

Sprawozdanie Bazy Danych i Systemy Informacyjne

Lista nr 4


Artur Trzeźniewski








Kamil Walter

Cel projektu





Celem projektu jest stworzenie aplikacji webowej obsługującej bazę danych, pomagającej w zarządzaniu siecią salonów samochodowych **SpeedyGad**. Aplikacja będzie oparta o relacyjną bazę danych z użyciem języka SQL (dialekt MySQL).

Tabele

 **Users** - dane użytkowników aplikacji, pracowników salonu samochodowego (przechowywane hasła zostały zahashowane używając bcrypt.js)

Users	
 ID	int unsigned
 login	varchar(50)
 password	varchar(100)
 type	enum('worker', 'manager', 'admin')
 name	varchar(50)
 lastname	varchar(50)
 gender	enum('m', 'k')

 **Stores** - salony samochodowe **SpeedyGad**


Stores	
 ID	int unsigned
 city	varchar(50)
 street	varchar(50)
 number	varchar(4)
 zip_code	char(5)
 phone_number	int unsigned





⚙️ **Brands** - Marki samochodów sprzedawanych w salonach

📊 Brands	
🔑 ID	int unsigned
📄 name	varchar(50)
📄 country	varchar(50)






🏎️ **Models** - szczegółowe informacje o samochodach

📊 Models	
🔑 ID	int unsigned
🔑 brandID	int unsigned
📄 name	varchar(50)
📄 price	float unsigned
📄 max_speed	decimal(5,2) unsigned

 **Cars in Stores** - ilość samochodów danego koloru w konkretnym sklepie, gdzie primary key to wektor (modelID, storeID, color)

Cars_in_stores	
 modelID	int unsigned
 storeID	int unsigned
 color	varchar(20)
 quantity	int unsigned

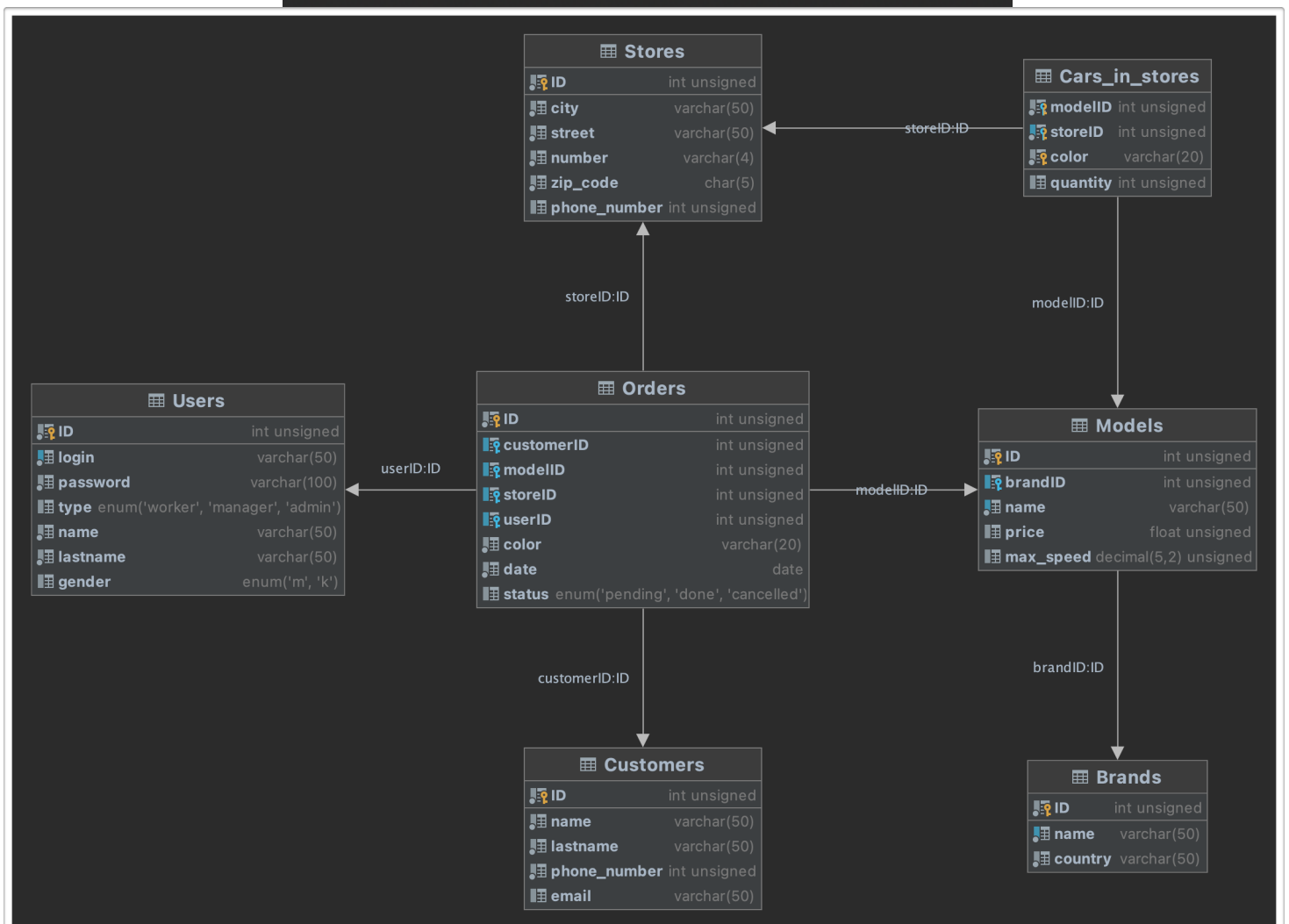
 **Customers** - dane osobowe Klientów

Customers	
 ID	int unsigned
 name	varchar(50)
 lastname	varchar(50)
 phone_number	int unsigned
 email	varchar(50)



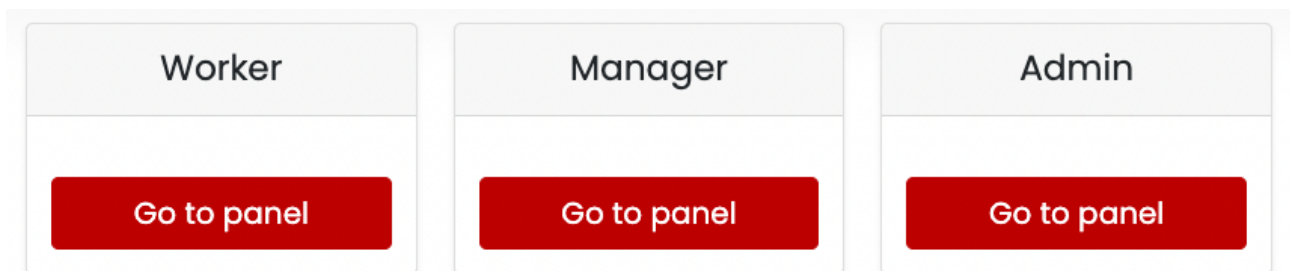
Orders - dane o zamówieniach: dla jakiego klienta, model auta w danym kolorze, z jakiego salonu, przez jakiego pracownika, data, na którą planowane jest dostarczenie zamówienia, oraz jego status

Orders	
ID	int unsigned
customerID	int unsigned
modelID	int unsigned
storeID	int unsigned
userID	int unsigned
color	varchar(20)
date	date
status	enum('pending', 'done', 'cancelled')




Użytkownicy

Korzystanie z aplikacji dostępne jest dla trzech rodzajów użytkowników: pracownika, kierownika i administratora.



Sama baza danych posiada czterech użytkowników. App - połączenie tylko z pozwoleniem na SELECT loginu i zaszyfrowanego hasła w celu logowania użytkownika. Każda grupa (worker, manager, admin) posiada własne połączenie z bazą danych wraz z dostosowanymi do potrzeb uprawnieniami. Poza tym grupom nadane są uprawnienia niższych grup tzn. Manager posiada również uprawnienia Workera, a Admin obu powyższych.

Wszystkie funkcjonalności aplikacji (z pominięciem tych trywialnych) dotyczące bazy danych wykonywane są poprzez wywołanie procedur w **prepared statements**.

 **Pracownik** - może dodawać nowe zamówienia na samochody poprzez formularz oraz wyświetlać listy zawierające następująco: klientów, wszystkie sprzedawane marki, modele, sklepy, oraz *widoki* prezentujące:

1. samochody dostępne w podanych salonach (połączenie Brands, Models, Stores i Cars in Stores)
2. Zamówienia złożone przez klientów (połączenie Orders, Customers, Models i Brands)

<div>Add new Order</div> <div>Go to Order form</div>		
<div>Show list of Orders</div> <div>Go to the List</div>	<div>Show list of Clients</div> <div>Go to the List</div>	<div>Show list of available Cars</div> <div>Go to the List</div>
<div>Show list of All Models</div> <div>Go to the List</div>	<div>Show list of All Brands</div> <div>Go to the List</div>	<div>Show list of All Stores</div> <div>Go to the List</div>

 **Kierownik** - może dodawać poprzez formularze:

☒ Dodać Nowe Marki samochodowe

Add new Car Brand

[Go to Brand Form](#)

☒ Dodać Nowe Modele samochodów

Add new Car Model

[Go to Car Model Form](#)

☒ Dodać Auta do swoich Salonów

Add Car Models to Your Store

[Go to Car-to-Store Form](#)

☒ Anulować niezrealizowane zamówienia

Show list of Orders (Cancel)

[Go to the List](#)

🏆 **Administrator** - ma dostęp do większości bazy danych

☒ Dodać Nowe Salony sieci
SpeedyGad

Add new Shop

Go to Shop Form

☒ Tworzyć nowych
użytkowników Aplikacji

Add new User

Go to User Form

☒ Wyświetlać listę
użytkowników

Show All Users

Show Users List

Procedury

- **add_brand** - jeśli marka o podanej nazwie i kraju pochodzenia nie widnieje w tabeli, zostaje dodana do Brands. **Manager+**
- **add_car_model** - jeśli model auta o podanej nazwie, marce, maksymalnej prędkości nie widnieje w tabeli, zostaje dodany do Models. (Wywołuje wewnętrznie **add_brand**). **Manager+**
- **add_store** - jeśli salon samochodowy o podanych danych adresowych nie widnieje na liście, zostaje dodany do tabeli Stores. **Admin**
- **add_car_to_store** - jeśli model auta (w konkretnym kolorze) nie jest dodany do podanego salonu to zostaje dodany do Cars in Stores w podanej ilości w przeciwnym przypadku tylko zwiększamy liczbę sztuk dla tego salonu. (Wywołuje wewnętrznie **add_car_model**). **Manager+**

- **add_customer** - jeśli klient o podanych danych osobowych nie widnieje w tabeli Customers, zostaje do niej dodany. **Worker+**
- **add_order** - jeśli auto w podanym kolorze znajduje się w danym sklepie, zostaje złożone zamówienie dla podane klienta (Wywołuje wewnętrznie **add_customer**). **Worker+**
- **cancel_order** - jeśli status podanego zamówienia to „*pending*”, zostaje on zmieniony na „*cancelled*” a samochód, który był zamówiony zostaje ponownie dodany do tego sklepu. (Wywołuje wewnętrznie **add_car_to_store** z argumentem ilość=1). **Manager+**
- **add_user** - jeśli użytkownik o podanym loginie nie widnieje w tabeli Customers, zostaje do niej dodany. **Admin**
- **sample_data_generator** - dodaje do wszystkich tabel przykładowe dane. **App**

Events



remove_cars_from_stores_if_0_left - Usunięcie samochodu z salonu, gdy jego ilość wyniesie 0. (uruchamiane co godzinę)



update_orders_status - Zmiana statusu zamówień z „*pending*” na „*done*”, jeśli minie data dostawy, porównywanie z `current_date` (Uruchamiane każdego dnia)

Triggers



remove_model_if_no_cars_left - Usunięcie modelu, jeśli nie występuje w żadnym ze sklepów.



remove_brand_if_no_cars_left - Usunięcie Marek samochodowych, jeśli nie występuje w żadnym ze sklepów.

Normalizacja bazy danych

-1NF:

W każdej kolumnie każdej tabeli mamy dokładnie jedną wartość. Dodatkowo w bazie nie ma powtórzeń, więc jest ona pierwszej postaci normalnej.

-2NF:

Tabela Brands jest 2NF, bo {ID} jest kluczem minimalnym (bo zbiór pusty nie może być kluczem), więc pozostałe atrybuty są od niego w pełni zależne.

Tabela Cars_in_stores jest 2NF, bo atrybut quantity jest zależna od klucza {modelID, storeID, color} i dodatkowo nie istnieje mniejszy pod-klucz, bo np. dla {modelID, storeID} możemy mieć kilka modeli w różnych kolorach.

Tabela Customers jest 2NF, bo {ID} jest kluczem minimalnym.

Tabela Models jest 2NF, bo {ID} jest kluczem minimalnym.

Tabela Orders jest 2NF, bo {ID} jest kluczem minimalnym.

Tabela Stores jest 2NF, bo {ID, city, street, number} jest kluczem i nie istnieje mniejszy pod-klucz, bo dla np. {ID, city, street} nie możemy wywnioskować zip-code (kod pocztowy na jednym końcu ulicy może być inny niż na drugim).

Tabela Users jest 2NF, bo {ID} jest kluczem minimalnym.

-3NF:

Tabela Cars_in_stores jest 3NF, bo ma tylko jeden atrybut niekluczowy.

Tabela Customers jest 3NF, bo z żadnej kombinacji atrybutów name, lastname, phone_number i email nie jesteśmy w stanie wywnioskować pozostałych.

Tabela Models jest 3NF, bo z żadnej kombinacji atrybutów brandID, name, price i max_speed nie jesteśmy w stanie wywnioskować pozostałych.

Tabela Orders jest **3NF**, bo z żadnej kombinacji atrybutów customerID, modelID, storeID, userID, color, date i status nie jesteśmy w stanie wywnioskować pozostałych.

Tabela Stores jest **3NF**, bo z żadnego z atrybutów zip_code i phone_number nie jesteśmy w stanie wywnioskować drugiego.

Tabela Users jest **3NF**, bo z żadnej kombinacji atrybutów login, password, type, name, last_name i gender nie jesteśmy w stanie wywnioskować pozostałych. (można się zastanawiać czy name -> gender, ale w dzisiejszych czasach nie wiadomo)