

CarRentalDB



Miraslau Alkhovik 248655
Praskouya Horbach 248656

1 . Podstawowe założenia projektu

Cel projektu: stworzenie systemu zarządzania wypożyczalnią samochodów, który umożliwi sprawne zarządzanie pojazdami, pracownikami, klientami, rezerwacjami, płatnościami oraz historią działań. System ma na celu uproszczenie i zautomatyzowanie procesu wypożyczania pojazdów.

Główne założenia:

1. **Relacje między tabelami:** Tabele są powiązane ze sobą za pomocą kluczy obcych, co zapewnia integralność danych i umożliwia szybkie wyszukiwanie oraz modyfikowanie powiązanych rekordów.
2. **Bezpieczeństwo:** Projekt uwzględnia autoryzację użytkowników oraz role i uprawnienia (Administrator, Manager, Employee), co zapewnia kontrolę dostępu do danych oraz funkcji systemu.
3. **Rezerwacje i płatności:** System umożliwia klientom dokonywanie rezerwacji pojazdów, dokonywanie płatności i śledzenie statusu rezerwacji.
4. **Zarządzanie pojazdami:** Użytkownicy mają możliwość dodawania nowych pojazdów, zarządzania ich dostępnością, typami oraz ubezpieczeniami.

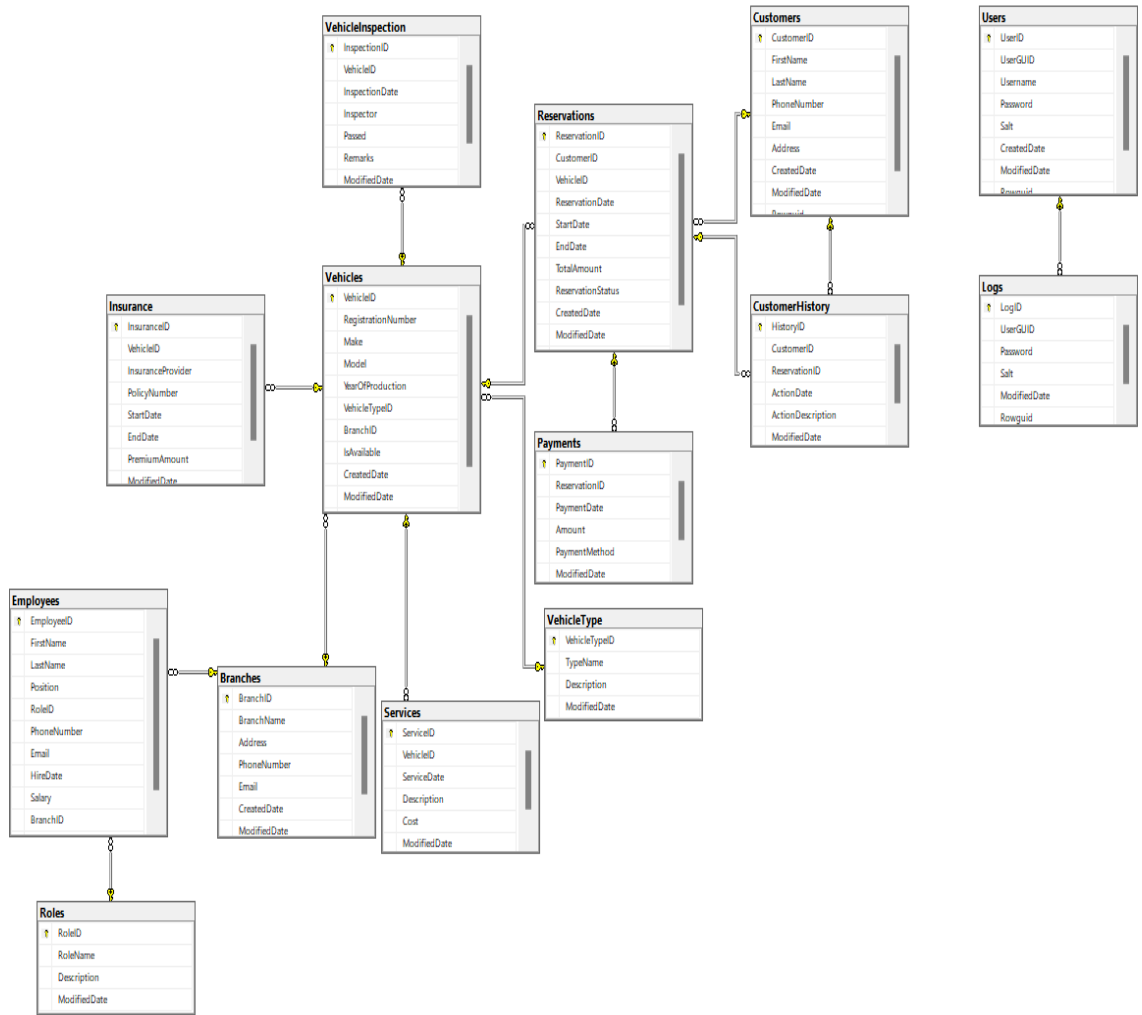
Możliwości:

1. **Zarządzanie oddziałami i pracownikami:** Pracownicy mogą być przypisani do różnych oddziałów, a ich dane, role oraz płace są przechowywane w systemie.
2. **Wyszukiwanie pojazdów:** System umożliwia łatwe wyszukiwanie dostępnych pojazdów według różnych kryteriów, takich jak typ pojazdu, marka, model, czy dostępność.
3. **Rejestracja historii:** System przechowuje historię działań klientów oraz transakcji płatniczych, co umożliwia śledzenie wszystkich operacji w systemie.
4. **Monitoring ubezpieczeń i serwisów:** Użytkownicy mogą sprawdzać status ubezpieczeń i przeglądów technicznych pojazdów, zapewniając ich bezpieczeństwo i aktualność.

Ograniczenia przyjęte przy projektowaniu:

1. **Wydajność:** System zakłada niewielką liczbę równocześnie pracujących użytkowników, dlatego nie uwzględnia zaawansowanej optymalizacji pod kątem obsługi dużych zbiorów danych.
2. **Brak integracji z zewnętrznymi systemami:** System nie zakłada integracji z zewnętrznymi serwisami, takimi jak systemy płatności, co może ograniczać jego elastyczność w kontekście rozwoju.
3. **Brak mobilnej wersji:** Projekt nie obejmuje aplikacji mobilnej ani interfejsu dostosowanego do urządzeń mobilnych, co ogranicza dostępność systemu na różnych platformach.

2 . Schemat bazy danych



3 . Obiekty

Branches

Przechowuje informacje o oddziałach wypożyczalni samochodów, takie jak nazwa oddziału, adres, numer telefonu oraz e-mail. Zawiera daty utworzenia i ostatniej modyfikacji.

- Pola:
 - `branch_id` (INT, PRIMARY KEY, AUTO_INCREMENT)
 - `name` (VARCHAR(100), NOT NULL)
 - `address` (VARCHAR(255), NOT NULL)
 - `phone` (VARCHAR(20), NOT NULL)
 - `email` (VARCHAR(100), UNIQUE, NOT NULL)
 - `created_at` (DATETIME, DEFAULT CURRENT_TIMESTAMP)
 - `updated_at` (DATETIME, DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP)
 - Relacje:
 - Powiązana z `Employees` (1 do wielu)
 - Powiązana z `Vehicles` (1 do wielu)
-

Roles

Przechowuje role użytkowników systemu, takie jak nazwa roli i jej opis. Używane w relacji z tabelą `Employees`.

- Pola:
 - `role_id` (INT, PRIMARY KEY, AUTO_INCREMENT)
 - `name` (VARCHAR(50), UNIQUE, NOT NULL)

- `description` (TEXT, NULL)
 - **Relacje:**
 - Powiązana z `Employees` (1 do wielu)
-

Employees

Przechowuje dane o pracownikach, w tym imię, nazwisko, stanowisko, rolę, numer telefonu, e-mail, datę zatrudnienia, wynagrodzenie oraz przypisany oddział. Jest powiązana z tabelami `Branches` i `Roles`.

- **Pola:**
 - `employee_id` (INT, PRIMARY KEY, AUTO_INCREMENT)
 - `first_name` (VARCHAR(100), NOT NULL)
 - `last_name` (VARCHAR(100), NOT NULL)
 - `position` (VARCHAR(50), NOT NULL)
 - `role_id` (INT, FOREIGN KEY do `Roles(role_id)`, NOT NULL)
 - `phone` (VARCHAR(20), NOT NULL)
 - `email` (VARCHAR(100), UNIQUE, NOT NULL)
 - `hire_date` (DATE, NOT NULL)
 - `salary` (DECIMAL(10,2), CHECK(salary >= 0))
 - `branch_id` (INT, FOREIGN KEY do `Branches(branch_id)`, NOT NULL)
 - **Relacje:**
 - Powiązana z `Roles` (wiele do 1)
 - Powiązana z `Branches` (wiele do 1)
-

VehicleType

Przechowuje typy pojazdów, takie jak sedan, SUV czy kombi. Zawiera unikalną nazwę typu oraz opcjonalny opis.

- **Pola:**
 - `vehicle_type_id` (INT, PRIMARY KEY, AUTO_INCREMENT)

- **name** (VARCHAR(50), UNIQUE, NOT NULL)
 - **description** (TEXT, NULL)
 - **Relacje:**
 - Powiązana z **Vehicles** (1 do wielu)
-

Vehicles

Przechowuje informacje o pojazdach dostępnych w wypożyczalni, takie jak numer rejestracyjny, marka, model, rok produkcji, typ pojazdu, oddział, a także dostępność. Powiązana z tabelami **VehicleType** i **Branches**.

- **Pola:**
 - **vehicle_id** (INT, PRIMARY KEY, AUTO_INCREMENT)
 - **registration_number** (VARCHAR(20), UNIQUE, NOT NULL)
 - **brand** (VARCHAR(50), NOT NULL)
 - **model** (VARCHAR(50), NOT NULL)
 - **year** (YEAR, CHECK(year >= 1900))
 - **vehicle_type_id** (INT, FOREIGN KEY do **VehicleType(vehicle_type_id)**, NOT NULL)
 - **branch_id** (INT, FOREIGN KEY do **Branches(branch_id)**, NOT NULL)
 - **is_available** (BOOLEAN, DEFAULT TRUE, NOT NULL)
 - **Relacje:**
 - Powiązana z **VehicleType** (wiele do 1)
 - Powiązana z **Branches** (wiele do 1)
 - Powiązana z **Insurance**, **Services**, **VehicleInspection** i **Reservations** (1 do wielu)
-

Insurance

Przechowuje informacje o ubezpieczeniach pojazdów, takie jak dostawca ubezpieczenia, numer polisy, daty ważności oraz wysokość składki. Powiązana z tabelą **Vehicles**.

- **Pola:**

- `insurance_id` (INT, PRIMARY KEY, AUTO_INCREMENT)
 - `provider` (VARCHAR(100), NOT NULL)
 - `policy_number` (VARCHAR(50), UNIQUE, NOT NULL)
 - `start_date` (DATE, NOT NULL)
 - `end_date` (DATE, NOT NULL)
 - `premium` (DECIMAL(10,2), CHECK(premium >= 0))
 - `vehicle_id` (INT, FOREIGN KEY do `Vehicles(vehicle_id)`, NOT NULL)
 - **Relacje:**
 - Powiązana z `Vehicles` (wiele do 1)
-

Services

Przechowuje informacje o serwisach pojazdów, w tym datę serwisu, opis usługi oraz koszt. Powiązana z tabelą `Vehicles`.

- **Pola:**
 - `service_id` (INT, PRIMARY KEY, AUTO_INCREMENT)
 - `service_date` (DATE, NOT NULL)
 - `description` (TEXT, NOT NULL)
 - `cost` (DECIMAL(10,2), CHECK(cost >= 0))
 - `vehicle_id` (INT, FOREIGN KEY do `Vehicles(vehicle_id)`, NOT NULL)
 - **Relacje:**
 - Powiązana z `Vehicles` (wiele do 1)
-

VehicleInspection

Przechowuje dane o przeglądach technicznych pojazdów, takie jak data przeglądu, inspektor, wynik (zaliczony/niezaliczony) oraz ewentualne uwagi. Powiązana z tabelą `Vehicles`.

- **Pola:**

- `inspection_id` (INT, PRIMARY KEY, AUTO_INCREMENT)
 - `inspection_date` (DATE, NOT NULL)
 - `inspector` (VARCHAR(100), NOT NULL)
 - `result` (ENUM('PASSED', 'FAILED'), NOT NULL)
 - `notes` (TEXT, NULL)
 - `vehicle_id` (INT, FOREIGN KEY do `Vehicles(vehicle_id)`, NOT NULL)
 - **Relacje:**
 - Powiązana z `Vehicles` (wiele do 1)
-

Customers

Przechowuje informacje o klientach, takie jak imię, nazwisko, numer telefonu, e-mail, adres oraz daty utworzenia i ostatniej modyfikacji.

- **Pola:**
 - `customer_id` (INT, PRIMARY KEY, AUTO_INCREMENT)
 - `first_name` (VARCHAR(100), NOT NULL)
 - `last_name` (VARCHAR(100), NOT NULL)
 - `phone` (VARCHAR(20), NOT NULL)
 - `email` (VARCHAR(100), UNIQUE, NOT NULL)
 - `address` (VARCHAR(255), NOT NULL)
 - `created_at` (DATETIME, DEFAULT CURRENT_TIMESTAMP)
 - `updated_at` (DATETIME, DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP)
 - **Relacje:**
 - Powiązana z `Reservations` i `CustomerHistory` (1 do wielu)
-

Reservations

Przechowuje informacje o rezerwacjach pojazdów, w tym klienta, pojazd, daty rezerwacji, daty wynajmu, kwotę całkowitą oraz status rezerwacji. Powiązana z tabelami `Customers` i `Vehicles`.

- **Pola:**

- `reservation_id` (INT, PRIMARY KEY, AUTO_INCREMENT)
- `customer_id` (INT, FOREIGN KEY do `Customers(customer_id)`, NOT NULL)
- `vehicle_id` (INT, FOREIGN KEY do `Vehicles(vehicle_id)`, NOT NULL)
- `reservation_date` (DATETIME, DEFAULT CURRENT_TIMESTAMP, NOT NULL)
- `rental_start_date` (DATE, NOT NULL)
- `rental_end_date` (DATE, NOT NULL)
- `total_amount` (DECIMAL(10,2), CHECK(total_amount >= 0), NOT NULL)
- `status` (ENUM('PENDING', 'CONFIRMED', 'CANCELLED'), DEFAULT 'PENDING', NOT NULL)

- **Relacje:**

- Powiązana z `Customers` i `Vehicles` (wiele do 1)
- Powiązana z `Payments` i `CustomerHistory` (1 do wielu)

Payments

Przechowuje dane o płatnościach za rezerwacje, takie jak data płatności, kwota oraz metoda płatności. Powiązana z tabelą `Reservations`.

- **Pola:**

- `payment_id` (INT, PRIMARY KEY, AUTO_INCREMENT)
- `reservation_id` (INT, FOREIGN KEY do `Reservations(reservation_id)`, NOT NULL)
- `payment_date` (DATETIME, DEFAULT CURRENT_TIMESTAMP, NOT NULL)
- `amount` (DECIMAL(10,2), CHECK(amount >= 0), NOT NULL)
- `payment_method` (ENUM('CASH', 'CREDIT_CARD', 'BANK_TRANSFER'), NOT NULL)

- **Relacje:**

- Powiązana z `Reservations` (wiele do 1)

CustomerHistory

Przechowuje historię działań klientów, takich jak rezerwacje, z datą i opisem działania. Powiązana z tabelami **Customers** i **Reservations**.

- **Pola:**
 - `history_id` (INT, PRIMARY KEY, AUTO_INCREMENT)
 - `customer_id` (INT, FOREIGN KEY do `Customers(customer_id)`, NOT NULL)
 - `reservation_id` (INT, FOREIGN KEY do `Reservations(reservation_id)`, NOT NULL)
 - `action_date` (DATETIME, DEFAULT CURRENT_TIMESTAMP, NOT NULL)
 - `description` (TEXT, NOT NULL)
- **Relacje:**
 - Powiązana z `Customers` i `Reservations` (wiele do 1)

Users

Przechowuje dane użytkowników systemu, takie jak unikalny identyfikator GUID, nazwa użytkownika, hasło, sól oraz daty utworzenia i modyfikacji.

- **Pola:**
 - `user_id` (INT, PRIMARY KEY, AUTO_INCREMENT)
 - `guid` (CHAR(36), UNIQUE, NOT NULL)
 - `username` (VARCHAR(50), UNIQUE, NOT NULL)
 - `password_hash` (CHAR(64), NOT NULL)
 - `password_salt` (CHAR(64), NOT NULL)
 - `created_at` (DATETIME, DEFAULT CURRENT_TIMESTAMP)
 - `updated_at` (DATETIME, DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP)
- **Relacje:**

- Powiązana z **Logs** (1 do wielu)
-

Logs

Przechowuje logi aktywności użytkowników systemu, zawierając GUID użytkownika, hasło i sól. Powiązana z tabelą **Users**.

- **Pola:**

- **log_id** (INT, PRIMARY KEY, AUTO_INCREMENT)
- **user_id** (INT, FOREIGN KEY do **Users(user_id)**, NOT NULL)
- **action** (TEXT, NOT NULL)
- **timestamp** (DATETIME, DEFAULT CURRENT_TIMESTAMP, NOT NULL)

- **Relacje:**

- Powiązana z **Users** (wiele do 1)

4 . Role i uprawnienia

Administrator: Pełna kontrola nad tabelą **Branches**.

Employee: Zarządzanie rezerwacjami w tabeli **Reservations** (SELECT, INSERT, UPDATE) i podgląd danych o pojazdach w tabeli **Vehicles** (SELECT).

Manager: Pełna kontrola nad tabelą **Customers**, umożliwiającą kompleksowe zarządzanie danymi klientów.

5 . Uwagi

Żadnych uwag, wszystko zrobione pomyślnie