



**POLITECHNIKA LUBELSKA  
WYDZIAŁ ELEKTROTECHNIKI  
I INFORMATYKI**

**KIERUNEK STUDIÓW  
INFORMATYKA**

Przedmiot: Wprowadzenie do systemów baz danych

*Raport z wykonania projektu pt.*

**Carsharing**

Autorzy:  
*Vladyslav Liulka  
Mykola Lutsyk  
Sofia Laba  
Nazar Malizderskyi*

Lublin, 2023



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz Społeczny



## **ROZDZIAŁ 1. OPIS WYBRANEGO OBSZARU RZECZYWISTOŚCI ORAZ WSKAZANIE PROBLEMU, KTÓRY ZOSTANIE ROZWIĄZANY PRZY POMOCY SYSTEMU INFORMATYCZNEGO WYKORZYSTUJĄCEGO PROJEKTOWANĄ BAZĘ DANYCH**

**Dokonaj ogólnej prezentacji wybranej rzeczywistości, wskazując procesy w niej realizowane oraz obiekty uczestniczące w tych procesach. Określ logiczną kolejność realizacji tych procesów.**

**Określ problem lub problemy jakie należy rozwiązać w wybranym obszarze rzeczywistości poprzez wdrożenie systemu informatycznego, w którym zastosujesz projektowaną bazę danych.**

### **1.1 Ogólny opis rzeczywistości:**

Carsharing jest rodzajem krótkoterminowego wynajmu samochodów osobowych, który opiera się na współdzieleniu floty przez użytkowników. Mówiąc o naszej rzeczywistości, carsharing początkowo będzie działał całodobowo w Polsce z możliwością skalowania w całej UE. Właściciel firmy decyduje, gdzie i jak chce się rozwijać. Ma on również możliwość zarządzania firmą. Dla klientów firma oferuje stronę internetową z informacją o firmie oraz instrukcję korzystania z usługi carsharingu przy pomocy aplikacji mobilnej. Także istnieje strona internetowa dla zarządzania firmą oraz zapewnienia działalności usługi carsharingu. Utrzymaniem systemu informatycznego oraz jego modyfikacją zajmuje się IT firma.

Aby rozpocząć korzystanie z car sharingu, klient musi zarejestrować się w systemie za pomocą aplikacji mobilnej. Aby zarejestrować się, należy mieć ukończone 21 lat i posiadać prawo jazdy od co najmniej 2 lat. Podczas rejestracji użytkownik podaje swoje dane (patrz 1.2 proces 2). Dla korzystania z usług konto musi być potwierdzone. Potwierdzeniem i komunikacją z klientami zajmują się pracownicy, z którymi klient może skontaktować się w dowolnym momencie za pośrednictwem numeru telefonu lub aplikacji mobilnej. Następnie, podczas korzystania z aplikacji system automatycznie wybiera miasto, w którym znajduje się użytkownik i pokazuje dostępne samochody na mapie. Samochód może znajdować się w dowolnym miejscu w mieście, w którym dozwolone jest parkowanie i jest dostęp do Internetu. Użytkownik ma możliwość zarezerwowania samochodu na pierwsze 20 minut bezpłatnie z możliwością jednorazowego przedłużenia na kolejne 10 minut pod warunkiem dodatkowej płatności, która od razu następuje po wyborze przedłużenia. W czasie rezerwacji klient może ją anulować, albo ona zostaje anulowana automatycznie po skończeniu czasu rezerwacji.

Po przyjeździe klient informuje nas o tym i sprawdza samochód przez 10 minut oraz zgłasza wszelkie możliwe uszkodzenia wypełniając formularz i, w razie potrzeby, dodając niezbędne zdjęcia. W tym czasie użytkownik może zrezygnować z wynajmu samochodu lub wybrać taryfę i potwierdzić rozpoczęcie podróży. Po upływie 10 minut system automatycznie rezygnuje z wynajmu. Maksymalny okres wynajmu samochodu wynosi 48 godzin. Jeśli klient przekroczy ten czas, zapłaci karę i samochód zabiera pracownik. Podczas podróży można pozostawić samochód w trybie postoju. Aby zakończyć podróż, klient musi zaparkować samochód zgodnie z przepisami ruchu drogowego w dozwolonej strefie parkowania w mieście, w którym samochód został wynajęty. Po zaparkowaniu samochodu użytkownik robi

jego zdjęcie i potwierdza zakończenie podróży. Następnie z jego karty pobierana jest odpowiednia kwota w zależności od czasu podróży i taryfy.

Firma car-sharingowa korzysta z usług innych firm w celu utrzymania swoich pojazdów. System automatycznie zgłasza o potrzebie przeglądu technicznego samochodu. Możliwe jest również zamawianie napraw samochodów od różnych firm.

## **1.2 Procesy występujące w carsharingu oraz obiekty uczestniczące w tych procesach:**

1. Zarządzanie firmą:
  - a. Właściciel firmy.
  - b. Kanał zarządzania.
  - c. Dane dotyczące taryf/personelu/stref/samochodów/współpracujących.
  - d. firm/statystyki.
2. Rejestracja klienta:
  - a. Klient.
  - b. Kanał rejestracji.
  - c. Dane klienta (imię, nazwisko, płeć, hasło, numer telefonu, adres e-mail, karta bankowa, zdjęcie paszportu, zdjęcie prawa jazdy oraz zdjęcie siebie z prawem jazdy, data urodzenia, wybiera miasto zamieszkania z listy dostępnych).
  - d. Pracownicy infolinii.
3. Autoryzacja klienta:
  - a. Klient.
  - b. Kanał autoryzacji (aplikacja).
  - c. Dane do autoryzacji klienta.
4. Rezerwacja samochodu:
  - a. Klient.
  - b. Kanał rezerwacji (aplikacja).
  - c. Dane o aktualnie dostępnych samochodach.
  - d. Transakcja (jednorazowe przedłużenie czasu rezerwacji).
5. Ogląd samochodu
  - a. Klient.
  - b. Samochód.
  - c. Kanał oglądu.
  - d. Formularz uszkodzeń(z możliwością dodawania zdjęć w razie potrzeby)
6. Wypożyczenie samochodu:
  - a. Klient.
  - b. Lokalizacja wynajmu.
  - c. Kanał wynajmu (aplikacja).
  - d. Samochód.
  - e. Stan samochodu.
  - f. Taryfa.
  - g. Czas rozpoczęcia wypożyczenia.
7. Podróż:
  - a. Klient.
  - b. Samochód.
  - c. Kanał do śledzenia lokalizacji wypożyczonego samochodu.
8. Zwrot samochodu:
  - a. Klient.



- b. Samochód.
  - c. Miejsce zwrotu.
  - d. Kanał zwrotu(aplikacja).
  - e. Stan samochodu.
  - f. Transakcja.
  - g. Czas końcowy wypożyczenia.
9. Rejestracja pracownika:
- a. Pracownik.
  - b. Właściciel.
  - c. Kanał rejestracji.
  - d. Dane pracownika (Data zatrudnienia, pensja, wykształcenie).
10. Autoryzacja pracownika:
- a. Pracownik.
  - b. Kanał autoryzacji.
  - c. Dane pracownika (Data zatrudnienia, pensja, wykształcenie).
11. Komunikacja klienta z infolinią:
- a. Klient.
  - b. Kanał komunikacji.
  - c. Pracownicy infolinii.
12. Kontrola techniczna/naprawa samochodów:
- a. Kanał zarządzania.
  - b. Dane dotyczące samochodów wymagających przeglądu technicznego/naprawy.
  - c. Firma, która przeprowadza kontrolę.
  - d. Pracownicy.
  - e. Transakcja.

### **1.3 Problemy, które mają zostać rozwiązane przez system informatyczny:**

1. Konieczność kontaktu na żywo z dostawcą w celu wypożyczenia samochodu.
2. Ograniczone godziny pracy biura wynajmu.
3. Wysoka cena zakupu samochodu osobowego.
4. Nie każda osoba posiada samochód na różne okazje.
5. Problem ograniczonej dostępności pojazdów.
6. Brak możliwości zdalnego podglądu dostępności samochodów do wynajęcia.

## **ROZDZIAŁ 2. SFORMUŁOWANIE CELU BUDOWY SYSTEMU INFORMATYCZNEGO, WYKORZYSTUJĄCEGO PROJEKTOWANĄ BAZĘ DANYCH, ORAZ PREZENTACJA WYMAGAŃ FUNKCJONALNYCH STAWIANYCH SYSTEMOWI**

**Określ cel budowy systemu informatycznego, wykorzystującego projektowaną bazę danych. W definicji celu pamiętaj o zastosowaniu zasady SMART, tj. zapewnij, aby zdefiniowany cel był skonkretyzowany, mierzalny, osiągalny, istotny i określony w czasie.**

**Przedstaw wymagania funkcjonalne jakie powinny być spełnione przez projektowany system IT, aby możliwe było osiągnięcie celu jego budowy. Jeśli system ma być wykorzystywany przez różnego rodzaju użytkowników (np. klient, pracownik, administrator systemu), dokonaj podziału tych wymagań uwzględniając specyficzne potrzeby każdego z nich.**

### **2.1 Cel Budowy Systemu Informatycznego:**

Celem budowy systemu informatycznego jest stworzenie narzędzia umożliwiającego efektywne zarządzanie procesami carsharingu, zwiększenie dostępności usług carsharingu w Polsce oraz ewentualnie w całej Unii Europejskiej. System ten ma na celu eliminację istniejących problemów i ograniczeń, takich jak konieczność kontaktu na żywo z dostawcą, ograniczone godziny pracy biura wynajmu, wysoka cena zakupu samochodu osobowego, ograniczona dostępność pojazdów oraz brak możliwości zdalnego podglądu dostępności samochodów do wynajęcia.

System zostanie wdrożony i uruchomiony w ciągu 12 miesięcy od rozpoczęcia projektu.

### **2.2 Wymagania Funkcjonalne Stawiane Systemowi**

1. Właściciel (strona internetowa):
  - a. Rejestracja.
  - b. Autoryzacja.
  - c. Zarządzania personelem.
  - d. Możliwość odczytu statystyki.
  - e. Możliwość odczytu zgłoszeń klientów dotyczących systemu informatycznego.
2. Klient (aplikacja mobilna):
  - a. Rejestracja.
  - b. Autoryzacja.
  - c. Rezerwacja samochodu.
  - d. Ogląd samochodu.
  - e. Wypożyczenie/zwrot samochodu.
  - f. Możliwość komunikacji z centrum pomocy(infolinia).
  - g. Obsługa płatności.



3. Pracownik (strona internetowa):

- a. Rejestracja.
- b. Autoryzacja.
- c. Obsługa zgłoszeń klientów dotyczących stanu samochodu.
- d. Obsługa zgłoszeń wypadków.
- e. Sprawdzenie dokumentów oraz potwierdzenie konta klientów.
- f. Zarządzanie serwisem samochodów.
- g. Dodawanie/usuwanie samochodów z decyzji właściciela.
- h. Dodawania/usuwania/edycji taryf z decyzji właściciela.
- i. Dodawania/usuwania/edycji stref samochodowych z decyzji właściciela.



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz Społeczny



### **ROZDZIAŁ 3. SZCZEGÓŁOWY OPIS PROCESÓW REALIZOWANYCH W WYBRANEJ RZECZYWISTOŚCI, ZALEŻNOŚCI MIĘDZY NIMI ORAZ OBIEKTÓW W NICH UCZESTNICZĄCYCH**

**Przedstaw szczegółowo procesy jakie będą zachodzić w wybranej rzeczywistości. Jeżeli dany proces ma charakter złożony, dokonaj jego podziału na działania (akcje) elementarne oraz opisz kolejność ich realizacji.**

**Opisz zależności pomiędzy poszczególnymi procesami, wskazując logiczną kolejność ich realizacji.**

**Wskaż (nazwij) obiekty, które będą uczestniczyć w realizacji poszczególnych procesów.**

#### **3.1. Procesy dotyczące strony klienta:**

1. Rejestracja(proces występujący jednorazowo):
  - a. Stworzenie konta: podanie imienia i nazwiska, daty urodzenia(wybór z kalendarza), płci, email'u, numeru telefonu, hasła, informacji dotyczącej karty bankowej (numer karty, CVV, data wygaśnięcia, adres rozliczeniowy, e-mail rozliczeniowy, numer telefonu rozliczeniowego(opcjonalnie)), zdjęcie paszportu, zdjęcie siebie z prawem jazdy, zdjęcie prawa jazdy, wybór miasta zamieszkania z listy dostępnych, pozwolenia na wysyłanie wiadomości na pocztę(opcjonalnie), użycie GPS oraz zaakceptowanie regulaminu.
  - b. Weryfikacja: wysyłanie wiadomości weryfikacyjnej na email, lub numeru weryfikacyjnego na numer telefonu. Wiadomość albo numer mogą być wysłane jeszcze, jeśli klient ma z tym problem(np. nie otrzymał wiadomość weryfikacyjną).
2. Autoryzacja(w przypadku pierwszego logowania do systemu po rejestracji lub w przypadku wylogowania się z konta):
  - a. Podanie email'u lub numeru telefonu.
  - b. Podanie hasła.
3. Możliwość komunikacji z centrum pomocy(infolinia).
  - a. Kontakt telefoniczny z infolinią poprzez telefon kontaktowy.
  - b. Kontakt z infolinią poprzez wypełnienie formularza.
4. Wyszukiwanie dostępnych samochodów:
  - a. Wyszukiwanie dostępnych samochodów w okolicy za pomocą filtrów takich jak odległość samochodu od klienta, ilość paliwa, typ nadwozia, marka, model samochodu.
5. Rezerwacja samochodu:
  - a. Użytkownik wybiera na mapie samochód po czym widzi jego specyfikację: Zdjęcie samochodu, marka samochodu, nazwa modelu, pojemność skokowa silnika, parametry samochodu (długość, szerokość, wysokość), masa, pojemność zbiornika paliwa, prędkość maksymalna, czas przyspieszenia, rodzaj paliwa, rodzaj nadwozia, efektywność paliwowa, pojemność



- pasażerska, pojemność ładunkowa, moc, emisja spalin, rodzaj skrzyni biegów, liczba drzwi, liczba miejsc siedzących, przebieg.
- b. Po potwierdzeniu rezerwacji klient ma 20 minut, żeby dotrzeć do samochodu.
  - c. Klient może zrezygnować z rezerwacji w ciągu tych 20 minut.
  - d. Po upływie tego czasu rezerwacji anulowanie występuje automatycznie.
  - e. Jeśli klient potrzebuje więcej czasu, ma on możliwość jednorazowego przedłużenia rezerwacji o 10 minut za dodatkową opłatę.
  - f. Klient może zrezygnować z rezerwacji w ciągu tych dodatkowych 10 minut.
  - g. Po upływie tego czasu rezerwacji anulowanie występuje automatycznie.
6. Ogląd samochodu:
- a. Po przyjeździe klient potwierdza przybycie i zaczyna ogląd, który trwa 10 minut.
  - b. Podczas oglądu klient zaznacza stan samochodu (obraz nadwozia, obraz szyb, obraz opon, obraz świateł, obraz lusterek samochodowych, obraz wnętrza, czy samochód się uruchamia), wypełniając formularz.
  - c. Klient ma możliwość zrezygnowania z wynajmu.
  - d. Po upływie czasu oglądu anulowanie występuje automatycznie.
  - e. Jeśli klient jest zadowolony stanem samochodu, to on potwierdza koniec oglądu.
7. Wypożyczenie samochodu:
- a. Po potwierdzeniu oglądu następuje wypożyczenie samochodu. Maksymalny czas wypożyczenia - 48 godzin.
  - b. Gdy samochód jest wynajęty system śledzi jego lokalizację.
8. Wypadek:
- a. W razie wypadku klient musi skontaktować się z linią informacyjną przez telefon komórkowy lub wypełniając formularz za pośrednictwem aplikacji.
9. Zwrot samochodu:
- a. Aby zakończyć podróż, klient musi zaparkować samochód zgodnie z przepisami ruchu drogowego w dozwolonej strefie parkowania w mieście, w którym samochód został wynajęty.
  - b. Po zaparkowaniu samochodu użytkownik robi jego zdjęcie i potwierdza zakończenie podróży.
  - c. Opcjonalnie klient może zostawić opinię w specjalnym formularzu w postaci tekstu.
10. Dokonanie płatności:
- a. Po zwrocie samochodu system oblicza kwotę do zapłacenia  $\text{taryfa} * (\text{ReturnDateTime} - \text{RentalDateTime})$ .
  - b. Po tym następuje automatyczne pobranie pieniędzy z preferowanej karty bankowej użytkownika.
  - c. W razie nieudania płatności kwota odejmuje się od salda klienta.
  - d. Jeśli saldo jest mniejsze od 0, to klient nie może rezerwować/wynajmowywać samochody.





Obiekty uczestniczące w procesach :

1. Pracownik.
2. Klient oraz jego dane.
3. Samochód oraz jego dane.
4. Aplikacja mobilna.
5. System płatności.
6. Strefy parkowania.

### **3.2. Procesy dotyczące strony pracowników:**

1. Rejestracja(proces występujący jednorazowo):
  - a. Właściciel rejestruje taką osobę przy zatrudnianiu używając aplikacji internetowej poprzez podanie imienia i nazwiska, daty urodzenia(wybór z kalendarza), płci, zdjęcia profilowego, email'u, numeru telefonu, hasła, zdjęcia paszportu, pensji oraz wykształcenia. Data zatrudnienia ustawia się domyślnie.
2. Autoryzacja(w przypadku pierwszego logowania do systemu po rejestracji lub w przypadku wylogowania się z konta):
  - a. Podanie email'u lub numeru telefonu.
  - b. Podanie hasła.
3. Weryfikacja dokumentów podanych przez użytkowników systemu za pośrednictwem rejestrów państwowych:
  - a. Pracownik sprawdza prawo jazdy przy pomocy rejestrów państwowych.
  - b. Pracownik sprawdza paszport porównując dane podane przez użytkownika oraz wysłane zdjęcia podczas rejestracji z danymi na dokumencie.
  - c. Jeśli wszystkie dane użytkownika sprawdzone, to pracownik potwierdza rejestrację użytkownika. Po potwierdzeniu klient może korzystać się z usług.
4. Obsługa wypadków:
  - a. Jeśli klient zgłosił wypadek przez wypełnienie formularza, to system powiadamia pracownika o tym. W razie kontaktu telefonicznego klient opowiada o problemie i pracownik wypełnia formularz za klienta.
  - b. Pracownik jest odpowiedzialny za rozwiązywanie zgłoszeń wypadków poprzez wezwania policji, straży pożarnej, pogotowia lub lawetę w razie potrzeby.
  - c. Po rozwiązaniu sytuacji z wypadkiem pracownik musi też zarządzać naprawą uszkodzonego samochodu.
5. Obsługa zgłoszeń klientów dotyczących stanu samochodu:
  - a. System powiadamia pracowników o problemach zgłoszonych przez klientów przy oglądzie samochodu.
  - b. Pracownik sprawdza zgłoszenie klienta, oglądając dołączone przez klienta zdjęcia uszkodzeń.
  - c. W razie potrzeby pracownik zamawia naprawę/sprzątanie.
6. Dodawanie/usuwanie samochodów:

- a. Po zakupie samochodów właściciel powierza zadanie ich dodania do systemu pracownikom.
  - b. Pracownicy dodają samochody do systemu podając: model, pojemność silnika, szerokość, długość, wysokość, pojemność zbiornika paliwa, maksymalną prędkość, przyspieszenie, oszczędność paliwa, przebieg, pojemność pasażerską, zdolność przewozową, moc silnika, opcjonalnie emisję zanieczyszczeń, ilość drzwi, ilość miejsc siedzących, taryfę, zdjęcie samochodu oraz wybierając z listy dostępnych: typ nadwozia, firmę, skrzynię biegów, typ paliwa, przebieg.
  - c. Pracownicy usuwają samochody zgodnie z życzeniem właściciela.
7. Dodawanie/usuwanie/edycja taryf:
- a. Po obejrzeniu statystyk właściciel powierza zadanie dodawania/usuwania/edycji taryf pracownikom.
  - b. Pracownicy dodają/usuwają/edytują taryfy zgodnie z życzeniem właściciela.
8. Dodawanie/usuwanie/edycja stref samochodowych:
- a. Po obejrzeniu statystyk właściciel powierza zadanie dodawania/usuwania/edycji stref samochodowych pracownikom.
  - b. Pracownicy dodają/usuwają/edytują strefy samochodowe zgodnie z życzeniem właściciela.
9. Zarządzanie serwisem samochodu:
- a. System automatycznie informuje o nastąpieniu terminu przeglądu okresowego konkretnego samochodu.
  - b. Pracownik firmy musi skontaktować się z firmą serwisową, z którą współpracuje nasza firma carsharing'owa.
  - c. Umawia się na przeprowadzenie przeglądu oraz tymczasowo usuwa samochód z mapy dostępnych samochodów.

Obiekty uczestniczące w procesach :

1. Pracownik i jego dane.
2. Klient oraz jego dane.
3. Samochód oraz jego dane.
4. Strona internetowa.

### **3.3. Procesy dotyczące strony właściciela biznesu**

1. Rejestracja:
  - a. Właściciel biznesu jest rejestrowany przez deweloperów systemu przez podanie(imienia i nazwiska, zdjęcia profilowego, daty urodzenia(wybór z kalendarza), płci, email'u, numeru telefonu, hasła oraz zdjęcia paszportu).
2. Autoryzacja(w przypadku pierwszego logowania do systemu po rejestracji lub w przypadku wylogowania się z konta):
  - a. Podanie email'u lub numeru telefonu.
  - b. Podanie hasła.



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz Społeczny



3. Zarządzania personelem.
  - a. Właściciel może dodawać/usuwać pracowników do/z systemu.
  - b. Właściciel ma możliwość edycji danych dotyczących pracownika(jak n.p. rozmiaru pensji).
4. Możliwość odczytu statystyki.
  - a. Właściciel ma możliwość odczytu statystyki dotyczących poziomu zadowolenia klientów, popytu na poszczególne marki i modele samochodów, zapotrzebowania na nasz system w poszczególnych strefach.
5. Możliwość odczytu zgłoszeń klientów dotyczących systemu informatycznego.
  - a. System udostępnia zgłoszenia klientów dotyczących systemu informatycznego.
  - b. Na podstawie zgłoszeń właściciel decyduje o rozwoju systemu.

Obiekty uczestniczące w procesach :

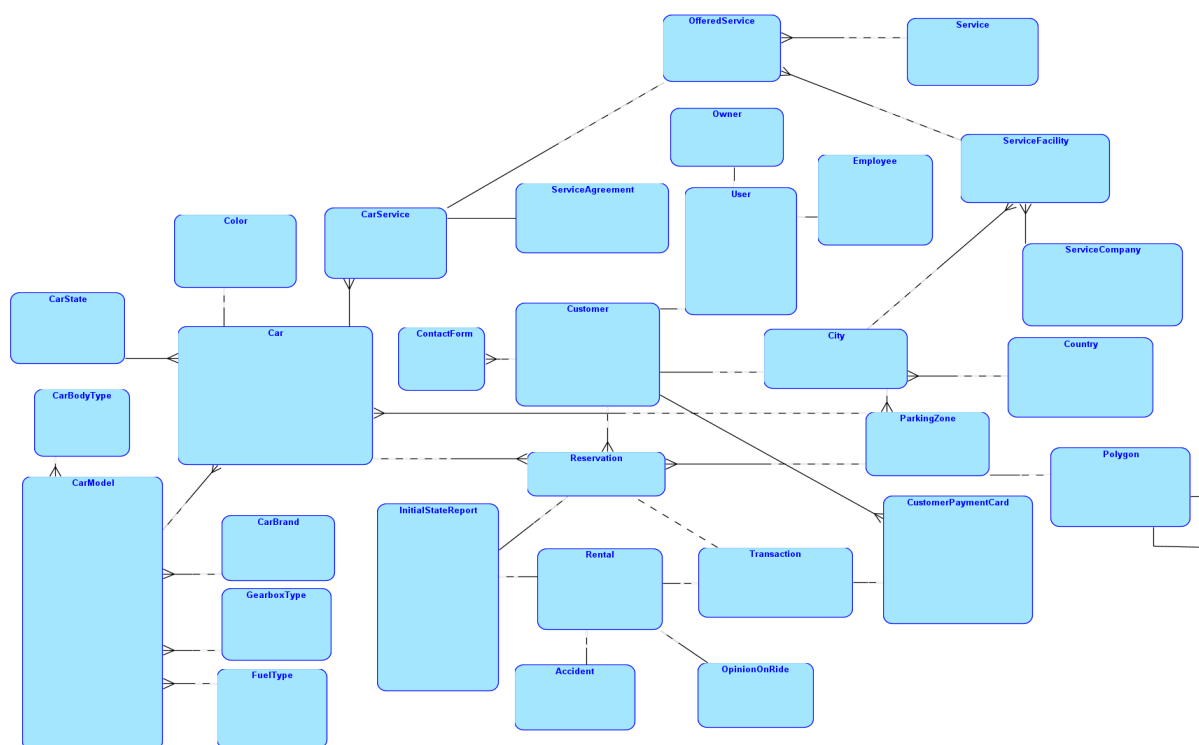
1. Właściciel oraz jego dane.
2. Pracownicy oraz ich dane.
3. Strona internetowa.

## ROZDZIAŁ 4. MODEL KONCEPTUALNY PROJEKTOWANEJ BAZY DANYCH

Przedstaw w formie graficznej model konceptualny projektowanej bazy danych.

Wymień i scharakteryzuj encje, które będą uwzględnione w modelowanej bazie danych. Podaj ich nazwy oraz opisz ogólnie, jakie obiekty rzeczywistości będą one opisywały. Opisz rodzaje związków pomiędzy poszczególnymi encjami.

### Graficzny model konceptualny projektowanej bazy danych:



### Encje:

1. Car - reprezentuje pojazdy w systemie, zawiera takie atrybuty jak numer rejestracyjny, kod VIN, rok produkcji, numer polisy ubezpieczeniowej, poziom paliwa, datę ostatniego przeglądu, przebieg oraz odniesienia do modelu samochodu, marki, typu nadwozia, koloru, stanu pojazdu, rodzaju paliwa i skrzyni biegów.
2. CarBrand - opisuje marki samochodów, posiada nazwę marki.
3. CarModel - zawiera modele samochodów, ich nazwy oraz parametry takie jak wymiary, masę, pojemność bagażnika itd.
4. Color - definiuje kolory samochodów.
5. CarBodyType - określa typy nadwozi pojazdów.
6. CarState - informuje o stanie pojazdów, czy są dostępne czy nie.
7. FuelType - określa rodzaje paliwa, np. benzyna, diesel.
8. GearboxType - typy skrzyń biegów: manualna, automatyczna.
9. Service - usługi serwisowe dostępne dla pojazdów.
10. ServiceCompany - firmy świadczące usługi serwisowe.

- Związki między tymi encjami:**

-

- 19. City - ServiceFacility: Związek jeden do wielu (1:N)
- 20. Country - City: Związek jeden do wielu (1:N)
- 21. Owner - User: Związek jeden do jednego (1:1)
- 22. User - Customer: Związek jeden do jednego (1:1)
- 23. Employee - User: Związek jeden do jednego (1:1)
- 24. Customer - ContactForm: Związek jeden do wielu (1:N)
- 25. Customer - CustomerPaymentCard: Związek jeden do wielu (1:N)
- 26. Customer - Reservation: Związek jeden do wielu (1:N)
- 27. Reservation - Transaction: Związek jeden do jednego (1:1)
- 28. Reservation - InitialStateReport: Związek jeden do jednego (1:1)
- 29. InitialStateReport - Rental: Związek jeden do jednego (1:1)
- 30. Rental - Accident: Związek jeden do jednego (1:1)
- 31. Rental - OpinionOnRide: Związek jeden do jednego (1:1)
- 32. Rental - Transaction: Związek jeden do jednego (1:1)
- 33. Transaction - CustomerPaymentCard: Związek jeden do jednego (1:1)



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz Społeczny



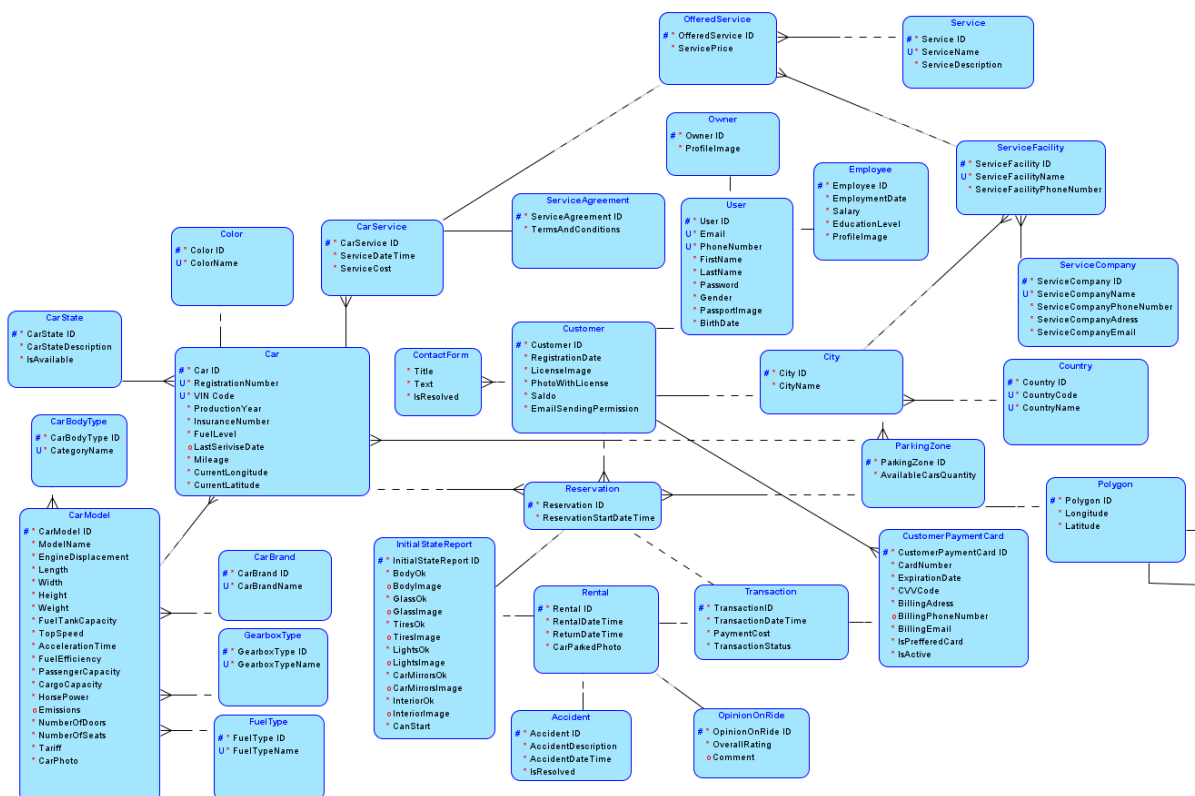
## ROZDZIAŁ 5. MODEL ZWIĄZKÓW ENCJI PROJEKTOWANEJ BAZY DANYCH

Przedstaw w formie graficznej model logiczny (związków encji) projektowanej bazy danych.

Wymień i opisz znaczenie atrybutów dla poszczególnych encji występujących w prezentowanym modelu logicznym bazy danych. Podaj nazwę i rodzaj każdego atrybutu encji.

Określ dla każdego atrybutu encji jego dziedzinę, maksymalny rozmiar danych, opcjonalność lub konieczność wystąpienia wartości określonej oraz inne ograniczenia integralnościowe, np. unikalność wartości atrybutu bądź też zawężenie jego dziedziny.

Dokonaj szczegółowego opisu każdego związku pomiędzy poszczególnymi encjami.



Car:

1. Car\_ID (Primary Key): Unikalny identyfikator samochodu, NUMERIC(6), dziedzina to liczby całkowite, musi być unikalny i jest wymagany.
2. RegistrationNumber: Numer rejestracyjny pojazdu, VARCHAR(7), unikalność wymagana.
3. VINCode: Numer identyfikacyjny pojazdu, VARCHAR(20), wymagany.
4. ProductionYear: Rok produkcji, data, wymagany.
5. FuelLevel: Poziom paliwa, liczby rzeczywiste, FLOAT(5), wymagany.
6. LastServiceDate: Data ostatniego przeglądu, data, nie jest wymagany.
7. Mileage: Przebieg, NUMERIC(7), dziedzina to liczby całkowite, wymagany.
8. CurrentLongitude: Współrzędna długości geograficznej bieżącej lokalizacji samochodu, liczby rzeczywiste, DECIMAL(9, 6), wymagany.



9. CurrentLatitude: Współrzędna szerokości geograficznej bieżącej lokalizacji samochodu, liczby rzeczywiste, DECIMAL(8, 6), wymagany.

Encja: CarModel

1. CarModel\_ID (Primary Key): Unikalny identyfikator modelu samochodu, NUMERIC(8), musi być unikalny i jest wymagany.
2. ModelName: Nazwa modelu samochodu, ciąg znaków, VARCHAR(30)
3. EngineDisplacement: Pojemność silnika, liczba zmiennoprzecinkowa, FLOAT(5).
4. Length: Długość samochodu, liczba rzeczywista, FLOAT(5).
5. Width: Szerokość samochodu, liczba rzeczywista, FLOAT(5).
6. Height: Wysokość samochodu, liczba rzeczywista, FLOAT(5).
7. Weight: Masa samochodu, liczba rzeczywista, FLOAT(5).
8. FuelTankCapacity: Pojemność zbiornika paliwa, liczba rzeczywista, FLOAT(5).
9. TopSpeed: Maksymalna prędkość samochodu, liczba rzeczywista, FLOAT(5).
10. AccelerationTime: Czas przyspieszenia do 100 km/h, liczba rzeczywista, FLOAT(5).
11. FuelEfficiency: Zużycie paliwa, liczba rzeczywista, FLOAT(5).
12. PassengerCapacity: Liczba miejsc pasażerskich, liczba całkowita, NUMERIC(2).
13. CargoCapacity: Pojemność bagażnika, FLOAT(5), liczba rzeczywista.
14. HorsePower: Moc silnika, NUMERIC(3), liczba całkowita.
15. Emissions: Emisje spalin, FLOAT(5), liczba rzeczywista.
16. NumberOfSeats: Liczba miejsc siedzących, NUMERIC(2), liczba całkowita.
17. Tariff: Stawka za wynajem samochodu, liczba rzeczywista, FLOAT(5).
18. CarPhoto: Zdjęcie samochodu, Image.

Encja: CarBodyType

1. CarBodyType\_ID (Primary Key): Unikalny identyfikator typu nadwozia samochodu, dziedzina to liczby całkowite, NUMERIC(5), musi być unikalny i jest wymagany.
2. CategoryName: Nazwa kategorii nadwozia samochodu, ciąg znaków, VARCHAR(5)

Encja: CarBrand

1. CarBrand\_ID (Primary Key): Unikalny identyfikator marki samochodu, dziedzina to liczby całkowite, NUMERIC(3), musi być unikalny i jest wymagany.
2. CarBrandName: Nazwa marki samochodu, ciąg znaków, VARCHAR(15).

Encja: GearboxType

1. GearboxType\_ID (Primary Key): Unikalny identyfikator typu skrzyni biegów, dziedzina to liczby całkowite, NUMERIC(1), musi być unikalny i jest wymagany.
2. GearboxTypeName: Nazwa typu skrzyni biegów, ciąg znaków, VARCHAR(15).

Encja: FuelType

1. FuelType\_ID (Primary Key): Unikalny identyfikator typu paliwa, dziedzina to liczby całkowite, NUMERIC(1), musi być unikalny i jest wymagany.
2. FuelTypeName: Nazwa typu paliwa, ciąg znaków, VARCHAR(15).

Encja: ParkingZone

1. ParkingZoneID (Primary Key): Unikalny identyfikator strefy parkingowej, dziedzina to liczby całkowite, NUMERIC(7), musi być unikalny i jest wymagany.





2. AvailableCarsQuantity: Dostępna liczba samochodów w strefie, dziedzina to liczby całkowite, NUMERIC(10), nie może być NULL.

Encja: Polygon

1. Polygon ID (Primary Key): Unikalny identyfikator strefy parkingowej, dziedzina to liczby całkowite, musi być unikalny i jest wymagany.
2. Longitude: Współrzędna długości geograficznej poligonu, DECIMAL(9, 6), nie jest wymagany.
3. Latitude: Współrzędna szerokości geograficznej poligonu, DECIMAL(8, 6), nie jest wymagany.

Encja: Reservation

1. ReservationID (Primary Key): Unikalny identyfikator rezerwacji, dziedzina to liczby całkowite, NUMERIC(20), musi być unikalny i jest wymagany.
2. ReservationStartDateTime: Data i czas rozpoczęcia rezerwacji, Datetime, wymagany.

Encja: InitialStateReport

1. InitialStateReport\_ID (Primary Key): Unikalny identyfikator raportu o stanie początkowym, dziedzina to liczby całkowite, NUMERIC(20), musi być unikalny i jest wymagany.
2. BodyOk: Flaga określająca, czy karoseria samochodu jest w dobrym stanie, wartość logiczna, jest wymagany.
3. GlassOk: Flaga określająca, czy szyby samochodu są w dobrym stanie, wartość logiczna, Boolean, jest wymagany.
4. TiresOk: Flaga określająca, czy opony samochodu są w dobrym stanie, wartość logiczna, Boolean, jest wymagany.
5. LightsOk: Flaga określająca, czy światła samochodu są w dobrym stanie, wartość logiczna, Boolean, jest wymagany.
6. CarMirrorsOk: Flaga określająca, czy lusterka samochodu są w dobrym stanie, wartość logiczna, Boolean, jest wymagany.
7. InteriorOk: Flaga określająca, czy wnętrze samochodu jest w dobrym stanie, wartość logiczna, Boolean, jest wymagany.
8. BodyImage: Zdjęcie karoserii samochodu, obraz, Image.
9. GlassImage: Zdjęcie szyb samochodu, obraz, Image.
10. TiresImage: Zdjęcie opon samochodu, obraz, Image.
11. LightsImage: Zdjęcie świateł samochodu, obraz, Image.
12. CarMirrorsImage: Zdjęcie lusterek samochodu, obraz, Image.
13. InteriorImage: Zdjęcie wnętrza samochodu, obraz, Image.

Encja: Rental

1. Rental\_ID (Primary Key): Unikalny identyfikator rezerwacji, dziedzina to liczby całkowite, NUMERIC(20), musi być unikalny i jest wymagany.
2. RentalDateTime: Data i czas rozpoczęcia rezerwacji, Datetime.
3. ReturnDateTime: Data i czas zakończenia rezerwacji, Datetime.
4. CarParkedPhoto: Zdjęcie zaparkowanego samochodu, obraz, Image.

Encja: Accident

1. Accident ID (Primary Key): Unikalny identyfikator wypadku drogowego, dziedzina to liczby całkowite, NUMERIC(15), musi być unikalny i jest wymagany.
2. AccidentDescription: Opis wypadku, ciąg znaków, VARCHAR(100).
3. AccidentDateTime: Data i czas wypadku, Datetime.
4. IsResolved: indykator rozwiązywania wypadku, wartość logiczna, Boolean.

Encja: OpinionOnRide

1. OpinionOnRideID (Primary Key): Unikalny identyfikator opinii na temat przejazdu, dziedzina to liczby całkowite, NUMERIC(20), musi być unikalny i jest wymagany.
2. OverallRating: Ocena ogólna, dziedzina to liczby całkowite, NUMERIC(2), nie może być NULL.
3. Comment: Komentarz, dziedzina to ciąg znaków, VARCHAR(100), nie jest wymagany.

Encja: Transaction

1. TransactionID (Primary Key): Unikalny identyfikator transakcji, dziedzina to liczby całkowite, NUMERIC(20), musi być unikalny i jest wymagany.
2. TransactionDateTime: Data i czas transakcji, Datetime, nie może być NULL.
3. PaymentCost: Koszt transakcji, NUMERIC(10), maksymalny rozmiar 10 cyfr po przecinku, nie może być NULL.
4. TransactionStatus: Status transakcji, dziedzina to ciąg znaków, VARCHAR(10), nie może być NULL.

Encja: User

1. UserID (Primary Key): Unikalny identyfikator użytkownika, NUMERIC(12), liczby całkowite, wymagany.
2. Email: Adres e-mail użytkownika, ciąg znaków, format e-mail, VARCHAR(30), unikalność wymagana, wymagany.
3. Phone Number: Numer telefonu, ciąg znaków, format zależny od kraju, VARCHAR(20), wymagany.
4. FirstName, LastName: Imię i nazwisko użytkownika, ciągi znaków, VARCHAR(20), wymagane.
5. Password: Hasło użytkownika, ciąg znaków, VARCHAR(40), wymagane, zaszyfrowane przechowywanie.
6. Gender: Płeć użytkownika, ciąg znaków, VARCHAR(10), wymagane.
7. PassportImage: Zdjęcie paszportu użytkownika, obraz, Image, wymagane.
8. BirthDate: Data urodzenia użytkownika, Date, wymagana.

Encja: Customer

1. CustomerID (Primary Key): Unikalny identyfikator klienta, dziedzina to liczby całkowite, NUMERIC(20), musi być unikalny i jest wymagany.
2. RegistrationDate: Data rejestracji klienta, Date, wymagany.
3. LicenceImage: Zdjęcie prawa jazdy klienta, obraz, Image, jest wymagany.
4. PhotoWithLicense : Obraz zdjęcia klienta z prawem jazdy, obraz, Image, jest wymagany.
5. Saldo : Stan konta klienta, dziedzina to liczby rzeczywiste, FLOAT(10), wymagany.
6. EmailSendingPermission: Czy klient wyraził zgodę na otrzymywanie wiadomości e-mail, wartość logiczna, Boolean, wymagany.



Encja: CustomerPaymentCard

1. CustomerPaymentCardID (Primary Key): Unikalny identyfikator karty płatniczej klienta, dziedzina to liczby całkowite, NUMERIC(16), musi być unikalny i jest wymagany.
2. CardNumber: Numer karty, dziedzina to liczby całkowite, NUMERIC(16), nie może być NULL.
3. ExpirationDate: Data ważności karty, Date, nie może być NULL.
4. CVVCode: Kod CVV/CVC, dziedzina to ciąg znaków, maksymalny rozmiar 3 znaki, nie może być NULL.
5. BillingAddress: Adres rozliczeniowy, ciąg znaków, VARCHAR(20), jest wymagany.
6. BillingPhoneNumber: Numer telefonu rozliczeniowy, ciąg znaków, VARCHAR(12), nie jest wymagany.
7. BillingEmail: Adres e-mail rozliczeniowy, ciąg znaków, VARCHAR(30), jest wymagany.
8. IspreferredCard: Flaga określająca, czy karta jest preferowana dla opłaty usług, wartość logiczna, Boolean, jest wymagany.
9. IsActive: Flaga określająca, czy karta bankowa jest aktywna i dostępna dla ściągania środków, wartość logiczna, Boolean, jest wymagany.

Encja: Employee

1. EmployeeID (Primary Key): Unikalny identyfikator pracownika firmy, dziedzina to liczby całkowite, NUMERIC(5), musi być unikalny i jest wymagany.
2. EmploymentDate: Data zatrudnienia pracownika, Date, nie może być NULL.
3. Salary: Wynagrodzenie pracownika, dziedzina to liczby całkowite, NUMERIC(12), nie może być NULL.
4. EducationLevel: Wykształcenie pracownika, ciąg znaków, VARCHAR(50), wymagane.
5. ProfileImage: Zdjęcie profilowe pracownika, obraz, Image, wymagane.

Encja: Owner

1. OwnerID (Primary Key): Unikalny identyfikator właściciela firmy, dziedzina to liczby całkowite, NUMERIC(2), musi być unikalny i jest wymagany.
2. ProfileImage: Obraz profilowy właściciela, obraz, Image, jest wymagany.

Encja: City

1. CityID (Primary Key): Unikalny identyfikator miasta, dziedzina to liczby całkowite, NUMERIC(5), musi być unikalny i jest wymagany.
2. CityName: Nazwa miasta, dziedzina to ciąg znaków, VARCHAR(20), nie może być NULL.

Encja: Country



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz Społeczny



1. CountryID (Primary Key): Identyfikator kraju, dziedzina to liczby całkowite, NUMERIC(3), nie może być NULL.
2. CountryCode: Kod kraju, dziedzina to ciąg znaków, VARCHAR(3), nie może być NULL.
3. CountryName: Nazwa kraju, dziedzina to ciąg znaków, VARCHAR(20), nie może być NULL.

Encja: ServiceFacility

1. ServiceFacilityID (Primary Key): Unikalny identyfikator firmy świadczącej usługi serwisu samochodów, dziedzina to liczby całkowite, NUMERIC(7), musi być unikalny i jest wymagany.
2. ServiceFacilityName: Nazwa firmy, ciąg znaków, VARCHAR(20), musi być unikalny, nie może być NULL.
3. ServiceFacilityPhoneNumber: Numer telefonu firmy, ciąg znaków, VARCHAR(20), jest wymagany.

Encja: ServiceCompany

1. ServiceCompanyID (Primary Key): Unikalny identyfikator firmy świadczącej usługi serwisu samochodów, dziedzina to liczby całkowite, NUMERIC(4), musi być unikalny i jest wymagany.
2. ServiceCompanyName: Nazwa firmy, to ciąg znaków, VARCHAR(20), nie może być NULL.
3. ServiceCompanyPhoneNumber: Numer telefonu firmy, ciąg znaków, VARCHAR(30), jest wymagany.
4. ServiceCompanyAddress: Adres firmy, to ciąg znaków, VARCHAR(30), jest wymagany.
5. ServiceCompanyEmail: Adres e-mail firmy, ciąg znaków, VARCHAR(20), jest wymagany.

Encja: Service

1. ServiceID (Primary Key): Unikalny identyfikator usługi, dziedzina to liczby całkowite, NUMERIC(4), musi być unikalny i jest wymagany.
2. ServiceName: Nazwa usługi, ciąg znaków, VARCHAR(20), nie może być NULL.
3. ServiceDescription: Opis usługi, ciąg znaków, VARCHAR(150), jest wymagany.

Encja: OfferedService

1. OfferedServiceID (Primary Key): Unikalny identyfikator usługi oferowanej przez firmę, liczby całkowite, NUMERIC(5), musi być unikalny i jest wymagany.
2. ServicePrice: Cena usługi, liczby całkowite, NUMERIC(6), nie może być NULL.

Encja: CarService

1. CarService\_ID (Primary Key): Unikalny identyfikator usługi serwisowej, liczby całkowite, NUMERIC(9), musi być unikalny i jest wymagany.
2. ServiceDateTime: Data i godzina wykonania usługi, Datetime.
3. ServiceCost: Koszt usługi, liczba całkowita, NUMERIC(9), nie może być NULL.

Encja: ServiceAgreement

1. ServiceAgreement\_ID (Primary Key): Unikalny identyfikator umowy serwisowej, dziedzina to liczby całkowite, NUMERIC(9), musi być unikalny i jest wymagany.



2. TermsAndConditions: Treść warunków umowy serwisowej, ciąg znaków, VARCHAR(500), jest wymagany.

### **Relacje między encjami:**

1. CarModel - Car: Związek jeden do wielu (1:N). Każdy samochód (Car) jest określonego modelu (CarModel), a jeden model może być wspólny dla wielu samochodów.
2. Car - Color: Związek jeden do jednego (1:1). Każdy samochód jest przypisany do dokładnie jednego koloru (Color).
3. CarState - Car: Związek jeden do wielu (1:N). Każdy samochód posiada stan (CarState), który określa np. czy jest dostępny czy w naprawie. Jeden stan może dotyczyć wielu samochodów.
4. Car - CarService: Związek jeden do wielu (1:N). Każdy samochód może przechodzić przez różne rodzaje serwisów (CarService), a każdy serwis może obejmować wiele różnych samochodów.
5. Car - Reservation: Związek jeden do wielu (1:N). Każdy samochód może być zarezerwowany wiele razy (Reservation), a każda rezerwacja dotyczy jednego konkretnego samochodu.
6. ParkingZone - Car: Związek jeden do wielu (1:N). Strefy parkingowe (ParkingZone) mogą pomieścić wiele samochodów, a każdy samochód jest przypisany do jednej strefy.
7. Polygon - ParkingZone: Związek jeden do jednego (1:1). Każda strefa parkingowa jest zdefiniowana przez unikatowy obszar geograficzny (Polygon).
8. ParkingZone - Reservation: Związek jeden do wielu (1:N). Miejsca parkingowe są zarezerwowane dla określonych rezerwacji; jedno miejsce parkingowe może być używane przez wiele rezerwacji.
9. City - ParkingZone: Związek jeden do wielu (1:N). Miasta (City) mogą zawierać wiele stref parkingowych, które są z nimi związane.
10. CarBrand - CarModel: Związek jeden do wielu (1:N). Marki samochodów (CarBrand) oferują różne modele (CarModel), a każdy model jest powiązany z jedną marką.
11. GearboxType - CarModel: Związek jeden do wielu (1:N). Modele samochodów mają określone typy skrzyń biegów (GearboxType), a jeden typ skrzyni biegów może być używany w różnych modelach.
12. FuelType - CarModel: Związek jeden do wielu (1:N). Modele samochodów mają określone typy paliw (FuelType), które mogą być wspólne dla różnych modeli.
13. CarBodyType - CarModel: Związek jeden do wielu (1:N). Modele samochodów są związane z typami nadwozia (CarBodyType), które określają ich ogólną strukturę i kształt.
14. ServiceAgreement - CarService: Związek jeden do jednego (1:1). Umowy serwisowe (ServiceAgreement) regulują warunki konkretnych usług serwisowych.



15. CarService - OfferedService: Związek jeden do wielu (1:N). Usługi serwisowe mogą obejmować różne oferowane usługi (OfferedService), które są częścią serwisów samochodowych.
16. Service - OfferedService: Związek jeden do wielu (1:N). Oferowane usługi są częścią szerszego zakresu usług (Service), które są dostępne dla klientów.
17. ServiceFacility - OfferedService: Związek jeden do wielu (1:N). Placówki serwisowe (ServiceFacility) mogą oferować różne usługi w ramach swojej działalności.
18. ServiceCompany - ServiceFacility: Związek jeden do wielu (1:N). Firmy serwisowe (ServiceCompany) zarządzają placówkami serwisowymi i świadczą różnorodne usługi.
19. City - ServiceFacility: Związek jeden do wielu (1:N). Placówki serwisowe znajdują się w miastach, które mogą mieć wiele takich placówek.
20. Country - City: Związek jeden do wielu (1:N). Miasta są częścią krajów (Country), a każde miasto należy do jednego kraju.
21. Owner - User: Związek jeden do jednego (1:1). Właściciel (Owner) jest jednocześnie użytkownikiem (User) w systemie.
22. User - Customer: Związek jeden do jednego (1:1). Klienci (Customer) są użytkownikami systemu, którzy korzystają z usług.
23. Employee - User: Związek jeden do jednego (1:1). Pracownicy (Employee) są użytkownikami systemu i mogą mieć różne role.
24. Customer - ContactForm: Związek jeden do wielu (1:N). Klienci mogą wysyłać wiele form kontaktowych (ContactForm) z zapytaniami lub uwagami.
25. Customer - CustomerPaymentCard: Związek jeden do wielu (1:N). Każdy klient posiada przynajmniej jedną kartę płatniczą (CustomerPaymentCard), która jest używana do transakcji.
26. Customer - Reservation: Związek jeden do wielu (1:N). Klienci mogą dokonywać wielu rezerwacji (Reservation).
27. Reservation - Transaction: Związek jeden do jednego (1:1). Każda rezerwacja jest powiązana z transakcją (Transaction), która odnosi się do płatności lub umowy.
28. Reservation - InitialStateReport: Związek jeden do jednego (1:1). Przy każdej rezerwacji sporządzany jest raport stanu początkowego (InitialStateReport), który dokumentuje warunki przed wynajmem.
29. InitialStateReport - Rental: Związek jeden do jednego (1:1). Każdy wynajem (Rental) jest dokumentowany raportem stanu początkowego, który odnosi się do stanu pojazdu na początku wynajmu.
30. Rental - Accident: Związek jeden do jednego (1:1). W przypadku wypadków (Accident) związanych z wynajmowanymi pojazdami, każdy wynajem ma powiązany unikatowy raport wypadku.
31. Rental - OpinionOnRide: Związek jeden do jednego (1:1). Po każdym wynajmie klienci mogą wyrazić swoją opinię (OpinionOnRide) na temat przejazdu.
32. Rental - Transaction: Związek jeden do jednego (1:1). Każdy wynajem wiąże się z unikatową transakcją, która odnosi się do płatności lub innych warunków finansowych.





33. Transaction - CustomerPaymentCard: Związek jeden do jednego (1:1). Każda transakcja (Transaction) jest bezpośrednio powiązana z jedną kartą płatniczą klienta (CustomerPaymentCard), która jest używana do jej realizacji.



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**



**Unia Europejska**  
Europejski Fundusz Społeczny







## **6.2 Ograniczenia integralnościowe:**

### **Ograniczenia relacji:**

- 1. Klucz główny**
- 2. Klucz obcy**

### **Ograniczenia atrybutu:**

- 1. Wartość musi być unikatowa**
- 2. Wartość nie może być NULL**

Encja: Service

1. Service\_ID - PrimaryKey(Service\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. ServiceName - UNIQUE(Service\_ServiceName\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna.
3. ServiceDescription - NOT NULL - wartość nie może być pusta.

Encja: OfferedService

1. ServiceFacility\_ServiceFacility\_ID - ForeignKey(OfferedService\_ServiceFacility\_FK), NOT NULL - klucz obcy odwołujący się do tabeli ServiceFacility, wartość nie może być pusta.
2. OfferedService\_ID - PrimaryKey(OfferedService\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
3. ServicePrice - NOT NULL - wartość nie może być pusta.
4. Service\_Service\_ID - ForeignKey(OfferedService\_Service\_FK), NOT NULL - klucz obcy odwołujący się do tabeli Service, wartość nie może być pusta.
5. ServiceName - NOT NULL - wartość nie może być pusta.

Encja: CarService

1. CarService\_ID - PrimaryKey(CarService\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. Car\_Car\_ID - ForeignKey(CarService\_Car\_FK), NOT NULL - klucz obcy odwołujący się do tabeli Car, wartość nie może być pusta.
3. OfferedService\_OfferedService\_ID - ForeignKey(CarService\_OfferedService\_FK), NOT NULL, Index(CarService\_\_IDXv1) - klucz obcy odwołujący się do tabeli OfferedService, wartość nie może być pusta, indeks.
4. ServiceAgreement\_ServiceAgreement\_ID - ForeignKey(CarService\_Car\_FK), NOT NULL, Index(CarService\_\_IDX) - klucz obcy odwołujący się do tabeli OfferedService, wartość nie może być pusta, indeks.
5. ServiceDateTime - NOT NULL - wartość nie może być pusta.
6. ServiceCost - NOT NULL - wartość nie może być pusta.



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz Społeczny



ServiceAgreement:

1. ServiceAgreement\_ID - PrimaryKey(ServiceAgreement\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. TermsAndConditions - NOT NULL - wartość nie może być pusta.
3. CarService\_CarService\_ID - ForeignKey(ServiceAgreement\_CarService\_FK), NOT NULL, Index(ServiceAgreement\_\_IDX) - klucz obcy odwołujący się do tabeli CarService, wartość nie może być pusta, indeks.ServiceFacility:
  1. ServiceFacility\_ID - PrimaryKey(ServiceFacility\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
  2. ServiceFacilityName - UNIQUE(ServiceFacility\_ServiceFacilityName\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna.
  3. ServiceFacilityPhoneNumber - NULL - wartość może być pusta.
  4. City\_City\_ID - ForeignKey(ServiceFacility\_City\_FK), NOT NULL - klucz obcy odwołujący się do tabeli City, wartość nie może być pusta.
  5. ServiceCompany\_ServiceCompany\_ID - ForeignKey(ServiceFacility\_ServiceCompany\_FK), NOT NULL - klucz obcy odwołujący się do tabeli ServiceCompany, wartość nie może być pusta.

Encja: ServiceFacility

1. ServiceFacility\_ID - PrimaryKey(ServiceFacility\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. ServiceFacilityName - UNIQUE(ServiceFacility\_ServiceFacilityName\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna.
3. ServiceFacilityPhoneNumber - NULL - wartość może być pusta.
4. City\_City\_ID - ForeignKey(ServiceFacility\_City\_FK), NOT NULL - klucz obcy odwołujący się do tabeli City, wartość nie może być pusta.
5. ServiceCompany\_ServiceCompany\_ID - ForeignKey(ServiceFacility\_ServiceCompany\_FK), NOT NULL - klucz obcy odwołujący się do tabeli ServiceCompany, wartość nie może być pusta.

Encja: ServiceCompany

1. ServiceCompany\_ID - PrimaryKey(ServiceCompany\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. ServiceCompanyName - UNIQUE(ServiceCompany\_ServiceCompanyName\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna.
3. ServiceCompanyPhoneNumber - NOT NULL - wartość nie może być pusta.
4. ServiceCompanyAddress - NOT NULL - wartość nie może być pusta.
5. ServiceCompanyEmail - NOT NULL - wartość nie może być pusta.



Encja: User

1. User\_ID - PrimaryKey(User\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. Email - UNIQUE(User\_Email\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna w całej tabeli.
3. PhoneNumber - UNIQUE(User\_PhoneNumber\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna w całej tabeli.
4. Password - NOT NULL - wartość nie może być pusta.
5. FirstName - NOT NULL - wartość nie może być pusta.
6. LastName - NOT NULL - wartość nie może być pusta.
7. Password - NOT NULL - wartość nie może być pusta.
8. Gender - NOT NULL - wartość nie może być pusta.
9. PassportImage - NOT NULL - wartość nie może być pusta.
10. BirthDate - NOT NULL - wartość nie może być pusta.

Encja: Employee

1. Employee\_ID - PrimaryKey(Employee\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. User\_User\_ID - ForeignKey(Employee\_User\_FK), NOT NULL, Index(User\_User\_ID) - klucz obcy odwołujący się do tabeli User, wartość nie może być pusta, indeks.
3. EmploymentDate - NOT NULL - wartość nie może być pusta.
4. Salary - NOT NULL - wartość nie może być pusta.
5. EducationLevel - NOT NULL - wartość nie może być pusta.
6. ProfileImage - NOT NULL - wartość nie może być pusta.

Encja: Owner

1. Owner\_ID - PrimaryKey(Owner\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. User\_User\_ID - ForeignKey(Owner\_User\_FK), Index(Owner\_\_IDX), NOT NULL - klucz obcy odwołujący się do tabeli User, wartość nie może być pusta, indeks.
3. ProfileImage - NOT NULL - wartość nie może być pusta.

Encja: Customer

1. Customer\_ID - PrimaryKey(Customer\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. User\_User\_ID - ForeignKey(Customer\_User\_FK), NOT NULL, Index(Customer\_\_IDX) - klucz obcy odwołujący się do tabeli User, wartość nie może być pusta.
3. City\_City\_ID - ForeignKey(Customer\_City\_FK), NOT NULL, Index(Customer\_\_IDXv1) - klucz obcy odwołujący się do tabeli City, wartość nie może być pusta.



4. RegistrationDate - NOT NULL - wartość nie może być pusta.
5. LicenseImage - NOT NULL - wartość nie może być pusta.
6. PhotoWithLicense - NOT NULL - wartość nie może być pusta.
7. Saldo FLOAT (10) - NOT NULL - wartość nie może być pusta.
8. EmailSendingPermission - NOT NULL - wartość nie może być pusta.

ContactForm:

1. Customer\_Customer\_ID - ForeignKey(ContactForm\_Customer\_FK), NOT NULL - klucz obcy odwołujący się do tabeli Customer, wartość nie może być pusta.
2. Title - NOT NULL - wartość nie może być pusta.
3. Text - NOT NULL - wartość nie może być pusta.
4. IsResolved - NOT NULL - wartość nie może być pusta.

Encja: Car

1. ParkingZone\_ParkingZone\_ID - ForeignKey(Car\_ParkingZone\_FK), NOT NULL - klucz obcy odwołujący się do tabeli ParkingZone, wartość nie może być pusta.
2. CarState\_CarState\_ID - ForeignKey(Car\_CarState\_FK), NOT NULL - klucz obcy odwołujący się do tabeli CarState, wartość nie może być pusta.
3. Car\_ID - PrimaryKey(Car\_ID), NOT NULL - klucz główny, wartość nie może być pusta.
4. RegistrationNumber - UNIQUE(Car\_RegistrationNumber\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna.
5. VIN\_Code - UNIQUE(Car\_VIN\_Code\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna.
6. Color\_Color\_ID - ForeignKey(Car\_Color\_FK ), NOT NULL, Index(Car\_\_IDXv1) - klucz obcy odwołujący się do tabeli Color, wartość nie może być pusta, indeks.
7. CarModel\_CarModel\_ID - ForeignKey(Car\_CarModel\_FK), NOT NULL - klucz obcy odwołujący się do tabeli CarModel, wartość nie może być pusta, indeks.
8. ProductionYear - NOT NULL - wartość nie może być pusta.
9. InsuranceNumber - NOT NULL - wartość nie może być pusta.
10. IFuelLevel - NOT NULL - wartość nie może być pusta.
11. ILastServiceDate
12. Mileage - NOT NULL - wartość nie może być pusta.
13. CurrentLongitude - NOT NULL - wartość nie może być pusta.
14. CurrentLatitude - NOT NULL - wartość nie może być pusta.
15. CarState\_ID - NOT NULL, Index(Car\_\_IDX) - wartość nie może być pusta, indeks.

Encja: Color

1. Color\_ID - PrimaryKey(Color\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. ColorName - UNIQUE(ColorName\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna w całej tabeli.



Encja: CarModel

1. CarModel\_ID - PrimaryKey(CarModel\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. ModelName - NOT NULL - wartość nie może być pusta.
3. EngineDisplacement - NOT NULL - wartość nie może być pusta.
4. Length - NOT NULL - wartość nie może być pusta.
5. Width - NOT NULL - wartość nie może być pusta.
6. Height - NOT NULL - wartość nie może być pusta.
7. Weight - NOT NULL - wartość nie może być pusta.
8. FuelTankCapacity - NOT NULL - wartość nie może być pusta.
9. TopSpeed - NOT NULL - wartość nie może być pusta.
10. AccelerationTime - NOT NULL - wartość nie może być pusta.
11. FuelType\_FuelType\_ID - ForeignKey(CarModel\_FuelType\_FK), NOT NULL - klucz obcy odwołujący się do tabeli FuelType, wartość nie może być pusta.
12. CarBodyType\_CarBodyType\_ID - ForeignKey(CarModel\_CarBodyType\_FK), NOT NULL - klucz obcy odwołujący się do tabeli CarBodyType, wartość nie może być pusta.
13. FuelEfficiency - NOT NULL - wartość nie może być pusta.
14. PassengerCapacity - NOT NULL - wartość nie może być pusta.
15. CargoCapacity - NOT NULL - wartość nie może być pusta.
16. HorsePower - NOT NULL - wartość nie może być pusta.
17. Emissions - NOT NULL - wartość nie może być pusta.
18. GearboxType\_GearboxType\_ID - ForeignKey(CarModel\_GearboxType\_FK), NOT NULL - klucz obcy odwołujący się do tabeli GearBoxType, wartość nie może być pusta.
19. NumberOfDoors - NOT NULL - wartość nie może być pusta.
20. NumberOfSeats - NOT NULL - wartość nie może być pusta.
21. Tariff - NOT NULL - wartość nie może być pusta.
22. CarPhoto - NOT NULL - wartość nie może być pusta.
23. CarBrand\_CarBrand\_ID - ForeignKey(CarModel\_CarBrand\_FK), NOT NULL - klucz obcy odwołujący się do tabeli CarBrand, wartość nie może być pusta.

Encja: CarState

1. CarState\_ID - PrimaryKey(CarModel\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. CarStateDescription - NOT NULL - wartość nie może być pusta.
3. IsAvailable - NOT NULL - wartość nie może być pusta.



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz Społeczny



Encja: CarBodyType

1. CarBodyType\_ID - PrimaryKey(CarBodyType\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. CategoryName - UNIQUE(CarBodyType\_CategoryName\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna w całej tabeli.

Encja: CarBrand

3. CarBrand\_ID - PrimaryKey(CarBrand\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
4. CarBrandName - UNIQUE(CarBrand\_CarBrandName\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna w całej tabeli.

Encja: GearboxType

1. GearboxType\_ID - PrimaryKey(GearboxType\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. GearboxTypeName - UNIQUE(GearBoxType\_GearboxTypeName\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna w całej tabeli.

Encja: FuelType

1. FuelType\_ID - PrimaryKey(FuelType\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. FuelTypeName - UNIQUE(FuelTypeName\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna w całej tabeli.

Encja: Country

1. Country\_ID - PrimaryKey(Country\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. CountryName - UNIQUE(Country\_CountryName\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna w całej tabeli.
3. CountryCode - UNIQUE(Country\_CountryCode\_UN), NOT NULL - wartość nie może być pusta i musi być unikalna w całej tabeli.

Encja: City

1. City\_ID - PrimaryKey(City\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. CityName - NOT NULL - wartość nie może być pusta.
3. Country\_ID - ForeignKey(City\_Country\_FK), NOT NULL - klucz obcy odwołujący się do tabeli Country, wartość nie może być pusta.



Encja: ParkingZone

1. ParkingZone\_ID - PrimaryKey(ParkingZone\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. Polygon\_Polygon\_ID - ForeignKey(ParkingZone\_Polygon\_FK), NOT NULL, Index(ParkingZone\_\_IDX) - klucz obcy odwołujący się do tabeli Polygon, wartość nie może być pusta, indeks.
3. City\_City\_ID - ForeignKey(ParkingZone\_City\_FK), NOT NULL - klucz obcy odwołujący się do tabeli City, wartość nie może być pusta.
4. AvailableCarsQuantity - NOT NULL - wartość nie może być pusta.

Encja: Polygon

1. Polygon\_ID - PrimaryKey(Polygon\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. Longitude - NOT NULL - wartość nie może być pusta.
3. Latitude - NOT NULL - wartość nie może być pusta.
4. Polygon\_Polygon\_ID - ForeignKey(Polygon\_Polygon\_FK), NOT NULL, Index(Polygon\_\_IDX) - klucz obcy odwołujący się do tabeli Polygon, wartość nie może być pusta, indeks.

Encja: Reservation

1. Customer\_Customer\_ID - ForeignKey(Reservation\_Customer\_FK), NOT NULL - klucz obcy odwołujący się do tabeli Customer, wartość nie może być pusta.
2. ParkingZone\_ParkingZone\_ID - ForeignKey(Reservation\_ParkingZone\_FK), NOT NULL - klucz obcy odwołujący się do tabeli ParkingZone, wartość nie może być pusta.
3. Reservation\_ID - PrimaryKey(Reservation\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
4. ReservationStartDateTime - NOT NULL - wartość nie może być pusta.
5. Car\_Car\_ID - ForeignKey(Reservation\_Car\_FK), NOT NULL - klucz obcy odwołujący się do tabeli Car, wartość nie może być pusta.
6. Transaction\_TransactionID - ForeignKey(Reservation\_Transaction\_FK), Index(Reservation\_\_IDX) - klucz obcy odwołujący się do tabeli Transaction.





Encja: InitialStateReport

1. InitialStateReport\_ID - PrimaryKey(InitialStateReport\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. BodyOk - NOT NULL - wartość nie może być pusta.
3. BodyImage
4. GlassOk - NOT NULL - wartość nie może być pusta.
5. GlassImage
6. TiresOk - NOT NULL - wartość nie może być pusta.
7. TiresImage
8. LightsOk - NOT NULL - wartość nie może być pusta.
9. LightsImage
10. CarMirrorsOk - NOT NULL - wartość nie może być pusta.
11. CarMirrorsImage
12. InteriorOk - NOT NULL - wartość nie może być pusta.
13. InteriorImage
14. CanStart - NOT NULL - wartość nie może być pusta.
15. Reservation\_Reservation\_ID - ForeignKey(InitialStateReport\_Reservation\_FK), NOT NULL, Index(InitialStateReport\_IDX) - klucz obcy odwołujący się do tabeli Reservation, wartość nie może być pusta, indeks.

Encja: Rental

1. Rental\_ID - PrimaryKey(Rental\_Pk), NOT NULL - klucz główny, wartość nie może być pusta.
2. RentalDateTime - NOT NULL - wartość nie może być pusta.
3. ReturnDateTime - NOT NULL - wartość nie może być pusta.
4. Transaction\_TransactionID - ForeignKey(Rental\_Transaction\_FK), NOT NULL, Index(Rental\_IDXv1) - klucz obcy odwołujący się do tabeli Transaction, wartość nie może być pusta, indeks.
5. InitialStateReport\_InitialStateReport\_ID - ForeignKey(Rental\_InitialStateReport\_FK), NOT NULL, Index(Rental\_IDX) - klucz obcy odwołujący się do tabeli InitialStateReport, wartość nie może być pusta, indeks.
6. CarParkedPhotoental - NOT NULL - wartość nie może być pusta.

Encja: Accident

1. Accident\_ID - PrimaryKey(Accident\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. Rental\_Rental\_ID - ForeignKey(Accident\_Rental\_FK), NOT NULL, Index(Accident\_IDX) - klucz obcy odwołujący się do tabeli Rental, wartość nie może być pusta, indeks.
3. AccidentDescription - NOT NULL - wartość nie może być pusta.
4. AccidentDateTime - NOT NULL - wartość nie może być pusta.
5. IsResolved - NOT NULL - wartość nie może być pusta.





Encja: OpinionOnRide

1. OpinionOnRide\_ID - PrimaryKey(OpinionOnRide\_ID), NOT NULL - klucz główny, wartość nie może być pusta.
2. Rental\_Rental\_ID - ForeignKey(OpinionOnRide\_Rental\_FK), NOT NULL, Index(OpinionOnRide\_IDX) - klucz obcy odwołujący się do tabeli Rental, wartość nie może być pusta, indeks.
3. OverallRating - NOT NULL - wartość nie może być pusta.
4. Comment

Encja: CustomerPaymentCard

1. Customer\_Customer\_ID - ForeignKey(CustomerPaymentCard\_Customer\_FK), NOT NULL - klucz obcy odwołujący się do tabeli Customer, wartość nie może być pusta.
2. CustomerPaymentCard\_ID - PrimaryKey(CustomerPaymentCard\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
3. CardNumber - NOT NULL - wartość nie może być pusta.
4. ExpirationDate - NOT NULL - wartość nie może być pusta.
5. CVVCode - NOT NULL - wartość nie może być pusta.
6. BillingAddress - NOT NULL - wartość nie może być pusta.
7. BillingPhoneNumber
8. BillingEmail - NOT NULL - wartość nie może być pusta.
9. IsPreferredCard - NOT NULL - wartość nie może być pusta.
10. IsActive - NOT NULL - wartość nie może być pusta.

Encja: Transaction

1. TransactionID - PrimaryKey(Transaction\_PK), NOT NULL - klucz główny, wartość nie może być pusta.
2. TransactionDateTime - NOT NULL - wartość nie może być pusta.
3. PaymentCost - NOT NULL - wartość nie może być pusta.
4. TransactionStatus - NOT NULL - wartość nie może być pusta.
5. CustomerPaymentCard\_CustomerPaymentCard\_ID - ForeignKey(Transaction\_CustomerPaymentCardr\_FK), NOT NULL, Index(Transaction\_IDX) - klucz obcy odwołujący się do tabeli CustomerPaymentCard, wartość nie może być pusta.
6. Reservation\_Reservation\_ID - ForeignKey(Transaction\_Reservation\_FK), Index(Transaction\_IDXv1) - klucz obcy odwołujący się do tabeli Reservation, indeks.



## **ROZDZIAŁ 7. KOD SQL – TWORZENIE BAZY DANYCH**

Przedstaw instrukcje do tworzenia poszczególnych tabel bazy danych.

Przedstaw instrukcje do tworzenia więzów integralności zarówno na poziomie poszczególnych kolumn jak i całych tabel.

Przedstaw instrukcje do tworzenia związków między tabelami.

Przedstaw instrukcje do tworzenia indeksów dla poszczególnych tabel.

-- Generated by Oracle SQL Developer Data Modeler 23.1.0.087.0806

-- at: 2023-12-28 10:11:23 CET

-- site: Oracle Database 11g

-- type: Oracle Database 11g

-- predefined type, no DDL - MDSYS.SDO\_GEOMETRY

-- predefined type, no DDL - XMLTYPE

```
CREATE TABLE accident (  
    rental_rental_id NUMBER(20) NOT NULL,  
    accident_id      NUMBER(15) NOT NULL,  
    accidentdescription VARCHAR2(100 CHAR) NOT NULL,  
    accidentdatetime DATE NOT NULL,  
    isresolved       CHAR(1) NOT NULL  
);
```

```
CREATE UNIQUE INDEX accident__idx ON  
    accident (  
        rental_rental_id  
    ASC );
```

```
ALTER TABLE accident ADD CONSTRAINT accident_pk PRIMARY KEY (  
    accident_id);
```

```
CREATE TABLE car (  
    parkingzone_parkingzone_id NUMBER(7) NOT NULL,  
    carstate_carstate_id      NUMBER(1) NOT NULL,  
    car_id                     NUMBER(6) NOT NULL,  
    registrationnumber         VARCHAR2(7) NOT NULL,  
    vin_code                   VARCHAR2(20) NOT NULL,  
    color_color_id             NUMBER(4) NOT NULL,  
    carmodel_carmodel_id      NUMBER(8) NOT NULL,  
    productionyear             DATE NOT NULL,  
    insurancenummer            NUMBER(20) NOT NULL,  
    fuellevel                   FLOAT(5) NOT NULL,
```



```
lastservicedate    DATE,  
mileage            NUMBER(7) NOT NULL,  
currentlongitude    NUMBER(9, 6) NOT NULL,  
currentlatitude     NUMBER(8, 6) NOT NULL,  
carstate_id        NUMBER NOT NULL  
);
```

```
CREATE UNIQUE INDEX car__idx ON  
  car (  
    carstate_carstate_id  
  ASC );
```

```
CREATE UNIQUE INDEX car__idxv1 ON  
  car (  
    color_color_id  
  ASC );
```

```
ALTER TABLE car ADD CONSTRAINT car_pk PRIMARY KEY ( car_id );
```

```
ALTER TABLE car ADD CONSTRAINT car_vin_code_un UNIQUE ( vin_code );
```

```
ALTER TABLE car ADD CONSTRAINT car_registrationnumber_un UNIQUE (  
registrationnumber );
```

```
CREATE TABLE carbodytype (  
  carbodytype_id NUMBER(2) NOT NULL,  
  categoryname VARCHAR2(15 CHAR) NOT NULL  
);
```

```
ALTER TABLE carbodytype ADD CONSTRAINT carbodytype_pk PRIMARY KEY (  
carbodytype_id );
```

```
ALTER TABLE carbodytype ADD CONSTRAINT carbodytype_categoryname_un  
UNIQUE ( categoryname );
```

```
CREATE TABLE carbrand (  
  carbrand_id NUMBER(3) NOT NULL,  
  carbrandname VARCHAR2(15) NOT NULL  
);
```

```
ALTER TABLE carbrand ADD CONSTRAINT carbrand_pk PRIMARY KEY (  
carbrand_id );
```

```
ALTER TABLE carbrand ADD CONSTRAINT carbrand_carbrandname_un UNIQUE (  
carbrandname );
```

```
CREATE TABLE carmodel (  
  carmodel_id NUMBER(8) NOT NULL,
```



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz Społeczny



```
modelname          VARCHAR2(30) NOT NULL,  
enginedisplacement  FLOAT(5) NOT NULL,  
length             FLOAT(5) NOT NULL,  
width              FLOAT(5) NOT NULL,  
height             FLOAT(5) NOT NULL,  
weight             NUMBER(5) NOT NULL,  
fueltankcapacity    FLOAT(5) NOT NULL,  
topspeed           FLOAT(5) NOT NULL,  
accelerationtime    FLOAT(5) NOT NULL,  
fueltype_fueltype_id  NUMBER(1) NOT NULL,  
carbodtype_carbodtype_id NUMBER(2) NOT NULL,  
fuelefficiency      FLOAT(5) NOT NULL,  
passengercapacity    NUMBER(2) NOT NULL,  
cargocapacity        FLOAT(5) NOT NULL,  
horsepower          NUMBER(3) NOT NULL,  
emissions           FLOAT(5),  
gearboxtype_gearboxtype_id NUMBER(1) NOT NULL,  
numberofdoors        NUMBER(2) NOT NULL,  
numberofseats        NUMBER(2) NOT NULL,  
tariff              FLOAT(4) NOT NULL,  
carphoto            BLOB NOT NULL,  
carbrand_carbrand_id  NUMBER(3) NOT NULL  
);
```

```
ALTER TABLE carmodel ADD CONSTRAINT carmodel_pk PRIMARY KEY (  
carmodel_id );
```

```
CREATE TABLE carservice (  
  car_car_id          NUMBER(6) NOT NULL,  
  serviceagreement_serviceagreement_id NUMBER(9) NOT NULL,  
  carservice_id        NUMBER(9) NOT NULL,  
  servicedatetime      DATE NOT NULL,  
  servicecost          NUMBER(6) NOT NULL,  
  offeredservice_offeredservice_id  NUMBER(5) NOT NULL  
);
```

```
CREATE UNIQUE INDEX carservice__idx ON  
  carservice (  
    serviceagreement_serviceagreement_id  
  ASC );
```

```
CREATE UNIQUE INDEX carservice__idxv1 ON  
  carservice (  
    offeredservice_offeredservice_id  
  ASC );
```

```
ALTER TABLE carservice ADD CONSTRAINT carservice_pk PRIMARY KEY (  
carservice_id );
```



```
CREATE TABLE carstate (  
    carstate_id      NUMBER(1) NOT NULL,  
    carstatedescription VARCHAR2(100) NOT NULL,  
    isavailable      CHAR(1) NOT NULL  
);
```

```
ALTER TABLE carstate ADD CONSTRAINT carstate_pk PRIMARY KEY ( carstate_id  
);
```

```
CREATE TABLE city (  
    city_id          NUMBER(5) NOT NULL,  
    cityname         VARCHAR2(20 CHAR) NOT NULL,  
    country_country_id NUMBER(3) NOT NULL  
);
```

```
ALTER TABLE city ADD CONSTRAINT city_pk PRIMARY KEY ( city_id );
```

```
CREATE TABLE color (  
    color_id NUMBER(4) NOT NULL,  
    colormame VARCHAR2(15 CHAR) NOT NULL  
);
```

```
ALTER TABLE color ADD CONSTRAINT color_pk PRIMARY KEY ( color_id );
```

```
ALTER TABLE color ADD CONSTRAINT color_colormame_un UNIQUE ( colormame );
```

```
CREATE TABLE contactform (  
    title          VARCHAR2(20) NOT NULL,  
    text           VARCHAR2(200) NOT NULL,  
    customer_customer_id NUMBER(12) NOT NULL,  
    isresolved     CHAR(1) NOT NULL  
);
```

```
CREATE TABLE country (  
    country_id NUMBER(3) NOT NULL,  
    countrycode NUMBER(3) NOT NULL,  
    countryname VARCHAR2(20 CHAR) NOT NULL  
);
```

```
ALTER TABLE country ADD CONSTRAINT country_pk PRIMARY KEY ( country_id  
);
```

```
ALTER TABLE country ADD CONSTRAINT country_countryname_un UNIQUE (  
countryname );
```

```
ALTER TABLE country ADD CONSTRAINT country_countrycode_un UNIQUE (  
countrycode );
```



```
CREATE TABLE customer (  
  customer_id      NUMBER(10) NOT NULL,  
  user_user_id     NUMBER(12) NOT NULL,  
  registrationdate  DATE NOT NULL,  
  licenseimage     BLOB NOT NULL,  
  city_city_id     NUMBER(5) NOT NULL,  
  photowithlicense BLOB NOT NULL,  
  saldo            FLOAT(10) NOT NULL,  
  emailsendingpermission CHAR(1) NOT NULL  
);
```

```
CREATE UNIQUE INDEX customer__idx ON  
  customer (  
    user_user_id  
  ASC );
```

```
CREATE UNIQUE INDEX customer__idxv1 ON  
  customer (  
    city_city_id  
  ASC );
```

```
ALTER TABLE customer ADD CONSTRAINT customer_pk PRIMARY KEY (  
customer_id );
```

```
CREATE TABLE customerpaymentcard (  
  customer_customer_id NUMBER(10) NOT NULL,  
  customerpaymentcard_id NUMBER(12) NOT NULL,  
  cardnumber           NUMBER(16) NOT NULL,  
  expirationdate       DATE NOT NULL,  
  cvvcode              NUMBER(3) NOT NULL,  
  billingaddress       VARCHAR2(20) NOT NULL,  
  billingphonenumber   VARCHAR2(12 CHAR),  
  billingemail         VARCHAR2(30 CHAR) NOT NULL,  
  isprefferedcard     CHAR(1) NOT NULL,  
  isactive             CHAR(1) NOT NULL  
);
```

```
ALTER TABLE customerpaymentcard ADD CONSTRAINT customerpaymentcard_pk  
PRIMARY KEY ( customerpaymentcard_id );
```

```
CREATE TABLE employee (  
  user_user_id NUMBER(12) NOT NULL,  
  employee_id  NUMBER(5) NOT NULL,  
  employmentdate DATE NOT NULL,  
  salary       NUMBER(10) NOT NULL,  
  educationlevel VARCHAR2(50) NOT NULL,  
  profileimage  BLOB NOT NULL,
```



```
role_role_id NUMBER NOT NULL  
);
```

```
CREATE UNIQUE INDEX employee__idx ON  
employee (  
    user_user_id  
    ASC );
```

```
ALTER TABLE employee ADD CONSTRAINT employee_pk PRIMARY KEY (  
employee_id);
```

```
CREATE TABLE fueltype (  
    fueltype_id NUMBER(1) NOT NULL,  
    fueltypename VARCHAR2(15) NOT NULL  
);
```

```
ALTER TABLE fueltype ADD CONSTRAINT fueltype_pk PRIMARY KEY ( fueltype_id  
);
```

```
ALTER TABLE fueltype ADD CONSTRAINT fueltype_fueltypename_un UNIQUE (  
fueltypename );
```

```
CREATE TABLE gearboxtype (  
    gearboxtype_id NUMBER(1) NOT NULL,  
    gearboxtypename VARCHAR2(15) NOT NULL  
);
```

```
ALTER TABLE gearboxtype ADD CONSTRAINT gearboxtype_pk PRIMARY KEY (  
gearboxtype_id );
```

```
ALTER TABLE gearboxtype ADD CONSTRAINT gearboxtype_gearboxtype_id_un  
UNIQUE ( gearboxtype_id );
```

```
ALTER TABLE gearboxtype ADD CONSTRAINT gearboxtype_gearboxtypename_un  
UNIQUE ( gearboxtypename );
```

```
CREATE TABLE initialstatereport (  
    initialstatereport_id NUMBER(20) NOT NULL,  
    bodyok CHAR(1) NOT NULL,  
    bodyimage BLOB,  
    glassok CHAR(1) NOT NULL,  
    glassimage BLOB,  
    tiresok CHAR(1) NOT NULL,  
    tiresimage BLOB,  
    lightsok CHAR(1) NOT NULL,  
    lightsimage BLOB,  
    carmirrorsok CHAR(1) NOT NULL,  
    carmirrorsimage BLOB,
```



```
interiorok          CHAR(1) NOT NULL,  
interiorimage       BLOB,  
canstart            CHAR(1) NOT NULL,  
reservation_reservation_id NUMBER(20) NOT NULL  
);
```

```
CREATE UNIQUE INDEX initialstatereport__idx ON  
initialstatereport (  
    reservation_reservation_id  
ASC );
```

```
ALTER TABLE initialstatereport ADD CONSTRAINT initialstatereport_pk PRIMARY  
KEY ( initialstatereport_id );
```

```
CREATE TABLE offeredservice (  
    servicefacility_servicefacility_id NUMBER(7) NOT NULL,  
    offeredservice_id                NUMBER(5) NOT NULL,  
    serviceprice                     NUMBER(6) NOT NULL,  
    service_service_id               NUMBER(4) NOT NULL,  
    servicename                      VARCHAR2(20) NOT NULL  
);
```

```
ALTER TABLE offeredservice ADD CONSTRAINT offeredservice_pk PRIMARY KEY (  
offeredservice_id );
```

```
CREATE TABLE opiniononride (  
    rental_rental_id NUMBER(20) NOT NULL,  
    opiniononride_id NUMBER(20) NOT NULL,  
    overallrating    NUMBER(2) NOT NULL,  
    "Comment"        VARCHAR2(100 CHAR)  
);
```

```
CREATE UNIQUE INDEX opiniononride__idx ON  
opiniononride (  
    rental_rental_id  
ASC );
```

```
ALTER TABLE opiniononride ADD CONSTRAINT opiniononride_pk PRIMARY KEY (  
opiniononride_id );
```

```
CREATE TABLE owner (  
    owner_id    NUMBER(2) NOT NULL,  
    user_user_id NUMBER(12) NOT NULL,  
    profileimage BLOB NOT NULL  
);
```

```
CREATE UNIQUE INDEX owner__idx ON  
owner (  
    owner_id  
ASC );
```





```
user_user_id  
ASC );
```

```
ALTER TABLE owner ADD CONSTRAINT owner_pk PRIMARY KEY ( owner_id );
```

```
CREATE TABLE parkingzone (  
    city_city_id      NUMBER(7) NOT NULL,  
    parkingzone_id    NUMBER(7) NOT NULL,  
    availablecarsquantity NUMBER(10) NOT NULL,  
    polygon_polygon_id NUMBER(10) NOT NULL  
);
```

```
CREATE UNIQUE INDEX parkingzone__idx ON  
    parkingzone (  
        polygon_polygon_id  
    ASC );
```

```
ALTER TABLE parkingzone ADD CONSTRAINT parkingzone_pk PRIMARY KEY (  
    parkingzone_id );
```

```
CREATE TABLE polygon (  
    polygon_id      NUMBER(20) NOT NULL,  
    longitude       NUMBER(9, 6) NOT NULL,  
    latitude        NUMBER(8, 6) NOT NULL,  
    polygon_polygon_id NUMBER(10) NOT NULL  
);
```

```
CREATE UNIQUE INDEX polygon__idx ON  
    polygon (  
        polygon_polygon_id  
    ASC );
```

```
ALTER TABLE polygon ADD CONSTRAINT polygon_pk PRIMARY KEY ( polygon_id  
);
```

```
CREATE TABLE rental (  
    rental_id          NUMBER(20) NOT NULL,  
    rentaldatetime     DATE NOT NULL,  
    returndatetime     DATE NOT NULL,  
    transaction_transactionid NUMBER(20) NOT NULL,  
    initialstatereport_initialstatereport_id NUMBER(20) NOT NULL,  
    carparkedphoto     BLOB NOT NULL  
);
```

```
CREATE UNIQUE INDEX rental__idx ON  
    rental (  
        initialstatereport_initialstatereport_id  
    ASC );
```



```
CREATE UNIQUE INDEX rental__idxv1 ON  
  rental (  
    transaction_transactionid  
  ASC );
```

```
ALTER TABLE rental ADD CONSTRAINT rental_pk PRIMARY KEY ( rental_id );
```

```
CREATE TABLE reservation (  
  customer_customer_id    NUMBER(10) NOT NULL,  
  parkingzone_parkingzone_id NUMBER(7) NOT NULL,  
  reservation_id          NUMBER(20) NOT NULL,  
  reservationstartdatetime DATE NOT NULL,  
  car_car_id              NUMBER(6) NOT NULL,  
  transaction_transactionid NUMBER(20)  
);
```

```
CREATE UNIQUE INDEX reservation__idx ON  
  reservation (  
    transaction_transactionid  
  ASC );
```

```
ALTER TABLE reservation ADD CONSTRAINT reservation_pk PRIMARY KEY (  
reservation_id );
```

```
CREATE TABLE service (  
  service_id    NUMBER(4) NOT NULL,  
  servicename   VARCHAR2(20 CHAR) NOT NULL,  
  servicedescription VARCHAR2(150 CHAR) NOT NULL  
);
```

```
ALTER TABLE service ADD CONSTRAINT service_pkv2 PRIMARY KEY ( service_id  
);
```

```
ALTER TABLE service ADD CONSTRAINT service_pk UNIQUE ( service_id );
```

```
CREATE TABLE serviceagreement (  
  serviceagreement_id    NUMBER(9) NOT NULL,  
  termsandconditions     VARCHAR2(500 CHAR) NOT NULL,  
  carservice_carservice_id NUMBER(9) NOT NULL  
);
```

```
CREATE UNIQUE INDEX serviceagreement__idx ON  
  serviceagreement (  
    carservice_carservice_id  
  ASC );
```



```
ALTER TABLE serviceagreement ADD CONSTRAINT serviceagreement_pk PRIMARY  
KEY ( serviceagreement_id );
```

```
CREATE TABLE servicecompany (  
    servicecompany_id      NUMBER(4) NOT NULL,  
    servicecompanyname     VARCHAR2(20 CHAR) NOT NULL,  
    servicecompanyphonenumber VARCHAR2(30 CHAR) NOT NULL,  
    servicecompanyaddress  VARCHAR2(20 CHAR) NOT NULL,  
    servicecompanyemail    VARCHAR2(20 CHAR) NOT NULL  
);
```

```
ALTER TABLE servicecompany ADD CONSTRAINT servicecompany_pk PRIMARY  
KEY ( servicecompany_id );
```

```
ALTER      TABLE      servicecompany      ADD      CONSTRAINT  
servicecompany_servicecompanyname_un UNIQUE ( servicecompanyname );
```

```
CREATE TABLE servicefacility (  
    city_city_id          NUMBER(5) NOT NULL,  
    servicecompany_servicecompany_id NUMBER(4) NOT NULL,  
    servicefacility_id     NUMBER(7) NOT NULL,  
    servicefacilityname     VARCHAR2(20 CHAR) NOT NULL,  
    servicefacilityphonenumber VARCHAR2(20 CHAR) NOT NULL  
);
```

```
ALTER TABLE servicefacility ADD CONSTRAINT servicefacility_pk PRIMARY KEY (  
servicefacility_id );
```

```
ALTER      TABLE      servicefacility      ADD      CONSTRAINT  
servicefacility_servicefacilityname_un UNIQUE ( servicefacilityname );
```

```
CREATE TABLE transaction (  
    transactionid          NUMBER(20) NOT NULL,  
    transactiondatetime    DATE NOT NULL,  
    paymentcost            NUMBER(10) NOT NULL,  
    transactionstatus      VARCHAR2(10 CHAR) NOT NULL,  
    customerpaymentcard_customerpaymentcard_id NUMBER(20) NOT NULL,  
    reservation_reservation_id NUMBER(20)  
);
```

```
CREATE UNIQUE INDEX transaction__idx ON  
transaction (  
    customerpaymentcard_customerpaymentcard_id  
ASC );
```

```
CREATE UNIQUE INDEX transaction__idxv1 ON
```



```
transaction (  
    reservation_reservation_id  
ASC );
```

```
ALTER TABLE transaction ADD CONSTRAINT transaction_pk PRIMARY KEY (  
transactionid );
```

```
CREATE TABLE "User" (  
    user_id    NUMBER(12) NOT NULL,  
    email      VARCHAR2(30 CHAR) NOT NULL,  
    phonenumber VARCHAR2(20) NOT NULL,  
    firstname  VARCHAR2(20 CHAR) NOT NULL,  
    lastname   VARCHAR2(20 CHAR) NOT NULL,  
    password   VARCHAR2(40 CHAR) NOT NULL,  
    gender     VARCHAR2(10) NOT NULL,  
    passportimage BLOB NOT NULL,  
    birthdate  DATE NOT NULL  
);
```

```
ALTER TABLE "User" ADD CONSTRAINT user_pk PRIMARY KEY ( user_id );
```

```
ALTER TABLE "User" ADD CONSTRAINT user_email_un UNIQUE ( email );
```

```
ALTER TABLE "User" ADD CONSTRAINT user_phonenumber_un UNIQUE (  
phonenumber );
```

```
ALTER TABLE accident  
    ADD CONSTRAINT accident_rental_fk FOREIGN KEY ( rental_rental_id )  
    REFERENCES rental ( rental_id );
```

```
ALTER TABLE car  
    ADD CONSTRAINT car_carmodel_fk FOREIGN KEY ( carmodel_carmodel_id )  
    REFERENCES carmodel ( carmodel_id );
```

```
ALTER TABLE car  
    ADD CONSTRAINT car_carstate_fk FOREIGN KEY ( carstate_carstate_id )  
    REFERENCES carstate ( carstate_id );
```

```
ALTER TABLE car  
    ADD CONSTRAINT car_color_fk FOREIGN KEY ( color_color_id )  
    REFERENCES color ( color_id );
```

```
ALTER TABLE car  
    ADD CONSTRAINT car_parkingzone_fk FOREIGN KEY (  
parkingzone_parkingzone_id )  
    REFERENCES parkingzone ( parkingzone_id );
```

```
ALTER TABLE carmodel
```



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz Społeczny



```
ADD CONSTRAINT carmodel_carbodytype_fk FOREIGN KEY (
carbodytype_carbodytype_id )
REFERENCES carbodytype ( carbodytype_id );
```

```
ALTER TABLE carmodel
ADD CONSTRAINT carmodel_carbrand_fk FOREIGN KEY ( carbrand_carbrand_id )
REFERENCES carbrand ( carbrand_id );
```

```
ALTER TABLE carmodel
ADD CONSTRAINT carmodel_fueltype_fk FOREIGN KEY ( fueltype_fueltype_id )
REFERENCES fueltype ( fueltype_id );
```

```
ALTER TABLE carmodel
ADD CONSTRAINT carmodel_gearboxtype_fk FOREIGN KEY (
gearboxtype_gearboxtype_id )
REFERENCES gearboxtype ( gearboxtype_id );
```

```
ALTER TABLE carservice
ADD CONSTRAINT carservice_car_fk FOREIGN KEY ( car_car_id )
REFERENCES car ( car_id );
```

```
ALTER TABLE carservice
ADD CONSTRAINT carservice_offeredservice_fk FOREIGN KEY (
offeredservice_offeredservice_id )
REFERENCES offeredservice ( offeredservice_id );
```

```
ALTER TABLE city
ADD CONSTRAINT city_country_fk FOREIGN KEY ( country_country_id )
REFERENCES country ( country_id );
```

```
ALTER TABLE contactform
ADD CONSTRAINT contactform_customer_fk FOREIGN KEY (
customer_customer_id )
REFERENCES customer ( customer_id );
```

```
ALTER TABLE customer
ADD CONSTRAINT customer_city_fk FOREIGN KEY ( city_city_id )
REFERENCES city ( city_id );
```

```
ALTER TABLE customer
ADD CONSTRAINT customer_user_fk FOREIGN KEY ( user_user_id )
REFERENCES "User" ( user_id );
```

```
ALTER TABLE customerpaymentcard
ADD CONSTRAINT customerpaymentcard_customer_fk FOREIGN KEY (
customer_customer_id )
REFERENCES customer ( customer_id );
```



```
ALTER TABLE employee
```

```
ADD CONSTRAINT employee_user_fk FOREIGN KEY ( user_user_id )  
REFERENCES "User" ( user_id );
```

```
ALTER TABLE initialstatereport
```

```
ADD CONSTRAINT initialstatereport_reservation_fk FOREIGN KEY ( reservation_reservation_id )  
REFERENCES reservation ( reservation_id );
```

```
ALTER TABLE offeredservice
```

```
ADD CONSTRAINT offeredservice_service_fk FOREIGN KEY ( service_service_id )  
REFERENCES service ( service_id );
```

```
ALTER TABLE offeredservice
```

```
ADD CONSTRAINT offeredservice_servicefacility_fk FOREIGN KEY ( servicefacility_servicefacility_id )  
REFERENCES servicefacility ( servicefacility_id );
```

```
ALTER TABLE opiniononride
```

```
ADD CONSTRAINT opiniononride_rental_fk FOREIGN KEY ( rental_rental_id )  
REFERENCES rental ( rental_id );
```

```
ALTER TABLE owner
```

```
ADD CONSTRAINT owner_user_fk FOREIGN KEY ( user_user_id )  
REFERENCES "User" ( user_id );
```

```
ALTER TABLE parkingzone
```

```
ADD CONSTRAINT parkingzone_city_fk FOREIGN KEY ( city_city_id )  
REFERENCES city ( city_id );
```

```
ALTER TABLE parkingzone
```

```
ADD CONSTRAINT parkingzone_polygon_fk FOREIGN KEY ( polygon_polygon_id )  
REFERENCES polygon ( polygon_id );
```

```
ALTER TABLE polygon
```

```
ADD CONSTRAINT polygon_polygon_fk FOREIGN KEY ( polygon_polygon_id )  
REFERENCES polygon ( polygon_id );
```

```
ALTER TABLE rental
```

```
ADD CONSTRAINT rental_initialstatereport_fk FOREIGN KEY ( initialstatereport_initialstatereport_id )  
REFERENCES initialstatereport ( initialstatereport_id );
```

```
ALTER TABLE rental
```

```
ADD CONSTRAINT rental_transaction_fk FOREIGN KEY ( transaction_transactionid )  
REFERENCES transaction ( transactionid );
```



```
ALTER TABLE reservation
```

```
ADD CONSTRAINT reservation_car_fk FOREIGN KEY ( car_car_id )  
REFERENCES car ( car_id );
```

```
ALTER TABLE reservation
```

```
ADD CONSTRAINT reservation_customer_fk FOREIGN KEY (  
customer_customer_id )  
REFERENCES customer ( customer_id );
```

```
ALTER TABLE reservation
```

```
ADD CONSTRAINT reservation_parkingzone_fk FOREIGN KEY (  
parkingzone_parkingzone_id )  
REFERENCES parkingzone ( parkingzone_id );
```

```
ALTER TABLE servicefacility
```

```
ADD CONSTRAINT servicefacility_city_fk FOREIGN KEY ( city_city_id )  
REFERENCES city ( city_id );
```

```
ALTER TABLE servicefacility
```

```
ADD CONSTRAINT servicefacility_servicecompany_fk FOREIGN KEY (  
servicecompany_servicecompany_id )  
REFERENCES servicecompany ( servicecompany_id );
```

```
ALTER TABLE transaction
```

```
ADD CONSTRAINT transaction_customerpaymentcard_fk FOREIGN KEY (  
customerpaymentcard_customerpaymentcard_id )  
REFERENCES customerpaymentcard ( customerpaymentcard_id );
```

-- Oracle SQL Developer Data Modeler Summary Report:

```
--  
-- CREATE TABLE          30  
-- CREATE INDEX           19  
-- ALTER TABLE           77  
-- CREATE VIEW             0  
-- ALTER VIEW             0  
-- CREATE PACKAGE          0  
-- CREATE PACKAGE BODY    0  
-- CREATE PROCEDURE        0  
-- CREATE FUNCTION         0  
-- CREATE TRIGGER          0  
-- ALTER TRIGGER          0  
-- CREATE COLLECTION TYPE  0  
-- CREATE STRUCTURED TYPE  0  
-- CREATE STRUCTURED TYPE BODY 0  
-- CREATE CLUSTER         0
```



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz Społeczny





*Zintegrowany Program Rozwoju Politechniki Lubelskiej – część druga*

```
-- CREATE CONTEXT          0
-- CREATE DATABASE          0
-- CREATE DIMENSION         0
-- CREATE DIRECTORY         0
-- CREATE DISK GROUP        0
-- CREATE ROLLBACK SEGMENT      0
-- CREATE SEQUENCE          0
-- CREATE MATERIALIZED VIEW    0
-- CREATE MATERIALIZED VIEW LOG 0
-- CREATE SYNONYM           0
-- CREATE TABLESPACE        0
-- CREATE USER              0
--
-- DROP TABLESPACE         0
-- DROP DATABASE            0
--
-- REDACTION POLICY         0
--
-- ORDS DROP SCHEMA         0
-- ORDS ENABLE SCHEMA       0
-- ORDS ENABLE OBJECT       0
--
-- ERRORS                   0
-- WARNINGS                 0
```



Raport powstał podczas zajęć laboratoryjnych z przedmiotu  
prowadzonego w ramach projektu  
*„Zintegrowany Program Rozwoju Politechniki Lubelskiej – część  
druga”*,  
umowa nr **POWR.03.05.00-00-Z060/18-00**  
w ramach Programu Operacyjnego Wiedza Edukacja Rozwój  
2014-2020  
współfinansowanego ze środków Europejskiego Funduszu  
Społecznego