

BAZY DANYCH

Wypożyczalnia Samochodów Sportowych

Etap 2 - Projekt, implementacja i testy bazy danych

Imię i Nazwisko:	Remigiusz Mielcarz, Mateusz Świątek
Nr indeksu:	252887, 252858

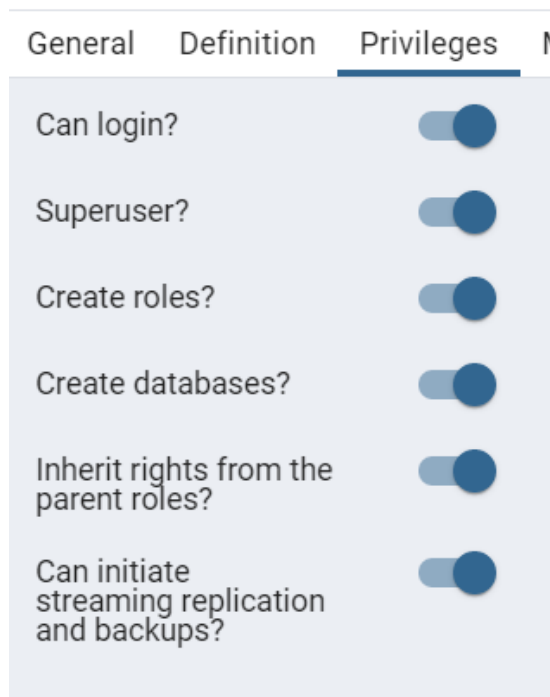
Termin zajęć: dzień tygodnia, godzina:	Piątek 15:15-16:55
Numer grupy ćwiczeniowej:	Y00-37m
Data wykonania ćwiczenia:	21.04.2022
Termin do oddania sprawozdania:	22.04.2022
Prowadzący kurs:	Dr inż. Paweł Głuchowski

Spis treści

1	Użytkownik bazy danych	2
2	Diagram Konceptualny	3
3	Diagram Fizyczny	3
4	Skrypty SQL	5
4.1	Tworzenie bazy danych	5
4.2	Tworzenie tabel i ich powiązań	5
4.3	Wypełnianie tabel danymi	6
4.4	Wyświetlanie zawartości	8
4.5	Filtracja	9
4.6	Wyzwalacze	9
5	Testy	11

1 Użytkownik bazy danych

Użytkownikiem bazy danych jest właściciel wypożyczalni samochodów. Powinien on posiadać wszystkie uprawnienia do zarządzania bazą danych. Naszym właścicielem jest użytkownik 'mateusz'. Ma on przyznane następujące przywileje:



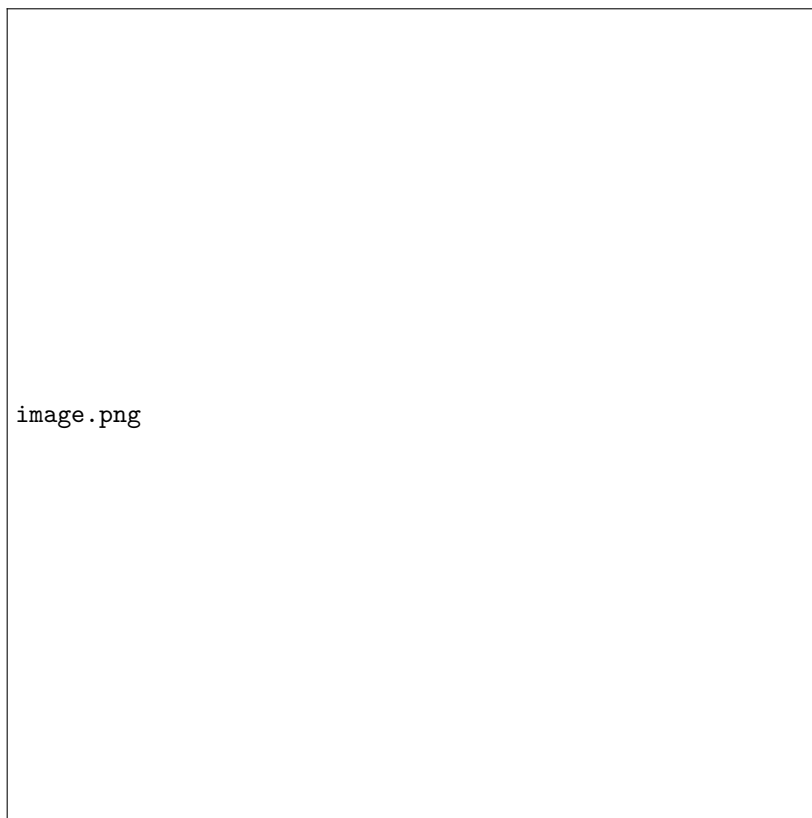
Rysunek 1: Przywileje użytkownika bazy danych

```
1 -- Role: mateusz
2 -- DROP ROLE IF EXISTS mateusz;
3
4 CREATE ROLE mateusz WITH
5     LOGIN
6     SUPERUSER
7     INHERIT
8     CREATEDB
9     CREATEROLE
10    REPLICATION
11    ENCRYPTED PASSWORD 'SCRAM-SHA-
```

Rysunek 2: Użytkownik mateusz

2 Diagram Konceptualny

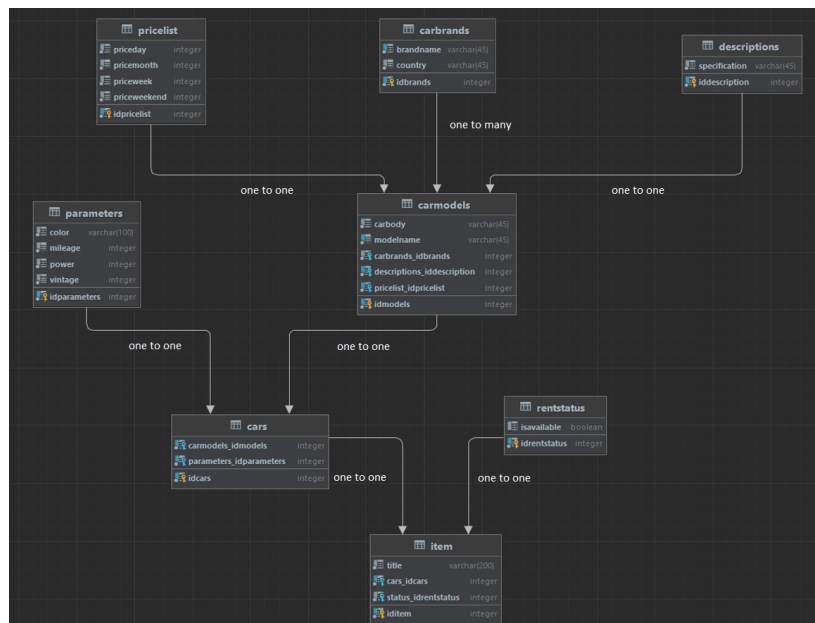
Na potrzeby planowania modelu bazy danych, należało w pierwszej kolejności stworzyć diagram konceptualny:



Rysunek 3: Diagram ERD konceptualny

3 Diagram Fizyczny

Po rozbudowaniu diagramu konceptualnego, otrzymaliśmy poniższy diagram fizyczny:



Rysunek 4: Diagram ERD fizyczny

4 Skrypty SQL

4.1 Tworzenie bazy danych

```
CREATE DATABASE baza1;
```

4.2 Tworzenie tabel i ich powiązań

```
CREATE TABLE CarBrands (idBrands SERIAL primary key ,
brandName varchar(50) NOT NULL UNIQUE,
country varchar(50) NOT NULL);

CREATE TABLE CarModels (idModels SERIAL primary key,
carBody varchar(50) NOT NULL,
modelName varchar(50) NOT NULL UNIQUE,
CarBrands_idBrands int NOT NULL,
Descriptions_idDescription int NOT NULL UNIQUE,
PriceList_idPriceList int NOT NULL UNIQUE);

CREATE TABLE Cars (idCars SERIAL primary key,
CarModels_idModels int NOT NULL UNIQUE,
Parameters_idParameters int NOT NULL UNIQUE);

CREATE TABLE Descriptions (idDescription SERIAL primary key,
specification varchar(50) NOT NULL);

CREATE TABLE Item (idItem SERIAL primary key,
title varchar(200) NOT NULL,
Cars_idCars int NOT NULL UNIQUE,
Status_idRentStatus int NOT NULL);

CREATE TABLE Parameters (idParameters SERIAL primary key,
color varchar(100) NOT NULL,
mileage int NOT NULL,
power int NOT NULL,
vintage int NOT NULL);

CREATE TABLE PriceList(idPriceList SERIAL primary key,
priceDay int NOT NULL,
priceMonth int NOT NULL,
priceWeek int NOT NULL,
priceWeekend int NOT NULL);

CREATE TABLE RentStatus(idRentStatus SERIAL primary key,
isAvailable boolean);
```

```

alter table CarModels add foreign key (CarBrands_idBrands) REFERENCES CarBrands (idBrands)
ON UPDATE CASCADE ON DELETE CASCADE;

alter table CarModels add foreign key (Descriptions_idDescription) REFERENCES Descriptions (idDescription)
ON UPDATE CASCADE ON DELETE CASCADE;

alter table CarModels add foreign key (PriceList_idPriceList) REFERENCES PriceList (idPriceList)
ON UPDATE CASCADE ON DELETE CASCADE;

alter table Cars add foreign key (CarModels_idModels) REFERENCES CarModels (idModels)
ON UPDATE CASCADE ON DELETE CASCADE;

alter table Cars add foreign key (Parameters_idParameters) REFERENCES Parameters (idParameters)
ON UPDATE CASCADE ON DELETE CASCADE;

alter table Item add foreign key (Cars_idCars) REFERENCES Cars (idCars)
ON UPDATE CASCADE ON DELETE CASCADE;

alter table Item add foreign key (Status_idRentStatus) REFERENCES RentStatus (idRentStatus)
ON UPDATE CASCADE ON DELETE CASCADE;

```

W powyższym fragmencie kodu zdefiniowano:

- Typy kolumn zgodnie z wymaganiami serwera
- Ustawienie odpowiednich kolumn jako unikalne
- Klucze główne i obce dla powiązań tabel

4.3 Wypełnianie tabel danymi

```

-- Adding rows to carbrands
INSERT INTO carbrands (brandname, country) VALUES ('Audi', 'Germany');
INSERT INTO carbrands (brandname, country) VALUES ('Lamborghini', 'Italy');
INSERT INTO carbrands (brandname, country) VALUES ('BMW', 'Germany');
INSERT INTO carbrands (brandname, country) VALUES ('Dodge', 'USA');
INSERT INTO carbrands (brandname, country) VALUES ('Nissan', 'Japan');

-- Adding rows to descriptions
INSERT INTO descriptions (specification) VALUES ('Czwarta generacja A3');
INSERT INTO descriptions (specification) VALUES ('Czwarta generacja A6');
INSERT INTO descriptions (specification) VALUES ('Szybki ekskluzywny samochód');
INSERT INTO descriptions (specification) VALUES ('Jeszcze szybszy ekskluzywny samochód');
INSERT INTO descriptions (specification) VALUES ('Najlepszy supersamochód');
INSERT INTO descriptions (specification) VALUES ('Drugi najlepszy supersamochód');
INSERT INTO descriptions (specification) VALUES ('Ciężki supersamochód');
INSERT INTO descriptions (specification) VALUES ('Jeszcze cięższy supersamochód');
INSERT INTO descriptions (specification) VALUES ('Inochę lżejszy supersamochód');
INSERT INTO descriptions (specification) VALUES ('Leciutki supersamochód');

-- Adding rows to pricelist
INSERT INTO pricelist (priceDay, priceWeekend, priceWeek, priceMonth) VALUES (599, 1500, 3000, 10500);
INSERT INTO pricelist (priceDay, priceWeekend, priceWeek, priceMonth) VALUES (2000, 7000, 14000, 49000);
INSERT INTO pricelist (priceDay, priceWeekend, priceWeek, priceMonth) VALUES (4000, 10000, 20000, 70000);
INSERT INTO pricelist (priceDay, priceWeekend, priceWeek, priceMonth) VALUES (5000, 12000, 24000, 80000);
INSERT INTO pricelist (priceDay, priceWeekend, priceWeek, priceMonth) VALUES (2300, 5750, 11500, 35000);
INSERT INTO pricelist (priceDay, priceWeekend, priceWeek, priceMonth) VALUES (2000, 4000, 8000, 24000);
INSERT INTO pricelist (priceDay, priceWeekend, priceWeek, priceMonth) VALUES (3000, 6000, 12000, 36000);
INSERT INTO pricelist (priceDay, priceWeekend, priceWeek, priceMonth) VALUES (2000, 4000, 8000, 24000);
INSERT INTO pricelist (priceDay, priceWeekend, priceWeek, priceMonth) VALUES (2000, 7000, 14000, 49000);
INSERT INTO pricelist (priceDay, priceWeekend, priceWeek, priceMonth) VALUES (2000, 7000, 14000, 49000);

```

```

-- Adding rows to carmodels
INSERT INTO carmodels (carBody, modelName, CarBrands_idBrands, Descriptions_idDescription, PriceList_idPriceList) VALUES ('Sedan', 'A3', 1, 1, 1);
INSERT INTO carmodels (carBody, modelName, CarBrands_idBrands, Descriptions_idDescription, PriceList_idPriceList) VALUES ('Sedan', 'A6', 1, 2, 2);
INSERT INTO carmodels (carBody, modelName, CarBrands_idBrands, Descriptions_idDescription, PriceList_idPriceList) VALUES ('Coupe', 'Aventador', 2, 3, 3);
INSERT INTO carmodels (carBody, modelName, CarBrands_idBrands, Descriptions_idDescription, PriceList_idPriceList) VALUES ('Roadster', 'Huracan', 2, 4, 4);
INSERT INTO carmodels (carBody, modelName, CarBrands_idBrands, Descriptions_idDescription, PriceList_idPriceList) VALUES ('Cabrio', 'E92M3', 3, 5, 5);
INSERT INTO carmodels (carBody, modelName, CarBrands_idBrands, Descriptions_idDescription, PriceList_idPriceList) VALUES ('Sedan', 'M5CS', 3, 6, 6);
INSERT INTO carmodels (carBody, modelName, CarBrands_idBrands, Descriptions_idDescription, PriceList_idPriceList) VALUES ('Sedan', 'Challenger', 4, 7, 7);
INSERT INTO carmodels (carBody, modelName, CarBrands_idBrands, Descriptions_idDescription, PriceList_idPriceList) VALUES ('Sedan', 'Charger', 4, 8, 8);
INSERT INTO carmodels (carBody, modelName, CarBrands_idBrands, Descriptions_idDescription, PriceList_idPriceList) VALUES ('Coupe', 'R35', 5, 9, 9);
INSERT INTO carmodels (carBody, modelName, CarBrands_idBrands, Descriptions_idDescription, PriceList_idPriceList) VALUES ('Coupe', 'Z', 5, 10, 10);

-- Adding rows to parameters
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('White', '120000', 120, 2016);
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('Black', '80000', 210, 2018);
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('Pink', '4700', 650, 2017);
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('Yellow', '200', 825, 2022);
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('Black', '120000', 90, 2012);
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('Black', '100', 600, 2022);
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('Red', '10000', 850, 2018);
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('White', '200000', 420, 1987);
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('Green', '12000', 300, 2020);
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('Blue', '10000', 350, 2021);

--Adding rows to cars
INSERT INTO cars(carmodels_idmodels, parameters_idparameters) VALUES (1,1);
INSERT INTO cars(carmodels_idmodels, parameters_idparameters) VALUES (2,2);
INSERT INTO cars(carmodels_idmodels, parameters_idparameters) VALUES (3,3);
INSERT INTO cars(carmodels_idmodels, parameters_idparameters) VALUES (4,4);
INSERT INTO cars(carmodels_idmodels, parameters_idparameters) VALUES (5,5);
INSERT INTO cars(carmodels_idmodels, parameters_idparameters) VALUES (6,6);
INSERT INTO cars(carmodels_idmodels, parameters_idparameters) VALUES (7,7);
INSERT INTO cars(carmodels_idmodels, parameters_idparameters) VALUES (8,8);
INSERT INTO cars(carmodels_idmodels, parameters_idparameters) VALUES (9,9);
INSERT INTO cars(carmodels_idmodels, parameters_idparameters) VALUES (10,10);

```

```

-- adding rows to rentstatus
INSERT INTO rentstatus (isavailable) VALUES (False);
INSERT INTO rentstatus (isavailable) VALUES (True);

-- Adding rows to item
INSERT INTO item(title, cars_idcars, status_idrentstatus) VALUES ('Piekne Audi A3 skus sie',1,2);
INSERT INTO item(title, cars_idcars, status_idrentstatus) VALUES ('Audi A6. Luksus i piekno',2,2);
INSERT INTO item(title, cars_idcars, status_idrentstatus) VALUES ('Pedz szybko',3,2);
INSERT INTO item(title, cars_idcars, status_idrentstatus) VALUES ('Pedz jak Huracan',4,1);
INSERT INTO item(title, cars_idcars, status_idrentstatus) VALUES ('Piekny bulgot V8',5,2);
INSERT INTO item(title, cars_idcars, status_idrentstatus) VALUES ('Szybka limuzyna',6,1);
INSERT INTO item(title, cars_idcars, status_idrentstatus) VALUES ('0d 0 do 100kmh w 3.4s',7,2);
INSERT INTO item(title, cars_idcars, status_idrentstatus) VALUES ('0d 0 do 100kmh w 3.7s',8,1);
INSERT INTO item(title, cars_idcars, status_idrentstatus) VALUES ('Tokyo Drift',9,2);
INSERT INTO item(title, cars_idcars, status_idrentstatus) VALUES ('Tokyo Style B)',10,2);

```

W powyższym kodzie wypełniamy utworzone wcześniej tabele danymi.

4.4 Wyświetlanie zawartości

```
-- full car info
SELECT brandname,modelname,priceday,priceweekend,priceweek,pricemonth,country,carbody,specification FROM carmodels
JOIN carbrands cb on carmodels.carbrands_idbrands = cb.idbrands
JOIN pricelist pl on carmodels.pricelist_idpricelist = pl.idpricelist
JOIN descriptions dsc on carmodels.descriptions_iddescription = dsc.iddescription;

-- price info
SELECT brandname,modelname,priceday,priceweekend,priceweek,pricemonth FROM carmodels
JOIN carbrands cb on carmodels.carbrands_idbrands = cb.idbrands
JOIN pricelist pl on carmodels.pricelist_idpricelist = pl.idpricelist;

-- car descriptions
SELECT brandname,modelname,specification FROM carmodels
JOIN carbrands cb on carmodels.carbrands_idbrands = cb.idbrands
JOIN descriptions dsc on carmodels.descriptions_iddescription = dsc.iddescription;

--horsepower
SELECT brandname,modelname,power FROM cars
JOIN carmodels cm on cars.carmodels_idmodels = cm.idmodels
JOIN carbrands cb on cb.idbrands = cm.carbrands_idbrands
JOIN parameters pm on cars.parameters_idparameters = pm.idparameters;

--cars availability
SELECT brandname,modelname,isavailable FROM item
JOIN rentstatus r on r.idrentstatus = item.status_idrentstatus
JOIN cars c on c.idcars = item.cars_idcars
JOIN carmodels cm on cm.idmodels = c.carmodels_idmodels
JOIN carbrands cb on cb.idbrands = cm.carbrands_idbrands;
```

Przy pomocy powyższego kodu, jesteśmy w stanie wyświetlić określone zestawy danych z naszej bazy. Niżej jeden z przykładowych zestawów danych:

	brandname	modelname	priceday	priceweekend	priceweek	pricemonth
1	Audi	A3	599	1500	3000	10500
2	Audi	A6	2000	7000	14000	49000
3	Lamborghini	Aventador	4000	10000	20000	70000
4	Lamborghini	Huracan	5000	12000	24000	80000
5	BMW	E92M3	2300	5750	11500	35000
6	BMW	M5CS	2000	4000	8000	24000
7	Dodge	Challenger	3000	6000	12000	36000
8	Dodge	Charger	2000	4000	8000	24000
9	Nissan	R35	2000	7000	14000	49000

Rysunek 5: Przykładowa tabela

4.5 Filtracja

Poniższe zapytania SQL możemy wykorzystać do wyświetlania określonych zestawów danych po filtracji o określonych kryteriach. Poniżej znajduje się tabela z samochodami dostępnymi do wynajęcia, która jest posortowana po mocy silnika.

```
-- filtered search, German and Italian cars between 2000-6000pln/day, order from cheapest to most expensive
SELECT brandname,modelname,priceday,priceweekend,priceweek,pricemonth,country,carbody,specification FROM carmodels
JOIN carbrands cb on carmodels.carbrands_idbrands = cb.idbrands
JOIN pricelist pl on carmodels.pricelist_idpricelist = pl.idpricelist
JOIN descriptions dsc on carmodels.descriptions_iddescription = dsc.iddescription
WHERE priceday BETWEEN 2000 AND 6000
AND country LIKE 'Germany' OR country LIKE 'Italy'
ORDER BY priceday;

--order available cars by horsepower
SELECT brandname,modelname,power,isavailable FROM item
JOIN rentstatus r on r.idrentstatus = item.status_idrentstatus
JOIN cars c on item.cars_idcars = c.idcars
JOIN carmodels cm on cm.idmodels = c.carmodels_idmodels
JOIN carbrands cb on cb.idbrands = cm.carbrands_idbrands
JOIN parameters pm on pm.idparameters = c.parameters_idparameters
WHERE isavailable IS TRUE
ORDER BY power;
```

	brandname	modelname	power	isavailable
1	BMW	E92M3	90	• true
2	Audi	A3	120	• true
3	Audi	A6	210	• true
4	Nissan	R35	300	• true
5	Lamborghini	Aventador	650	• true
6	Dodge	Challenger	850	• true

4.6 Wyzwalacze

Przy pomocy poniższych wyzwalaczy po usunięciu jednego elementu z tablicy Item, usunięte zostaną dane z pozostałych powiązanych tabel.

```

CREATE FUNCTION deleteItem() RETURNS TRIGGER AS $_$
BEGIN
    DELETE FROM cars WHERE cars.idcars = OLD.cars_idcars;
    RETURN OLD;
END $_$ LANGUAGE 'plpgsql';

CREATE TRIGGER deleteItem
AFTER DELETE ON item
FOR EACH ROW
EXECUTE PROCEDURE deleteItem();
--
CREATE FUNCTION deleteParameters() RETURNS TRIGGER AS $_$
BEGIN
    DELETE FROM parameters WHERE parameters.idparameters = OLD.parameters_idparameters;
    RETURN OLD;
END $_$ LANGUAGE 'plpgsql';

CREATE TRIGGER deleteparameters
AFTER DELETE ON cars
FOR EACH ROW
EXECUTE PROCEDURE deleteParameters();
--
CREATE FUNCTION deleteCarModels() RETURNS TRIGGER AS $_$
BEGIN
    DELETE FROM carmodels WHERE carmodels.idmodels = OLD.carmodels_idmodels;
    RETURN OLD;
END $_$ LANGUAGE 'plpgsql';

```

```

CREATE TRIGGER deletemodels
AFTER DELETE ON cars
FOR EACH ROW
EXECUTE PROCEDURE deleteCarModels();
--
CREATE FUNCTION deletePriceList() RETURNS TRIGGER AS $_$
BEGIN
    DELETE FROM pricelist WHERE pricelist.idpricelist = OLD.pricelist_idpricelist;
    RETURN OLD;
END $_$ LANGUAGE 'plpgsql';

CREATE TRIGGER deletepricelist
AFTER DELETE ON carmodels
FOR EACH ROW
EXECUTE PROCEDURE deletePriceList();
--
CREATE FUNCTION deleteDescriptions() RETURNS TRIGGER AS $_$
BEGIN
    DELETE FROM descriptions WHERE descriptions.iddescription = OLD.descriptions_iddescription;
    RETURN OLD;
END $_$ LANGUAGE 'plpgsql';

CREATE TRIGGER deletedescriptions
AFTER DELETE ON carmodels
FOR EACH ROW
EXECUTE PROCEDURE deleteDescriptions();
--
DELETE FROM item WHERE item.iditem = 10;

```

Wykonano w systemie L^AT_EX

5 Testy

W celu sprawdzenia poprawności implementacji tablic baz danych, przeprowadziliśmy szereg testów na wprowadzaniu do nich danych:

- Podanie zbyt małej ilości argumentów

```
-- Adding rows to parameters
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('White', '120000', 120);
```

```
mysql> INSERT INTO parameters (color, mileage, power, vintage) VALUES ('White', '120000', 120)
[2022-04-28 23:26:24] [42601] ERROR: INSERT has more target columns than expressions
[2022-04-28 23:26:24] Pozycja: 48
```

- Podanie zbyt dużej ilości argumentów

```
-- Adding rows to parameters
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('White', '120000', 120, 2019, "Za duzo argumentow");
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('White', '120000', 120, 2019, "Za duzo argumentow");
```

```
mysql> INSERT INTO parameters (color, mileage, power, vintage) VALUES ('White', '120000', 120, 2019, "Za duzo argumentow")
[2022-04-28 23:27:39] [42703] ERROR: column "Za duzo argumentow" does not exist
[2022-04-28 23:27:39] Pozycja: 93
```

- Podanie złego rodzaju danych do jednej z kolumn

```
⚠ Adding rows to parameters
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('White', 'Zly rodzaj danych', 120, 2019);
```

```
mysql> INSERT INTO parameters (color, mileage, power, vintage) VALUES ('White', 'Zly rodzaj danych', 120, 2019)
[2022-04-28 23:31:36] [22P02] ERROR: invalid input syntax for type integer: "Zly rodzaj danych"
[2022-04-28 23:31:36] Pozycja: 74
```

- Podanie nieistniejącego klucza

```
INSERT INTO item(title, cars_idcars, status_idrentstatus) VALUES ('0d 0 do 100kmh w 3.7s', 14, 1);
```

```
mysql> INSERT INTO item(title, cars_idcars, status_idrentstatus) VALUES ('0d 0 do 100kmh w 3.7s', 14, 1)
[2022-04-28 23:41:13] [23503] ERROR: insert or update on table "item" violates foreign key constraint "item_cars_idcars_fkey"
[2022-04-28 23:41:13] Szczegóły: Key (cars_idcars)=(14) is not present in table "cars".
```

- Podanie poprawnych danych do INSERT INTO Statement

```
-- Adding rows to parameters
INSERT INTO parameters (color, mileage, power, vintage) VALUES ('White', '120000', 120, 2019);
```

```
mysql> INSERT INTO parameters (color, mileage, power, vintage) VALUES ('White', '120000', 120, 2019)
[2022-04-28 23:32:29] 1 row affected in 7 ms
```