

Dokumentacja - Bazy Danych

Karol Hoerner de Roithberg

17 Styczeń 2023

1 Założenia, koncepcja projektu

1.1 Temat projektu

Strona internetowa "Paczka" ma za zadanie pozwolić użytkownikowi wysyłać paczki oraz kurierowi zarządzanie systemem.

1.2 Analiza wymagań użytkownika

Użytkownik:

- Wysyłanie paczki (nadawca oraz adresat muszą być zarejestrowani w bazie), do wyboru miejsce (paczkomat z listy lub dom przypisany do użytkownika)
- Rejestracja użytkownika
- Sprawdzenie statusu oraz danych paczki po ID

Kurier:

Po zalogowaniu się

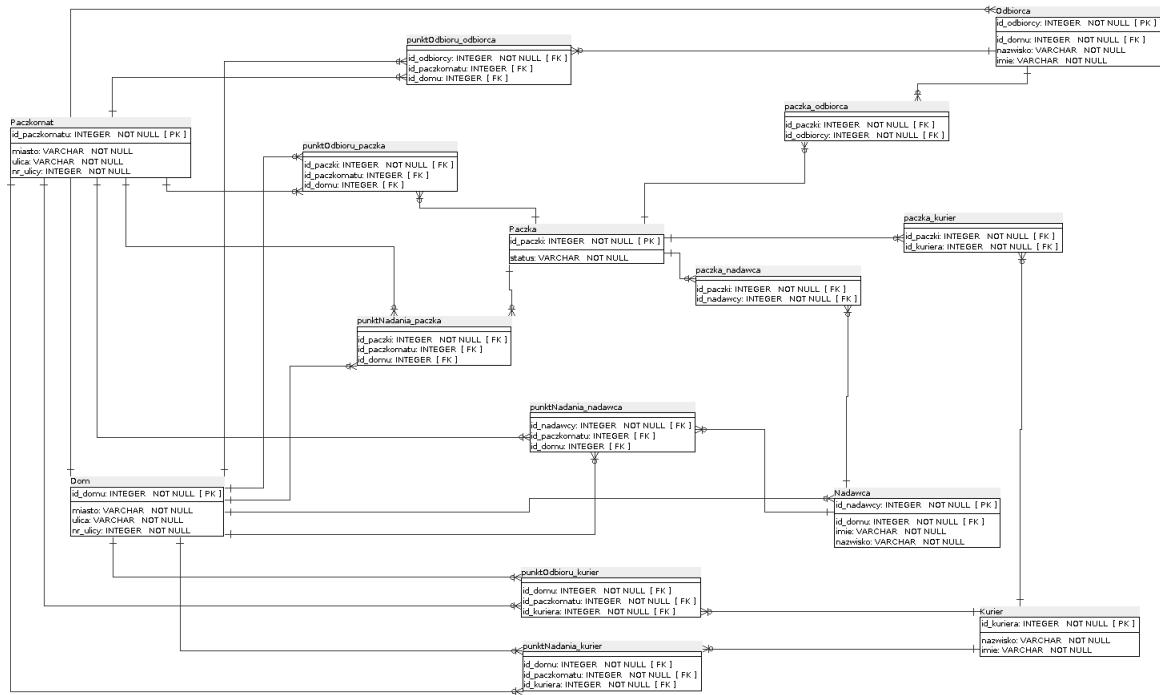
- Dodanie nowego paczkomatu
- Dodanie nowego kuriera
- Zmiana statusu paczek przypisanych do obecnie zalogowanego kuriera
- Wyświetlenie wszystkich paczek obecnie zalogowanego kuriera

1.3 Zaprojektowanie funkcji

Funkcje realizowane w bazie danych służą do zapisu, aktualizacji, wypisywania poprawnych danych oraz do zapobiegania dodawaniu niepoprawnych danych.

2 Projekt diagramów

2.1 Budowa diagramu ERD



Rysunek 1: Diagram ERD (Dla lepszej widoczności załączony również w plikach projektu).

2.2 Zdefiniowanie encji

Encje:

- Paczka
- Odbiorca
- Nadawca
- Kurier
- Paczkomat
- Dom

Encje asocjacyjne:

- paczka_nadawca
- paczka_odbiorca
- paczka_kurier

- punktNadania_nadawca
- punktOdbioru_odbiorca
- punktNadania_paczka
- punktOdbioru_paczka
- punktOdbioru_kurier
- punktNadania_kurier

2.3 Zaprojektowanie relacji pomiędzy encjami

Większość relacji między tabelami to relacje 1 : N . Relacje opisane encjami asocjacyjnymi o nazwach (punktNadania_kurier, punktOdbioru_kurier, punktOdbioru_odbiorca, punktNadania_nadawca) są relacjami $M : N$.

3 Projekt Logiczny

3.1 Projektowanie tabel, kluczy, indeksów

Encja **Paczka** - reprezentuje dodaną do systemu paczkę:

- id_paczki - klucz główny, typu INTEGER
- _status - atrybut przechowujący status paczki, typu VARCHAR

Encja **Nadawca** - reprezentuje dodanego do systemu nadawcę:

- id_nadawcy - klucz główny, typu INTEGER
- imie - atrybut przechowujący imię nadawcy, typu VARCHAR
- nazwisko - atrybut przechowujący nazwisko nadawcy, typu VARCHAR
- id_domu - atrybut przechowujący ID domu nadawcy, typu INTEGER

Encja **Odbiorca** - reprezentuje dodanego do systemu odbiorcę:

- id_odbiorcy - klucz główny, typu INTEGER
- imie - atrybut przechowujący imię odbiorcy, typu VARCHAR
- nazwisko - atrybut przechowujący nazwisko odbiorcy, typu VARCHAR
- id_domu - atrybut przechowujący ID domu odbiorcy, typu INTEGER

Encja **Kurier** - reprezentuje dodanego do systemu kuriera:

- id_kuriera - klucz główny, typu INTEGER
- imie - atrybut przechowujący imię kuriera, typu VARCHAR
- nazwisko - atrybut przechowujący nazwisko kuriera, typu VARCHAR

Encja **Paczkomat** - reprezentuje dodany do systemu paczkomat:

- id_paczkomatu - klucz główny, typu INTEGER
- miasto - atrybut przechowujący miasto paczkomatu, typu VARCHAR
- ulica - atrybut przechowujący ulicę paczkomatu, typu VARCHAR
- nr_ulicy - atrybut przechowujący nr. ulicy paczkomatu, typu INTEGER

Encja **Dom** - reprezentuje dodany przez użytkownika dom:

- id_domu - klucz główny, typu INTEGER
- miasto - atrybut przechowujący miasto domu, typu VARCHAR
- ulica - atrybut przechowujący ulicę domu, typu VARCHAR
- nr_ulicy - atrybut przechowujący nr. ulicy domu, typu INTEGER

3.2 Słowniki danych

Typy danych użyte w aplikacji to wbudowane w PostgreSQL typy danych z zastrzeżeniami NOT NULL.

3.3 Zaprojektowanie operacji na danych

Użytkownik aplikacji może dodawać wartości do tabel poprzez formularze oraz wyświetlać raporty dotyczące danych w bazie. Raporty zasadniczo oparte są o widoki. Widoki zostały utworzone aby łatwiej można było operować na danych.

Kod wszystkich widoków znajduje się w pliku 'views.sql' w folderze 'sql'. **Widoki w bazie**

- paczkomat_view
- uzytkownik_view
- paczka_dodanie_view
- kurier_login_view
- kurier_dodaj_view

Zaprojektowane zostały także funkcje oraz wyzwalacze wypisujące dane, dodające dane lub zapobiegające wprowadzeniu niepoprawnych danych do tabel.

Funkcja	Wyzwalacz	Tabela/Widok
paczka_wypisz(id)	-----	-----
paczki_wypisz_kurier(imie, nazwisko)	-----	-----
dodaj_paczkomat()	paczkomat_dodaj	paczkomat_view
dodaj_kuriera()	kurier_dodaj	kurier_dodaj_view
kurier_login()	login_kurier	kurier_login_view
dom_validate()	dodaj_view_paczka	paczka_dodanie_view
dom_validate()	validate_dom	dom
uzytkownik_validate()	validate_uzytkownik	nadawca
uzytkownik_dodaj()	dodaj_uzytkownik	nadawca
uzytkownik_view_dodaj()	dodaj_view_uzytkownik	uzytkownik_view

Tabela 1: Zależność funkcji od triggerów oraz tabel/widoków na których są wykonywane

Kod wszystkich funkcji oraz wyzwalaczy znajduje się w pliku 'functions.sql' w folderze 'sql'.

W funkcjach sql-owych zablokowano wpisywanie niepoprawnych wartości (tam gdzie się dało).

Serwer aplikacji został napisany w frameworku Express.js, serwer wysyła zapytania query do bazy danych oraz obsługuje wypisywanie danych z bazy danych. W serwerze wykorzystano templatki EJS oparte na języku html.

Język javascript został użyty do odznaczania miejsca wysyłki oraz odbioru przy dodawaniu paczki.

Wygląd strony zmodyfikowany został za pomocą języka css.