



# Node.js MySQL

## SPRAWDZIAN

© PIOTR SIEWNIAK 2024

### Program pomocniczy 1

Napisać program pozwalający na utworzenie na serwerze MySQL (lub MariaDB) bazy danych o nazwie `baza_auto`. W bazie `baza_auto` utworzyć tabelę o nazwie `auto` posiadającą następujące kolumny: `id`, `marka`, `model`, `rok_produkcji`, `data_pierwszej_rejestracji`, `cena`, `stan`, `nr_rejestracyjny`.

### Program pomocniczy 2

Napisać program pozwalający na dopisanie do tabeli `auto` w bazie danych `baza_auto` danych co najmniej 10 samochodów pobranych z pliku tekstowego w formacie JSON.

### Zadanie 1

Napisać program pozwalający na prezentację danych wybranych samochodów z tabeli `auto` w bazie danych `baza_auto`, spełniających zadane kryteria, np.:

- a) `marka`;
- b) `marka` i `model`;
- c) `marka`, `model` i `cena`.

### Zadanie 2

Napisać program pozwalający na dopisanie danych nowego samochodu do tabeli `auto` w bazie danych `baza_auto`.

### Zadanie 3

Napisać program pozwalający na modyfikację danych wybranego samochodu w tabeli `auto` w bazie danych `baza_auto`. Założyć, że modyfikowane pole to `cena`, a samochód wybierany jest na podstawie numeru rejestracyjnego.

### Zadanie 4

Napisać program pozwalający na usunięcie z tabeli `auto` w bazie danych `baza_auto` danych wybranego samochodu. Założyć, że samochód jest wybierany na podstawie numeru rejestracyjnego.

### Zadanie dodatkowe\*

Założyć, że samochody są sprzedawane w trzech komisach – filiach: w Sosnowcu, Katowicach i Gliwicach. Zaprojektować i zaimplementować kompleksową obsługę bazy danych uwzględniającą wymienione powyżej filie. Założyć, że baza danych składa się z dwóch tabel połączonych ze sobą za pomocą relacji „jeden-do-wielu”. Uwzględnić w aplikacji wszystkie operacje należące do grupy CRUD (Create, Read, Update i Delete).

## UWAGI

1. Dane wejściowe:
  - a) zdefiniować w kodzie aplikacji (na ocenę 2);
  - b) pobrać z formularza (na oceny 3, 4, 5).

2. Jako wyjście aplikacji wykorzystać:
  - a) konsolę (*na ocenę 2*);
  - b) stronę WWW (*na oceny 3, 4, 5*).
3. Zaimplementować kompleksową obsługę błędów. Wykorzystać funkcje asynchroniczne `async` wraz z operatorem `await` lub łańcuch wywołań metody `then()` połączony z pojedynczym wywołaniem metody `catch()` (*na ocenę 2*), albo oba wymienione sposoby (*na oceny 3, 4, 5*).
4. Na ocenę 5 lub 6 wykonać dodatkowo zadanie \*.