwość edycji pupili

- możliwość usuwania pupili

- możliwość dodawania typów wizyt

- możliwość wyświetlania szczegółowych informacji o typach wizyt

- możliwość edycji typów wizyt

- możliwość usuwania typów wizyt

- możliwość dodawania nowych wizyt

- możliwość wyświetlania szczegółowych informacji o wizytach

- możliwość edycji wizyt

- możliwość usuwania wizyt

- możliwość dodawania nowych leków

- możliwość wyświetlania szczegółowych informacji o lekach

- możliwość edycji leków

- możliwość usuwania leków

- możliwość dodawania nowych leków użytych na wizycie

- możliwość wyświetlania szczegółowych informacji o lekach użytych na wizycie

- możliwość edycji leków użytych na wizycie

- możliwość usuwania leków użytych na wizycie

- możliwość dodawania nowych przebiegów leczenia / opisu wizyt dla pupila

- możliwość wyświetlania szczegółowych informacji o przebiegach leczenia

- możliwość edycji przebiegów leczenia

- możliwość usuwania przebiegów leczenia

- możliwość wyszukiwania zwierząt podanego klienta

- możliwość obliczania kosztu za leki

**5. Wymagania niefunkcjonalne:**

- system bazy danych napisany w środowisku Oracle SQL Developer

- system wymaga połączenia z internetem

- system wymaga zainstalowanej przeglądarki internetowej

**6. Opis działania systemu.**

System implementuje interfejs dla użytkownika systemu. Na stronie głównej kliniki znajdują się wszystkie potrzebne odnośniki do zarządzania systemem. Poruszając się po nich otwierają się strony prowadzące do wybranych list rekordów z tabeli, a w nich operacje CRUD. Jeśli chodzi o same tabele to każda z nich posiada mechanizm CRUD, czyli dodawanie pojedynczego rekordu, wyświetlanie szczegółów o pojedynczym rekordzie, edycję pojedynczego rekordu, usunięcie pojedynczego rekordu. Istniejące tabele zostały wymienione już wyżej, a poniżej znajdzie się też ich implementacja w kodzie bazy danych. Dodatkowo istnieje wyszukiwarka klientów w bazie i wyświetlania ich pupili oraz funkcja obliczająca koszt leków użytych na wizycie.

**7. Zarządzanie systemem.**

Użytkownik na samym początku musi się zalogować (przed wejśćiem na stronę główną). Podczas wykonywania operacji Create oraz Edit (dodawania i edycji rekordów) dodawane są informacje o dacie modyfikacji/dodania rekordu oraz nazwa użytkownika, który dodał/dokonał zmian w rekordzie. Na koniec wyświetlane są zmienione/dodane informacje o rekordzie. Podczas poruszania się na stronie użytkownik ma możliwość wracania do poprzednio wczytanej strony poprzez przycisk „Wstecz”.  
  
**8. Skrypt PL/SQL.**

CREATE TABLE Klienci

(

id NUMBER(10),

imie VARCHAR2(50),

nazwisko VARCHAR2(50),

numer NUMBER(11),

data\_ost\_mod DATE,

uzytkownik VARCHAR2(30),

CONSTRAINT klient\_pk PRIMARY KEY (id)

);

CREATE TABLE Gatunki

(

id NUMBER(10),

nazwa VARCHAR2(50),

data\_ost\_mod DATE,

uzytkownik VARCHAR2(30),

CONSTRAINT gatunek\_pk PRIMARY KEY (id)

);

CREATE TABLE Rasy

(

id NUMBER(10),

nazwa VARCHAR2(50),

data\_ost\_mod DATE,

uzytkownik VARCHAR2(30),

CONSTRAINT rasa\_pk PRIMARY KEY (id)

);

CREATE TABLE Weterynarze

(

id NUMBER(10),

imie VARCHAR2(50),

nazwisko VARCHAR2(50),

specjalizacja VARCHAR2(50),

data\_ost\_mod DATE,

uzytkownik VARCHAR2(30),

CONSTRAINT wet\_pk PRIMARY KEY (id)

);

CREATE TABLE Pupile

(

id NUMBER(10),

nazwa VARCHAR2(50),

plec VARCHAR2(15),

id\_klienta NUMBER(10),

wiek NUMBER(4),

id\_gatunku NUMBER(10),

id\_rasy NUMBER(10),

data\_ost\_mod DATE,

uzytkownik VARCHAR2(30),

CONSTRAINT pupil\_pk PRIMARY KEY (id),

CONSTRAINT pupil\_klient\_fk FOREIGN KEY (id\_klienta) REFERENCES Klienci (id),

CONSTRAINT pupil\_gatunek\_fk FOREIGN KEY (id\_gatunku) REFERENCES Gatunki (id),

CONSTRAINT pupil\_rasa\_fk FOREIGN KEY (id\_rasy) REFERENCES Rasy (id)

);

CREATE TABLE Typ\_Wizyty

(

id NUMBER(10),

czynnosc VARCHAR2(200),

koszt NUMBER(8),

data\_ost\_mod DATE,

uzytkownik VARCHAR2(30),

CONSTRAINT typ\_wiz\_pk PRIMARY KEY (id)

);

CREATE TABLE Wizyty

(

id NUMBER(10),

nazwa VARCHAR2(100),

id\_pupila NUMBER(10),

id\_typu\_wizyty NUMBER(10),

id\_klienta NUMBER(10),

id\_weterynarza NUMBER(10),

godzina NUMBER(4),

data\_wizyty VARCHAR2(100),

data\_ost\_mod DATE,

uzytkownik VARCHAR2(30),

CONSTRAINT wizyty\_pk PRIMARY KEY (id),

CONSTRAINT wizyty\_klient\_fk FOREIGN KEY (id\_klienta) REFERENCES Klienci (id),

CONSTRAINT wizyty\_pupil\_fk FOREIGN KEY (id\_pupila) REFERENCES Pupile (id),

CONSTRAINT wizyty\_typ\_fk FOREIGN KEY (id\_typu\_wizyty) REFERENCES Typ\_Wizyty (id),

CONSTRAINT wizyty\_wet\_fk FOREIGN KEY (id\_weterynarza) REFERENCES Weterynarze (id)

);

CREATE TABLE Leki

(

id NUMBER(10),

nazwa VARCHAR2(30),

zastosowanie VARCHAR2(100),

cena NUMBER(4),

data\_ost\_mod DATE,

uzytkownik VARCHAR2(30),

CONSTRAINT lek\_pk PRIMARY KEY (id)

);

CREATE TABLE LekiUzyte

(

id NUMBER(10),

id\_wizyty NUMBER(10),

id\_leku NUMBER(10),

ilosc NUMBER(2),

kosztLeku NUMBER(4),

data\_ost\_mod DATE,

uzytkownik VARCHAR2(30),

CONSTRAINT leku\_pk PRIMARY KEY (id),

CONSTRAINT leku\_wizyta\_fk FOREIGN KEY (id\_wizyty) REFERENCES Wizyty (id),

CONSTRAINT leku\_fk FOREIGN KEY (id\_leku) REFERENCES Leki (id)

);

CREATE TABLE Przebieg\_Leczenia

(

id NUMBER(10),

id\_pupila NUMBER(10),

przebieg VARCHAR2(200),

data\_ost\_mod DATE,

uzytkownik VARCHAR2(30),

CONSTRAINT przebieg\_pk PRIMARY KEY (id),

CONSTRAINT przebieg\_pupil\_fk FOREIGN KEY (id\_pupila) REFERENCES Pupile (id)

);