

Manager lotniczy - projekt BD1 - dokumentacja

Krzysztof Czerwiński

Styczeń 2023

1 Projekt logiczny

W projekcie zdefiniowane zostały następujące encje, które tworzą relację 1-1, 1-N, M-N:

1. Samolot
2. Linia lotnicza
3. Rejs (zgodnie z rzeczywistością Rejs określa trasę z punkta A do B realizowaną w danych godzinach przed daną linię lotniczą, nie zawiera informacje o dacie, samolocie)
4. Lot (zawiera informacje o tym kiedy odbędzie się dany rejs, jakim samolotem oraz inne szczegóły)
5. Pracownik
6. Bilet
7. Dostęp pracownika
8. Lotnisko

Ponadto stworzonych zostało jeszcze 8 tabel, które realizują relacje pomiędzy encjami, lub dzięki nim unikami redundancji danych(kategorie sklepów/ funkcje pracowników jako osobne tabele).

Poszczególne encje połączone są kluczami obcymi jak pokazano na diagramie.

1.1 Walidacja danych

Dane w bazie są walidowane przy użyciu:

1. constraintow - sprawdzanie dlugosci wprowadzonych danych, sprawdzenie czy dane są unikatowe z pomocą klauzuli check(sprawdzenie czy dla danego lotu wprowadzono 2 innych pilotow) oraz innych warunków biznesowych(np. czy wprowadzono przynajmniej numer jednego z dokumentow tożsamosci dowodu/paszportu)

Dane są oczywiście walidowane również w aplikacji zarówno po stronie serwera jak i klienta.

1.2 Dostęp do danych

W zależności czy z aplikacji korzysta użytkownik zalogowany(admin) czy klient używamy jedną z dwóch ról, które specyfikują dostęp do danych:

1. app_admin - ma możliwość edycji większości tabel, oraz korzystania ze wszystkich widoków i funkcji
2. app_user - ma możliwość selekcji tylko z tabel, widoków i funkcji, które używane są przez aplikację, ma możliwość wstawiania tylko i wyłącznie do tabeli bilet-przy pomocy funkcji

Przełączenie użytkownika bazy danych następują w momencie zalogowania/wylogowania.

1.3 Sortowanie i paginacja

Aby ułatwić przeszukiwanie tabel użytkownik ma zapewnioną możliwość sortowania i paginacji danych wyświetlanych w aplikacji. Jest to realizowane po stronie danych przy pomocy widoków oraz funkcji wbudowanych(używających widoków). Dzięki temu nasza aplikacja lepiej wygląda, i nie obciąża niepotrzebnie bazy danych i serwera bazy danych ściągając duże ilości niepotrzebnych danych. Paginacja odbywa się przy pomocy LIMIT i OFFSET, żeby zapewnić sortowanie po każdej wyświetlanej kolumnie używamy dynamicznych QUERY, zamkniętych w funkcjach aby zminimalizować ryzyko SQL injection.

1.4 Filtrowanie

W kluczowych tabelach wprowadziliśmy również możliwość filtrowania danych, odbywa się to przy pomocy funkcji wbudowanych.

1.5 Widoki

Aby zapewnić możliwość komfortowego pobierania danych dla klienta zostało stworzonych wiele widoków. Widoki te łączą dane z wielu tabel przy pomocy JOIN oraz formatują do postaci przyjaznej dla klienta. Wiele z widoków korzysta z wyrażeń CTE aby zapewnić łatwą agregację danych z innych tabel.

W widokach generujących raporty wykorzystane są klauzule GROUP BY oraz HAVING.

1.6 Struktura tabel i możliwość rozbudowy

Tabele i relacje pomiędzy nimi zostały tak stworzone, aby zapewnić możliwość rozwoju. Projekt bazy danych można rozbudować o między innymi:

1. Tabele lot_zaloga - zawierającą pracowników, którzy operują na danym locie
2. Informację o pracownikach mających dostęp do systemu Manager Lotniczy

3. Oddziały linii lotniczych na lotniskach
4. Informacje na biletach o rezerwacji poszczególnych miejsc w samolocie i wiele innych.

1.7 Ograniczenia

Projekt bazy posiada również pewne ograniczenia, wynikają one w większości z faktu, iż stworzenie ich w optymalny sposób wymagało by bardzo dużego nakładu czasu. Poniżej wymieniono ograniczenia naszego projektu bazy danych (których jestem świadomy):

1. w przypadku pasażerów podróżujących wielokrotnie ich dane w tabeli bilet są powtarzane - problem zostałby rozwiązany tworząc system logowania klientów, i zapisywania ich danych w osobnej tabeli
2. nie ma walidacji czy piloci lądują na lotnisku, do którego mają dostęp - można by rozwiązać prostą walidacją w bazie, lecz wymagany by był duży nakład pracy w aplikacji

1.8 Normalizacja

Tabele zostały sprawdzone zgodnie z 3 postaciami normalnymi

1.8.1 Pierwsza postać normalna

W tym kroku stworzyliśmy kilka nowych tabeli, aby uniknąć redundancji:

1. Z tabeli sklep wyłączyliśmy tabelę sklep_kategoria zawierającą informacje o kategorii sklepu
2. dla każdego z typów pracownika wyłączyliśmy osobną tabelę z funkcjami dla danego typu pracownika

1.8.2 Druga postać normalna

W tym kroku wykonaliśmy następujące operacje:

1. Zamiast tworzyć 3 tabele (pracownik lotniska, pracownik sklepu, pracownik linii) stworzyliśmy jedną tabelę zawierającą informację o wszystkich pracownikach, klucz główny w tej tabeli jest używany w osobnych tabelach, aby określić funkcje pracowników, miejsca pracy jak i dostęp

1.8.3 Trzecia postać normalna

W przypadku tej zasady nie zdecydowałem się na kolejne podziały tabel.

2 Projekt Funkcjonalny

W ramach tego podpunktu omówione zostaną funkcje, na które pozwala aplikacja.

2.1 Funkcje dla klienta

1. listowanie lotów, kupno biletu
Klient ma możliwość wyszukać loty poprzez zdefiniowany filtr zawierający lotnisko odlotu i przylotu, cenę, terminy. Oprócz tego jest możliwość sortowania po każdej z kolumn oraz paginacji. Po kliknięciu na interesujący lot otwiera się popup, który umożliwia zakup biletu.
2. listowanie sklepów
Klient ma możliwość filtracji sklepów po lotniskach i kategoriach sklepu. Do filtru zaciągane są dane z bazy danych, i klient ma możliwość wyboru tylko z istniejących lotnisk i kategorii, jest również opcja wszystkie sklepy.

2.2 Funkcje admina

Admin po zalogowaniu się do aplikacji (domyślne dane logowania to login: admin, hasło: admin) ma możliwość.

1. wyświetlania, sortowania, paginacji lotów, dodawania nowych lotów
2. wyświetlania, sortowania, paginacji linii lotniczych, dodawania nowych linii lotniczych
3. wyświetlenia 3 raportów: lotnisko-pracownicy, linia lotnicza, lotnisko-kategorie sklepów
4. wyświetlania, sortowania, paginacji rejsów, dodawanie nowych rejsów
5. wyświetlania sortowania, paginacji lotów, dodawania lotów (specjalny formularz używający AJAX do polepszenia user experience i ułatwienia wybrania interesujących nas opcji)
6. wyświetlania, sortowania, paginacji, filtracji funkcji - pracowników
Po kliknięciu na pracownika w tabeli otrzymujemy listę lotnisk, do których pracownik ma dostęp.
Możemy również dodać nowy dostęp dla pracownika.
7. wyświetlania, sortowania, paginacji, filtracji sklepów, dodawania nowych sklepów
8. wyświetlania, sortowania, paginacji samolotów, dodawanie nowych samolotów

Encje są ze sobą wzajemnie powiązane dlatego w poszczególnych formularzach wykonujemy zapytania do serwera, aby otrzymać dostępne opcje, np. listę lotnisk.

Wszystkie formularze są walidowane po stronie klienta, walidujemy również po stronie bazy danych.

3 Dokumentacja

Przykładowe dane do bazy danych możemy wprowadzić przy pomocy skryptów `create_*` dołączonych jako pliki `.sql` do projektu.