

Projektowanie bazy danych - Projekt

Spis treści

Podstawy projektowania baz danych 1

Poniższe ćwiczenie będzie polegało na tym, że w pierwszej kolejności powinieneś / powinnaś zapoznać się z materiałami, które zostały przez nas przygotowane / zgromadzone. Następnie będziemy mogli przejść do realizacji zadania.

Podstawy projektowania baz danych

Jest takie powiedzenie:

Im dalej w las, tym więcej drzew trzeba na ścieżce Zajavka rąbać samodzielnie.

A jak nie ma takiego powiedzenia, to już jest ☺.

Na tym etapie na pewno już udało Ci się zauważyć, że praca programisty polega na rozwiązywaniu problemów i szukaniu rozwiązań. Nam jako twórcom Zajavki zależy na tym, żeby pokazywać Ci, jak wygląda praca programisty w praktyce. Dlatego, co jakiś czas będziemy wprowadzali wstawki / ćwiczenia, które mają dać Ci możliwość znalezienia jakiejś wiedzy na własną rękę. Nie zostawimy Cię oczywiście z niczym. Na koniec takiego ćwiczenia znajdziesz link do artykułu albo gotowe rozwiązanie od nas, w którym będzie opisane zagadnienie, na którym nam zależy, żebyś sobie przyswoił / przyswoiła.

W ramach tego ćwiczenia będziemy chcieli rozwiązać problem, jakim jest zaprojektowanie bazy danych. W poprzednich materiałach pracowaliśmy już z bazą danych (projekt **zajavka store**), ale nie zagłębialiśmy się w tematykę tego, jak podejść do jej zaprojektowania, na co zwracać uwagę. Baza danych była już w poleceniu "zaprojektowana". Musieliśmy jedynie napisać odpowiednie DDL. Tym razem, przejdziemy przez proces myślowy projektowania takiej bazy danych.

Co ma do tego wszystkiego wcześniejsze powiedzenie z lasem? Zanim przejdziemy do realizacji zadania, chcemy dać Ci możliwość poćwiczenia szukania wiedzy na własną rękę. Oczywiście przedstawimy też linki do artykułów, które są *must have* przed realizacją zadania.



Zdajemy sobie sprawę, że nasze podejście w pierwszym momencie może wywołać frustrację. Mamy jednak nadzieję, że zrozumiesz, dlaczego tak podchodzimy do tematu i nam wybaczysz, gdy będziesz już mieć problem w praktyce i czeka Cię wtedy dużo researchu. Chcemy Cię przyzwyczaić i przygotować do szukania rozwiązań w Internecie.

W ramach tego ćwiczenia zrobimy kilka rzeczy:

1. Chcę, żebyś najpierw zaczął/zaczęła od zrobienia researchu na temat projektowania baz danych.

Zacznij od poszukania informacji na ten temat samodzielnie. Podpowiem, że zależy nam na znalezieniu informacji na temat tego, czym jest **Diagram ERD** oraz na co zwracać uwagę przy projektowaniu baz danych.

2. Jeżeli masz dosyć szukania, to zapoznaj się z tymi artykułami:

- [link](#) - jak można koncepcyjnie podejść do projektowania baz danych,
- [link](#) - podstawy projektowania baz danych,
- [link](#) oraz [link](#) - czym jest Diagram ERD.

3. Mając już zrobiony research, możesz przejść do zaprojektowania bazy danych, która będzie rozwiązywała poniższy problem (wyobraź sobie, że poniższy cytat to są wymagania jakie otrzymałeś / otrzymałaś do realizacji):

Stwórz **Physical ERD** oraz **DDL queries** dla aplikacji dealera samochodowego. Dealer sprzedaje nowe oraz używane samochody i prowadzi serwis. Projekt ma spełniać następujące reguły biznesowe:

- Sprzedawca może sprzedawać wiele samochodów, ale każdy samochód jest sprzedawany tylko przez jednego sprzedawcę,
- Sprzedawca wystawia jedną fakturę za każdy sprzedany samochód,
- Klient otrzymuje fakturę za każdy zakupiony samochód,
- Klient nie musi kupować samochodu, aby zostać sklasyfikowanym jako klient, czyli jeżeli ktoś przyjdzie oddać auto do serwisu, to również jest klientem dealera samochodowego,
- Jeżeli klient przyjeżdża oddać auto do serwisu, to jeden klient może oddać do serwisu jeden lub więcej samochodów, ale na każdy samochód tworzone jest oddzielne zgłoszenie serwisowe.
- Klient może kupić wiele samochodów, ale każdy może być zakupiony tylko przez jednego klienta.
- Jeden samochód może być obsługiwany w serwisie przez wielu mechaników, do tego każdy mechanik może obsługiwać wiele samochodów.
- Samochód, który został oddany do serwisu może, ale nie musi potrzebować części do naprawy. Czyli, może zostać wykonana naprawa wymagająca części, np. wymiana oleju, ale serwis może też wykonać naprawę niewymagającą części, np. regulacja turbosprężarki.
- Dealer prowadzi historię serwisową dla każdego serwisowanego auta. Baza danych powinna pozwolić na znalezienie historii serwisowej samochodu na podstawie jego numeru VIN.

Projekt ten nazwijmy **Car Dealership**. Tak samo, jak wcześniej pracowaliśmy z projektem, który został nazwany **zajavka store**, tak ten projekt będzie nosił nazwę **Car Dealership**. Jest to o tyle istotne, że później będziemy odwoływać się do tej nazwy.



Mała uwaga dotycząca wyglądu Physical ERD, który będzie dostarczony jako

przykładowe rozwiązanie. W dostarczonych linkach przedstawiony został pewien schemat rysowania diagramów ERD. W praktyce natomiast (tak samo, jak w przypadku UML) w każdej firmie i na każdym projekcie może się zdarzyć tak, że diagramy takie będą wyglądały inaczej, będą rysowane zgodnie z innym schematem lub będą wykorzystywały inne oznaczenia.

My oczywiście chcemy Cię do tego przygotować, dlatego właśnie dostarczony przez nas diagram Physical ERD będzie wyglądał inaczej niż w dostarczonych przez nas linkach (do zapoznania się) i nadal będziemy go nazywać Physical ERD. Istotą tego zagadnienia jest przygotowanie diagramu, który będzie reprezentował model bazy danych. To, czy stosujemy takie, czy inne strzałki jest już kwestią wtórną 😊. Ważne jest to, żeby każdy członek zespołu rozumiał znaczenie diagramu.