

MySQL (MariaDB) - polecenia SQL

Tworzenie bazy danych

```
CREATE DATABASE `cars_rental` CHARACTER SET `utf8mb4` COLLATE `utf8mb4_unicode_ci`;
```

- `CHARACTER SET [kodowanie]` - ustawi kodowanie znaków (utf8mb4)
- `COLLATE [metoda porównywania]` - ustawi domyślną metodę porównywania napisów dla pól tekstowych (utf8mb4_unicode_ci lub utf8mb4_polish_ci)

Tworzenie tabeli

```
CREATE TABLE `manufacturer` (  
  `id` INT AUTO_INCREMENT PRIMARY KEY,  
  `name` VARCHAR(30) NOT NULL,  
  `status` ENUM('active','closed') NOT NULL DEFAULT 'active'  
);  
  
CREATE TABLE `class` (  
  `id` INT AUTO_INCREMENT PRIMARY KEY,  
  `name` VARCHAR(30) NOT NULL  
);
```

- typy pól:
 - `INT` - liczby całkowite
 - `VARCHAR(x)` - pole tekstowe o określonej długości x
 - `ENUM('opcja 1','opcja 2'[,...])` - pole listy, elementy na liście przechowywane są w postaci liczbowej
 - `FLOAT` - liczby zmiennoprzecinkowe
 - `YEAR` - rok
 - `DATE` - data
 - `TEXT` - pole pozwalające na przechowanie dużej ilości tekstu
 - `BOOLEAN` - prawda/fałsz
- `AUTO_INCREMENT` - przy dodawaniu wierszy w polu będzie wstawiana kolejna liczba całkowita, używane głównie przy `id`
- `PRIMARY KEY` - pole będzie kluczem głównym tabeli
- `NOT NULL` - pole nie może być puste
- `NULL` - pole może być puste
- `DEFAULT wartość` - domyślna wartość, która zostanie użyta jeśli użytkownik nie zadeklaruje wartości przy wstawianiu wiersza

Tworzenie tabeli z kluczem obcym + indexy

```
CREATE TABLE `cars` (  
  `id` INT AUTO_INCREMENT PRIMARY KEY,  
  `model` VARCHAR(30) NOT NULL,  
  `manufacturer_id` INT NOT NULL,  
  INDEX (`manufacturer_id`),  
  UNIQUE (`model`),  
  FOREIGN KEY (`manufacturer_id`) REFERENCES `manufacturer` (`id`)  
);
```

- INDEX ([pole tabeli]) - tworzy indeks w oparciu o pole tabeli
- UNIQUE ([pole tabeli]) - wymusza unikatowość wartości w polu
- FOREIGN KEY (pole_tabeli) REFERENCES nazwa_tabeli_x(pole_tabeli_x) - tworzy relację, pole_tabeli będzie przechowywać klucz obcy z tabeli nazwa_tabeli_x

Zmiana nazwy tabeli

```
ALTER TABLE `cars` RENAME TO `car`;
```

Modyfikacja istniejącej kolumny

```
ALTER TABLE `car` CHANGE `model` `model` VARCHAR(40) NOT NULL;  
# powtórzenie nazwy model, ponieważ polecenie zakłada, że możemy też chcieć zmienić nazwę
```

Dodawanie kolumny do tabeli

```
ALTER TABLE `car` ADD COLUMN `class_id` INT NOT NULL;  
# doda kolumnę na końcu  
ALTER TABLE `car` ADD COLUMN `safetyRating` INT NOT NULL AFTER `model`;  
# doda kolumnę safetyRating po kolumnie model  
ALTER TABLE `car` ADD COLUMN `maxSpeed` INT NOT NULL AFTER `safetyRating`;  
# doda kolumnę maxSpeed po kolumnie safetyRating
```

Dodawanie klucza obcego do tabeli

```
ALTER TABLE `car` ADD CONSTRAINT FOREIGN KEY (`class_id`) REFERENCES `class` (`id`);
```

Dodawanie danych do tabeli

```
INSERT INTO `class` VALUES (NULL, 'standard'), (NULL, 'luxury');
```

- jeśli nie podano pól po nazwie tabeli trzeba wprowadzić wartości dla wszystkich, w przypadku pola autonumerowanego jako wartość zawsze podajemy NULL

```
INSERT INTO `manufacturer` SET `name`='Daewoo', `status`='active';
INSERT INTO `manufacturer` SET `name`='Fiat', `status`=1;
```

- alternatywna metoda wprowadzania, niezbyt wygodna przy dużej ilości pól
- warto zauważyć, że dla pola status wartości mogą być wprowadzane jako tekst lub liczba

```
INSERT INTO
  `car`(`model`,`manufacturer_id`,`safetyRating`,`maxSpeed`,`class_id`)
VALUES
  ('Matiz',1,5,100,1),('Panda',2,5,100,2);
```

- wyszczególnione dla których pól zostaną podane wartości, pominięte pola zostaną, zależnie od ich ustawień, wypełnione wartościami domyślnymi, autonumerowaniem lub null

Modyfikowanie danych w tabeli

```
UPDATE `car` SET `model`='Lanos' WHERE `id`=1;
# zmień nazwę modelu na Lanos dla id równego 1
UPDATE `car` SET `class_id`=1 WHERE `model` LIKE '%Panda%';
# zmień id klasy na 1 dla samochodu w którego nazwie jest słowo Panda
```

Usuwanie danych z tabeli

```
DELETE FROM `car` WHERE `id`>5;
# usuń z tabeli wszystkie samochody o id większym niż 5
DELETE FROM `car` WHERE `id` IN (1,6,7,5);
# usuń z tabeli samochody o id równym 1,6,7 lub 5
DELETE FROM `car` WHERE `id` BETWEEN 10 AND 150
# usuń z tabeli samochody o id między 10 a 150
```

Usuwanie kolumny, tabeli, bazy

```
ALTER TABLE `car` DROP COLUMN `safetyRating`;
DROP TABLE `car`;
DROP DATABASE `cars_rental`;
```