Politechnika Warszawska
Cyberbezpieczeństwo

Sprawozdanie Projekt BADA
CZĘŚĆ I

Michał Bogusz 331158
Mateusz Kowalczuk 331172

Listopad 2024

Spis treści

1. Zakres i cel projektu

Celem projektu jest zaprojektowanie i zaimplementowanie relacyjnej bazy danych dla systemu zarządzania salonem samochodowym. System obsługuje procesy związane z działalnością salonów samochodowych, takich jak sprzedaż i wynajem pojazdów, zarządzanie usługami serwisowymi oraz danymi klientów i pracowników. Dodatkowo, baza danych umożliwia przechowywanie informacji o różnych placówkach związanych z działalnością danego salonu (np. warsztaty, punkty handlowe), co pozwala na usprawnienie procesów wewnętrznych i poprawę jakości obsługi klientów.
Projekt ma na celu:

— Zapewnienie kompleksowego przechowywania danych: umożliwienie efektywnego zarządzania informacjami o klientach, pracownikach, pojazdach i usługach.
— Ułatwienie analizy i obsługi danych: poprzez eliminację redundancji, zastosowanie normalizacji oraz stworzenie intuicyjnej struktury logicznej bazy.
— Poprawę integralności danych: wprowadzenie odpowiednich kluczy, reguł integralności i mechanizmów kontroli, aby minimalizować błędy

System jest przeznaczony dla pracowników administracyjnych salonów, pracowników warsztatów oraz personelu serwisowego, a także umożliwia raportowanie dla kadry kierowniczej.

2. Definicja systemu

System zaprojektowany został dla salonu samochodowego zajmującego się sprzedażą, wynajmem i serwisowaniem samochodów. Czynności te salon wykonuje za pośrednictwem placówek - warsztatów i punktów handlowych - w których zatrudniani są pracownicy. Warsztaty są przystosowane do serwisowania danych marek i wykonywane są w nich serwisy, punkty handlowe posiadaja pulę samochodów które wynajmują lub sprzedają.

2.1. Perspektywy użytkowników

System będzie wykorzystywany przez:

— Kierowników: zarządzanie danymi o pracownikach, klientach i pojazdach.
— Sprzedawców: obsługa procesu sprzedaży i wynajmu samochodów.
— Mechaników: przeprowadzanie zleconych serwisów.
— Kierowników warsztatów: planowanie napraw pojazdów.

2.2. zidentyfikowane operacje na danych

System będzie wykonywał następujące operacje na danych: W obszarze pracowników:

— dodawanie usuwanie pracownika
— modyfikacja danych osobowych
— zmiana stanowiska
— zmiana wynagrodzenia
— zmiana przypisanej placówki

W obszarze placówek:
-dodawanie usuwanie placówek -modyfikacja adresu, nazwy -przypisywanie pracowników i samochodów (jeżeli punk sprzedaży) -modyfikacja cech specjalnych placówki

W obszarze samochodów:

— dodawanie usuwanie samochodów
— zmiana przypisanej placówki
— zmiana numeru rejestracyjnego
— zmiana stanu technicznego, przebiegu

W obszarze klientów:

— dodawanie usuwanie,
— zmiana danych
— przypisywanie samochodu
— przypisywanie do placówki

W obszarze serwisów:

— dodawanie usuwanie serwisów
— przypisywanie pracowników
— przypisywanie samochodów i klientów

3. Model konceptualny

3.1. Definicja zbiorów encji określonych w projekcie (decyzje projektowe)

— **Salon Samochodowy**: informacje o salonach, takich jak nazwa, adres, i menedżerowie.
— **Placówka**: ogólna reprezentacja lokalizacji, w której odbywają się różne działania.
— **Pracownik**: dane personalne i zawodowe pracowników w placówkach.
— **Klient**: dane kontaktowe klientów korzystających z usług.
— **Samochód**: szczegółowe informacje o pojazdach i ich stanie technicznym.
— **Serwis**: dane o przeprowadzonych usługach serwisowych.
— **Warsztat i Punkt Handlowy**: specjalistyczne typy placówek.

3.2. Ustalenie związków między encjami i ich typów

Poniżej przedstawiono wszystkie związki między encjami widoczne na schemacie bazy danych:

1. **Jest pracownikiem salonu** Relacja 0:N między encjami **Salon Samochodowy** i **Pracownik**. Pracownik jest przypisany do konkretnego salonu samochodowego. Każdy salon może zatrudniać wielu pracowników.
2. **Pracuje w placówce** Relacja 1:N między encjami **Placówka** i **Pracownik**. Pracownik może pracować w jednej placówce, podczas gdy placówka może zatrudniać wielu pracowników.
3. **Pracownik pracuje przy serwisie** Relacja N:M między encjami **Serwis** i **Pracownik**. Pracownik może być przypisany do wielu serwisów, a serwis może być obsługiwany przez wielu pracowników.
4. **Serwis odbywa się w placówce** Relacja 1:N między encjami **Placówka** i **Serwis**. Serwis jest realizowany w konkretnej placówce. Placówka może obsługiwać wiele serwisów.
5. **Samochód jest serwisowany** Relacja 1:N między encjami **Samochód** i **Serwis**. Każdy serwis obejmuje jeden samochód, ale samochód może być serwisowany wielokrotnie.
6. **Samochód jest wynajęty** Relacja 1:N między encjami **Klient** i **Samochód**. Klient może wynająć wiele samochodów, ale jeden samochód jest wynajmowany przez jednego klienta w danym momencie.
7. **Samochód jest kupiony** Relacja 1:N między encjami **Klient** i **Samochód**. Klient może kupić wiele samochodów, ale każdy samochód jest przypisany do jednego nabywcy.
8. **Korzysta z usług placówki** Relacja N:M między encjami **Klient** i **Placówka**. Klient może korzystać z usług różnych placówek, a placówka obsługuje wielu klientów.
9. **Posiada placówkę** Relacja 1:N między encjami **Salon Samochodowy** i **Placówka**. Salon samochodowy może posiadać wiele placówek, natomiast każda placówka jest przypisana do jednego salonu.
10. **Posiada samochody** Relacja 1:N między encjami **Placówka** i **Samochód**. Placówka może posiadać wiele samochodów na stanie, ale każdy samochód należy do jednej placówki.

3.3. Określenie atrybutów i ich dziedzin

Pracownik

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
ID_Pracownika	Unikatowy identyfikator pracownika	Integer	T	T	T	Klucz główny
Imię	Imię pracownika	Varchar(20)	T	N	T	-
Nazwisko	Nazwisko pracownika	Varchar(20)	T	N	T	-
PESEL	Identyfikator osobowy	Char(11)	N	N	T	-
Numer_Telefonu	Numer telefonu pracownika	Varchar(12)	T	N	T	Zgodny z formatem numerów
Adres	Adres zamieszkania	Varchar(200)	T	N	N	Pole segmentowe
Stanowisko	Stanowisko pracownika	Varchar(20)	T	N	T	-
Wynagrodzenie	Wynagrodzenie pracownika	Decimal	T	N	T	-

Salon Samochodowy

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
ID_Salonu	Unikatowy identyfikator salonu	Integer	T	T	T	Klucz główny
Nazwa	Nazwa salonu	Varchar(50)	T	N	T	-
Adres	Adres salonu	Varchar(200)	T	N	N	Pole segmentowe

Placówka

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
ID_Placówki	Unikatowy identyfikator placówki	Integer	T	T	T	Klucz główny
Nazwa	Nazwa placówki	Varchar(50)	T	N	T	-
Adres	Adres placówki	Varchar(200)	T	N	N	Pole segmentowe

Serwis

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
ID_Serwisu	Unikatowy identyfikator serwisu	Integer	T	T	T	Klucz główny
Data_rozpoczęcia	Data rozpoczęcia serwisu	Date	T	N	T	-
Koszt	Koszt serwisu	Decimal	T	N	T	-
Opis	Opis serwisu	Varchar(500)	T	N	T	-

Klient

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
ID_Klienta	Unikatowy identyfikator klienta	Integer	T	T	T	Klucz główny
Imię	Imię klienta	Varchar(20)	T	N	T	-
Nazwisko	Nazwisko klienta	Varchar(20)	T	N	T	-
Adres	Adres klienta	Varchar(200)	T	N	N	Pole segmentowe
Numer_Telefonu	Numer telefonu klienta	Varchar(12)	T	N	T	-

Samochód

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
ID_Samochodu	Unikatowy identyfikator samochodu	Integer	T	T	T	Klucz główny
Numer_Rejestracyjny	Numer rejestracyjny samochodu	Varchar(10)	T	N	T	-
Model	Model samochodu	Varchar(30)	T	N	T	-
Marka	Marka samochodu	Varchar(30)	T	N	T	-
Data_Produkcji	Data produkcji samochodu	Date	T	N	T	-
Przebieg	Przebieg samochodu	Integer	T	N	T	W kilometrach
Stan_Techniczny	Stan techniczny samochodu	Varchar(50)	T	N	T	-
Czy_Powypadkowy	Informacja o wypadkach	Boolean	T	N	T	-
Cena_Wynajmu	Cena wynajmu samochodu	Decimal	T	N	T	-
Cena_Sprzedaży	Cena sprzedaży samochodu	Decimal	T	N	T	-

Warsztat

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Liczba_Stanowisk	Liczba stanowisk warsztatowych	Integer	T	N	T	-
Serwisowane_Marki	Lista serwisowanych marek	Varchar(200)	T	N	N	Lista rozdzielana przecinkami
Czy_Wymiana_Opon	Czy warsztat oferuje wymianę opon	Boolean	T	N	T	-

Punkt Handlowy

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Czy_Wynajem	Czy punkt oferuje wynajem	Boolean	T	N	T	-
Czy_Sprzedaż	Czy punkt oferuje sprzedaż	Boolean	T	N	T	-

3.4. Dodatkowe reguły integralnościowe (reguły biznesowe)

Aby nasz projekt miał sens biznesowy dodaliśmy kilka reguł:

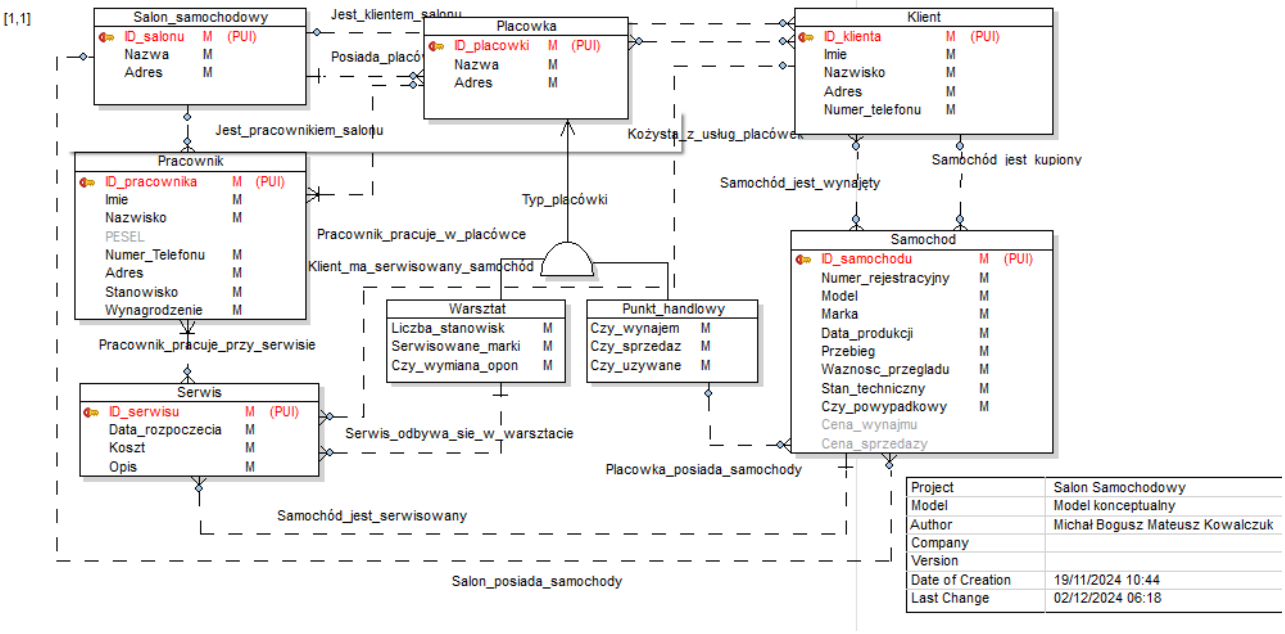
— Każdy serwis dotyczy jednego samochodu,
— Każdy samochód wynajmuje (na raz)/posiada maksymalnie 1 klient
— Każdy samochód znajduje się w maksymalnie jednej placówce.
— Każdy pracownik pracuje w maksymalnie jednej placówce.

3.5. Klucze kandydujące i główne (decyzje projektowe)

Każda Encja dostała atrybut ID_(Nazwa_encji), który jest kluczem głównym tej encji.
Klucze kandydujące to: W encji Samochód kluczem kandydującym może być numer rejestracyjny, dla salonu i placówki kluczami kandydującymi są nazwy.

3.6. Schemat ER na poziomie konceptualnym

Nasz schemat ER na poziomie konceptualnym zrobiliśmy w narzędziu Toad Data Modeler. Wygląda on w ten sposób (powiększona wersja na Rys 3 na końcu dokumentu):



Rys. 1. Schemat konceptualny

3.7. Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

Relacja ma serwisowany samochód rozwiązuje potencjalną pułapkę wachlarzową. Bez niej nie można przypisać klienta do serwisu, ponieważ dany serwis przypisany jest do placówki, która posiada wielu klientów. Potencjalna pułapka wachlarzowa - pracownik - samochód brak logicznego powiązania. Nie możliwym jest przypisanie danego pracownika do danego samochodu, ponieważ pracownik przypisany jest do placówki w której jest wiele samochodów. Relacja salon posiada samochody rozwiązuje potencjalna pułapkę szczelinową. W przypadku usunięcia samochodu z placówki (z przyczyn innych niż zakup), lecz pozostawienia go w salonie, nie możliwym jest powiązanie go z salonem.

4. Model logiczny

Model logiczny opisuje szczegóły implementacyjne bazy danych, bazując na modelu konceptualnym i uwzględniając ograniczenia wynikające z modelu relacyjnego. Jego celem jest przekształcenie modelu konceptualnego w schemat możliwy do zaimplementowania w relacyjnej bazie danych.

4.1. Charakterystyka modelu relacyjnego

Model relacyjny opiera się na relacjach (tabelach), które przechowują dane w postaci wierszy i kolumn. Każda tabela odpowiada jednej encji z modelu konceptualnego, a kolumny reprezentują atrybuty tej encji. Kluczowe cechy modelu relacyjnego w naszym projekcie:

— **Klucze główne (Primary Keys)**: Każda tabela zawiera jedno pole (lub zestaw pól), które jednoznacznie identyfikuje każdy wiersz.
— **Klucze obce (Foreign Keys)**: Relacje między tabelami są definiowane przez klucze obce, które wskazują na klucze główne innych tabel.
— **Integracja danych**: Relacje między tabelami zachowują spójność i eliminują redundancję poprzez proces normalizacji.
— **Nazwy tabel**: Tabelom, które tworzymy z encji z modelu konceptualnego zmieniamy nazwę na liczbę mnogą.

4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym – przykład

W trakcie tworzenia modelu logicznego musieliśmy usunąć elementy, które nie są bezpośrednio zgodne z modelem relacyjnym. Przykłady problemów rozwiązanych podczas transformacji:

— **Relacje wielowartościowe (M:N):**
W modelu konceptualnym relacja między samochodami a klientami była wielowartościowa, ponieważ jeden klient mógł kupić lub wynająć wiele samochodów, a jeden samochód mógł mieć historię różnych klientów. W modelu logicznym wprowadziliśmy tabelę pośrednią `Transakcje`, która zawiera informacje o każdej sprzedaży lub wynajmie samochodu.
W tej tabeli będą przechowywane wszystkie wynajmy i kupna, których podejmowali się klienci. Dzięki temu będziemy mieli całą historię wszystkich transakcji. **Tabela Transakcje:**
— Klucz główny: `ID_Transakcji`
— Klucze obce: `ID_Klienta`, `ID_Samochodu`
— Dodatkowe pola: `Data_transakcji`, `Typ_transakcji` (kupno/wynajem) `Data_końca_wynajmu` (pole w przypadku wynajmu)
Kolejną przykładową relacją wielu do wielu której się pozbyliśmy to relacja pracowników i serwisów - każdy pracownik mógł pracować przy wielu serwisach i każdy serwis mógł być przeprowadzany przez wielu pracowników, więc wprowadzona została tablica Przydziały_serwisowe, która łączy klucze główne pracowników i serwisów - jest to standardowe podejście

4.3. Proces normalizacji – analiza i przykłady

Proces normalizacji służy do eliminowania redundancji danych i unikania anomalii aktualizacyjnych. W naszym projekcie dokonaliśmy następujących etapów normalizacji:

1. **Pierwsza postać normalna (1PN):**
Wszystkie atrybuty w tabelach zostały podzielone na atomowe wartości, co dotyczyło w naszym przypadku adresów, do których stworzona została osobna tabela Adresy, połączona ze wszystkimi tablami w których przed normalizacją znajdowały się pola adresy.
Ponadto w celu wyeliminowania powtarzających się grup, utworzone zostały tablice: Marki i Modele, dzięki którym wyeliminowanie powtarzanie dla każdego samochodu informacji o marce i modelu. Dodatkowo pozwoliło to również na robicie pola wielowartościowego w Warsztacie: Serwisowane_marki. Zostało ono zastąpione relacją wiele do wielu (zapisaną przy pomocy tabeli łączącej Serwisowane_marki) z markami.
Podobnie postąpiono w przypadku wynagrodzenia i stanowiska, co dodatkowo pozwoliło na utworzenie akceptowalnego zbioru stanowisk i zachowywanie historii wypłat.
2. **Druga postać normalna (2PN):**
Po przejściu na 1PN, nasza baza okazała się już być w 2PN, ponieważ kluczami głównymi są atrybuty ID, które są kluczami prostymi.
3. **Trzecia postać normalna (3PN):**
Dzięki wyeliminowaniu powtarzających się grup w 1PN i zastosowaniu kluczy prostych, kazdy atrybut jest zależny tylko od całego klucza, co daje nam 3PN.

Poprzez przejście na 3PN skutecznie znormalizowaliśmy nasz model dzięki czemu zmniejszyliśmy redundancje danych oraz poprawiliśmy możliwości wykonywania założonych operacji na danych unikając anomalii.

4.4. Schemat ER na poziomie modelu logicznego

Powiększony model można znaleźć na Rys. 4 na końcu dokumentu. Tabela: `Salony Samochodowe`

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
`ID_Salonu`	`Integer`	Tak	Nie	Tak	Unikalny identyfikator salonu samochodowego.
`Nazwa`	`Varchar(30)`	Nie	Nie	Tak	Nazwa salonu samochodowego.
`ID_Adresu`	`Integer`	Nie	Tak	Tak	Klucz obcy adres salonu

Tabela: `Placówki`

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_Placowki	Integer	Tak	Nie	Tak	Unikalny identyfikator placówki należącej do salonu.
Nazwa	Varchar(30)	Nie	Nie	Tak	Nazwa placówki.
ID_Salonu	Integer	Nie	Tak	Tak	Powiązanie placówki z salonem, do którego należy.
ID_Adresu	Integer	Nie	Tak	Tak	Klucz obcy adres placówki

Tabela: Pracownicy

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_Pracownika	Integer	Tak	Nie	Tak	Unikalny identyfikator pracownika.
Imie	Varchar(20)	Nie	Nie	Tak	Imię pracownika.
Nazwisko	Varchar(30)	Nie	Nie	Tak	Nazwisko pracownika.
PESEL	Char(11)	Nie	Nie	Nie	Numer PESEL pracownika(może nie występować)
Numer_Telefonu	Varchar(15)	Nie	Nie	Tak	Numer telefonu kontaktowego.
ID_Stanowiska	Integer	Nie	Tak	Nie	Id Stanowiska zajmowanego przez pracownika.
Wynagrodzenie	Integer	Nie	Nie	Tak	Wynagrodzenie pracownika (miesięczne).
ID_Salonu	Integer	Nie	Tak	Tak	Identyfikator salonu, w którym pracuje pracownik.
ID_Placówki	Integer	Nie	Tak	Tak	Identyfikator placówki, w której pracuje pracownik.
ID_Adresu	Integer	Nie	Tak	Tak	Klucz obcy adres pracownika

Tabela: Klienci

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_Klienta	Integer	Tak	Nie	Tak	Unikalny identyfikator klienta.
Imie	Varchar(20)	Nie	Nie	Tak	Imię klienta.
Nazwisko	Varchar(30)	Nie	Nie	Tak	Nazwisko klienta.
Numer_Telefonu	Varchar(15)	Nie	Nie	Tak	Numer kontaktowy klienta.
ID_Salonu	Integer	Nie	Tak	Tak	Identyfikator salonu, z którego usług korzysta klient.
ID_Adresu	Integer	Nie	Tak	Tak	Klucz obcy adres Klienta

Tabela: Transakcje

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_Samochodu	Integer	Tak	Tak	Tak	Identyfikator samochodu będącego przedmiotem transakcji.
ID_Klienta	Integer	Tak	Tak	Tak	Identyfikator klienta dokonującego transakcji.

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
Typ_transakcji	Typ_transakcjiD	Nie	Nie	Tak	Typ transakcji (kupno-/wynajem).
Data_transakcji	Date	Nie	Nie	Tak	Data wykonania transakcji.
Data_końca_wypożyczenia	Date	Nie	Nie	Nie	Data zakończenia wynajmu (jeśli dotyczy).

Tabela: Serwisy

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_serwisu	Integer	Tak	Nie	Tak	Unikalny identyfikator serwisu.
data_rozpoczęcia	Date	Nie	Nie	Tak	Data rozpoczęcia serwisu.
koszt	Number(10,2)	Nie	Nie	Tak	Koszt wykonania serwisu.
Opis	Varchar(1000)	Nie	Nie	Tak	Szczegółowy opis serwisu.
ID_samochodu	Integer	Nie	Tak	Tak	Powiązanie z samochodem, który jest serwisowany.
ID_Placowki	Integer	Nie	Tak	Tak	Powiązanie z placówką, w której odbywa się serwis.
ID_klienta	Integer	Nie	Tak	Tak	Powiązanie z klientem, który zlecił serwis.

Tabela: Samochody

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_Samochodu	Integer	Tak	Nie	Tak	Unikalny identyfikator samochodu.
Numer_rejestracyjny	Varchar(28)	Nie	Nie	Tak	Numer rejestracyjny samochodu.
data_produkcji	Date	Nie	Nie	Tak	Data produkcji samochodu.
przebieg	Integer	Nie	Nie	Tak	Przebieg samochodu w kilometrach.
waznosc_przegladu	Date	Nie	Nie	Tak	Ważność przeglądu technicznego.
stan_techniczny	Stan_technicznyD	Nie	Nie	Tak	Stan techniczny pojazdu.
Czy_powypadkowy	Char(1)	Nie	Nie	Tak	Informacja, czy samochód jest powypadkowy (T/N).
Cena_wynajmu	Number(10,2)	Nie	Nie	Nie	Cena wynajmu
Cena_sprzedazy	Number(10,2)	Nie	Nie	Nie	Cena sprzedazy
ID_Placowki	Integer	Nie	Tak	Tak	Powiązanie z placówką, w której znajduje się samochód.
ID_klienta	Integer	Nie	Tak	Nie	Powiązanie z klientem, który jest właścicielem samochodu.

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_Salonu	Integer	Nie	Tak	Nie	Powiązanie z salonem, który posiada samochód.
ID_Modelu	Integer	Nie	Tak	Tak	Powiązanie z modelem samochodu.

Tabela: Punkty Handlowe

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_Placowki	Integer	Tak	Nie	Tak	Unikalny identyfikator punktu handlowego.
Czy_wynajem	Char(1)	Nie	Nie	Tak	Czy punkt handlowy oferuje wynajem (T/N).
Czy_sprzedaz	Char(1)	Nie	Nie	Tak	Czy punkt handlowy oferuje sprzedaż (T/N).
Czy_Uzywane	Char(1)	Nie	Nie	Tak	Czy punkt handlowy zajmuje się samochodami używanymi (T/N).

Tabela: Warsztaty

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_Placowki	Integer	Tak	Nie	Tak	Unikalny identyfikator warsztatu.
Liczba_stanowisk	Integer	Nie	Nie	Tak	Liczba stanowisk w warsztacie.
Czy_wymiana_opon	Char(1)	Nie	Nie	Tak	Czy warsztat oferuje wymianę opon (T/N).

Tabela: Adresy

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_Adresu	Integer	Tak	Nie	Tak	Unikalny identyfikator adresu.
Miasto	Varchar(30)	Nie	Nie	Tak	Nazwa miasta.
Ulica	Varchar(30)	Nie	Nie	Tak	Nazwa ulicy.
Numer_budynku	Varchar(25)	Nie	Nie	Tak	Numer budynku.
Numer_lokalu	Varchar(25)	Nie	Nie	Nie	Numer lokalu.
Kod_pocztowy	Char(6)	Nie	Nie	Tak	Kod pocztowy.

Tabela: Modele

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_Modelu	Integer	Tak	Nie	Tak	Unikalny identyfikator modelu.
Nazwa_modelu	Varchar(10)	Nie	Nie	Tak	Nazwa modelu samochodu.
ID_marki	Integer	Nie	Tak	Tak	Powiązanie modelu z marką.
ID_Salonu	Integer	Nie	Tak	Tak	Powiązanie modelu z salonem.

Tabela: Marki

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_marki	Integer	Tak	Nie	Tak	Unikalny identyfikator marki samochodów.
Nazwa_marki	Varchar(210)	Nie	Nie	Tak	Nazwa marki samochodów.
ID_Salonu	Integer	Nie	Tak	Tak	Powiązanie marki z salonem.

Tabela: Serwisowane marki

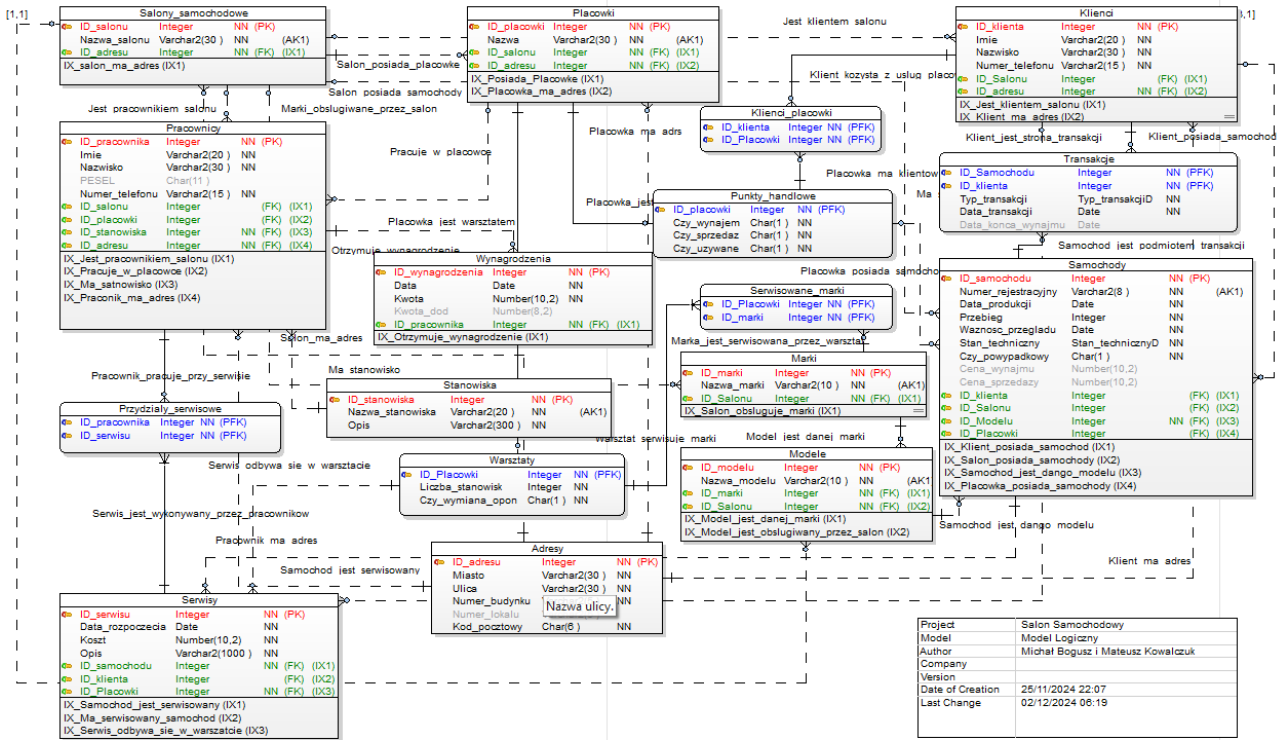
Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_Placowki	Integer	Nie	Tak	Tak	Powiązanie modelu z placówką serwisową.
ID_marki	Integer	Nie	Tak	Tak	Powiązanie serwisowanego modelu z marką.

Tabela: Stanowiska

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_stanowiska	Integer	Tak	Nie	Tak	Unikalny identyfikator stanowiska
Nazwa stanowiska	Varchar2(20)	Nie	Nie	Tak	Nazwa stanowiska.
Opis	Varchar2(300)	Nie	Nie	Tak	Szczegółowy opis stanowiska

Tabela: Wynagrodzenia

Atrybut	Typ danych	Klucz główny	Klucz obcy	Wymagany	Opis
ID_wynagrodzenia	Integer	Tak	Nie	Tak	Unikalny identyfikator wynagrodzenia
Data	Date	Nie	Nie	Tak	Data wynagrodzenia
Kwota	Number(10, 2)	Nie	Nie	Tak	Kwota wynagrodzenia
Kwota_dod	Number(8, 2)	Nie	Nie	Nie	Kwota dodatkowa wynagrodzenia
ID_pracownika	Integer	Nie	Tak	Tak	pracownik którego dotyczy dane zmówienie



Rys. 2. Model Logiczny w Toad Data Modeler

4.5. Więzy integralności

Wprowadzone więzy integralności zapewniają spójność i poprawność danych w bazie:

— **Więzy kluczy głównych:** Każda tabela ma klucz główny tworzony schematem Id_"nazwa encji", które są unikatowe oraz nie równe NULL. W przypadku tabeli łączących kluczami głównymi są klucze główne encji, które ta tabela łączy. Klucze główne tworzone są automatycznie na podstawie zdefiniowanych sekwencji.

— **Więzy kluczy obcych** Tabele są połączone z innymi tabelami za pomocą kluczy obcych (foreign key). Przykładem modelowego użycia jest użycie w tabeli Samochody. Tabela Samochody ma 4 klucze obce, każdy ten klucz, jeśli nie jest null musi istnieć rekord o takim kluczu w tabeli, dla której jest kluczem głównym. Jednym z tych kluczy, który może być równy NULL - jest to Id_klienta oznaczające powiązanie z klientem, który jest właścicielem samochodu - jeśli samochód nie ma właściciela jest on równy NULL. Analogicznie dla

— **Integralność dziedzin** Każdy atrybut w każdej tabeli ma swoją dziedzinę. Klucze główne mają typ danych Integer, Pola tekstowe mają odpowiednio dostosowane długości Varchar, jak pokazano w tabelach, numer pesel jest typu Char(11), kod pocztowy Char(6), pola opisowe Varchar2(1000) lub Varchar2(300) (dla z założenia krótszego opisu stanowiska), daty typ Date. Wprowadziliśmy również niestandardowe dziedziny takie jak Stan_technicznyD, który może przyjmować wartości: ('nowy', 'dobry', 'akceptowalny', 'wymaga serwisu'), oraz Typ_transakcjiD, który może przyjmować wartości "kupno" lub "wypożyczenie", które są sprawdzane za pomocą odpowiednich warunków.

— **Wartości NULL** Ważnym więzem integralnościowym jest decyzja o tym czy wartość może być NULL. W kilku miejscach jest pozwolenie na nie wstawienie do danego atrybutu danych - przykładowo wspomniane wcześniej pole Id_klienta w tabeli samochody.

4.6. Proces denormalizacji – analiza i przykłady

Dla naszej bazy danych nie zastosowaliśmy denormalizacji, jednak jeśli byśmy chcieli można by ją przeprowadzić, można by wrzucić modele to tablicy samochody. Usuwa nam to konieczność posiadania tablicy

modele i odwoływania się do niej za każdym razem gdy jest potrzebna informacja o modelu samochodu, jednak tracimy wtedy możliwość pamiętania o modelu bez samochodu, co może być przydatne, natomiast uzyskana poprawa w działaniu byłaby znikoma. W wyniku analizy doszliśmy również do wniosku, ze wszystkie inne próby deneormalizacji doprowadzały by do utraty niektórych danych, np. historii wypłat, historii transakcji itd.

5. Faza fizyczna

W warstwie fizycznej nastąpiło zaimplementowanie bazy danych do Silnika Zarządzania Bazą Danych. Zostało nam to ułatwione przez możliwość utworzenia skryptu SQL do założenia bazy przez narzędzie do modelowania - przez to wystarczyło dokonać tylko kilka zmian, co wynikało z dobrego modelu logicznego.

5.1. Projekt transakcji i weryfikacja ich wykonalności

W ramach transakcji wyróżniliśmy te operacje:

Tabela 24. Projekt transakcji i weryfikacja ich wykonalności

Transakcja	**Wykorzystywane tabele**	**Czy do zrealizowania?**
Dodawanie, usuwanie pracownika	Pracownicy, Placówki	TAK
Modyfikacja danych osobowych pracownika	Pracownicy, (Adres, Wynagrodzenie)	TAK
Zmiana stanowiska pracownika	Stanowiska	TAK
Zmiana wynagrodzenia pracownika	Wynagrodzenia	TAK
Zmiana przypisanej placówki pracownika	Pracownicy, Placówki	TAK
Dodawanie, usuwanie placówek	Placowki, Klienci_placowki	TAK
Modyfikacja adresu placówki	Placówki, Adresy	TAK
Przypisywanie pracowników i samochodów do placówek	Pracownicy, Samochody, Placówki	TAK
Modyfikacja cech specjalnych placówek	Placówki	TAK
Dodawanie, usuwanie samochodów	Samochody	TAK
Zmiana przypisanej placówki dla samochodu	Samochody	TAK
Zmiana numeru rejestracyjnego samochodu	Samochody	TAK
Zmiana stanu technicznego lub przebiegu samochodu	Samochody	TAK
Dodawanie, usuwanie klientów	Klienci, Placówki	TAK
Zmiana danych klienta	Klienci	TAK
Przypisywanie klientów do placówek	Klienci, Placówki	TAK
Dodawanie, usuwanie serwisów	Serwisy	TAK
Przypisywanie pracowników do serwisów	Pracownicy, Serwisy	TAK
Przypisywanie samochodów i klientów do serwisów	Samochody, Klienci, Serwisy	TAK
Transakcja kupienia samochodu	Samochody,Transakcje	TAK
Transakcja wynajmu samochodu	Samochody,Transakcje	TAK
Dodanie wynagrodzenia dla pracownika	Wynagrodzenia, Pracownicy	TAK
Dodanie serwisu	Serwisy	TAK
Dodanie wynagrodzenia dla pracownika	Wynagrodzenia, Pracownicy	TAK
Dodanie wynagrodzenia dla pracownika	Wynagrodzenia, Pracownicy	TAK

5.2. Struktura bazy danych – dobór indeksów

Odpowiedni i optymalny dobór indeksów wymaga sporego doświadczenia w związku z czym zdecydowaliśmy się na pozostawienie domyślnie utworzonych indeksów powiązanych z kluczami obcymi. Takie rozwiązanie pozwala na przyspieszenie działania wszystkich zapytań związanych z łączeniem tabel, kosztem zwiekszena zajmowanej pamieci.

5.3. Skrypt SQL zakładający bazę danych

Listing 1. Skrypt SQL zakładający bazę danych

```
-- create tables section -------------------------------------------------

-- table salony_samochodowe

create table salony_samochodowe(
  id_salonu integer generated always as identity(
    start with 1
    increment by 1
    nomaxvalue
    nominvalue
    cache 20) not null,
  nazwa_salonu varchar2(30 ) not null,
  id_adresu integer not null
)
/

-- create indexes for table salony_samochodowe

create index ix_salon_ma_adres on salony_samochodowe (id_adresu)
/

-- add keys for table salony_samochodowe

alter table salony_samochodowe add constraint unique_identifier1 primary key (id_salonu)
/

alter table salony_samochodowe add constraint nazwa_salonu unique (nazwa_salonu)
/

-- table pracownicy

create table pracownicy(
  id_pracownika integer generated always as identity(
    start with 1
    increment by 1
    nomaxvalue
    nominvalue
    cache 20) not null,
  imie varchar2(20 ) not null,
  nazwisko varchar2(30 ) not null,
  pesel char(11 ),
  numer_telefonu varchar2(15 ) not null,
  id_salonu integer,
  id_placowki integer,
  id_stanowiska integer not null,
  id_adresu integer not null
)
/
```

```
-- create indexes for table pracownicy

create index ix_jest_pracownikiem_salonu on pracownicy (id_salonu)
/

create index ix_pracuje_w_placowce on pracownicy (id_placowki)
/

create index ix_ma_satnowisko on pracownicy (id_stanowiska)
/

create index ix_praconik_ma_adres on pracownicy (id_adresu)
/

-- add keys for table pracownicy

alter table pracownicy add constraint unique_identifier2 primary key (id_pracownika)
/

-- table samochody

create table samochody(
  id_samochodu integer generated always as identity(
    start with 1
    increment by 1
    nomaxvalue
    nominvalue
    cache 20) not null,
  numer_rejestracyjny varchar2(8 ) not null,
  data_produkcji date not null,
  przebieg integer not null,
  waznosc_przegladu date not null,
  stan_techniczny varchar2(30 ) not null
        check (stan_techniczny in ('nowy', 'dobry', 'akceptowalny', 'wymaga serwisu')),
  czy_powypadkowy char(1 ) not null,
  cena_wynajmu number(10,2),
  cena_sprzedazy number(10,2),
  id_klienta integer,
  id_salonu integer,
  id_modelu integer not null,
  id_placowki integer
)
/

-- create indexes for table samochody

create index ix_klient_posiada_samochod on samochody (id_klienta)
/

create index ix_salon_posiada_samochody on samochody (id_salonu)
/

create index ix_samochod_jest_dango_modelu on samochody (id_modelu)
/

create index ix_placowka_posiada_samochody on samochody (id_placowki)
/

-- add keys for table samochody
```

```
alter table samochody add constraint unique_identifier5 primary key (id_samochodu)
/

alter table samochody add constraint numer_rejestracyjny unique (numer_rejestracyjny)
/

-- table klienci

create table klienci(
  id_klienta integer generated always as identity(
    start with 1
    increment by 1
    nomaxvalue
    nominvalue
    cache 20) not null,
  imie varchar2(20 ) not null,
  nazwisko varchar2(30 ) not null,
  numer_telefonu varchar2(15 ) not null,
  id_salonu integer,
  id_adresu integer not null
)
/

-- create indexes for table klienci

create index ix_jest_klientem_salonu on klienci (id_salonu)
/

create index ix_klient_ma_adres on klienci (id_adresu)
/

-- add keys for table klienci

alter table klienci add constraint unique_identifier9 primary key (id_klienta)
/

-- table serwisy

create table serwisy(
  id_serwisu integer generated always as identity(
    start with 1
    increment by 1
    nomaxvalue
    nominvalue
    cache 20) not null,
  data_rozpoczecia date not null,
  koszt number(10,2) not null,
  opis varchar2(1000 ) not null,
  id_samochodu integer not null,
  id_klienta integer,
  id_placowki integer not null
)
/

-- create indexes for table serwisy

create index ix_samochod_jest_serwisowany on serwisy (id_samochodu)
/

create index ix_ma_serwisowany_samochod on serwisy (id_klienta)
/
```

```
create index ix_serwis_odbywa_sie_w_warszatcie on serwisy (id_placowki)
/

-- add keys for table serwisy

alter table serwisy add constraint unique_identifier10 primary key (id_serwisu)
/

-- table placowki

create table placowki(
  id_placowki integer generated always as identity(
    start with 1
    increment by 1
    nomaxvalue
    nominvalue
    cache 20) not null,
  nazwa varchar2(30 ) not null,
  id_salonu integer not null,
  id_adresu integer not null
)
/

-- create indexes for table placowki

create index ix_posiada_placowke on placowki (id_salonu)
/

create index ix_placowka_ma_adres on placowki (id_adresu)
/

-- add keys for table placowki

alter table placowki add constraint unique_identifier11 primary key (id_placowki)
/

alter table placowki add constraint nazwa unique (nazwa)
/

-- table punkty_handlowe

create table punkty_handlowe(
  id_placowki integer not null,
  czy_wynajem char(1 ) not null,
  czy_sprzedaz char(1 ) not null,
  czy_uzywane char(1 ) not null
)
/

-- add keys for table punkty_handlowe

alter table punkty_handlowe add constraint unique_identifier12 primary key (id_placowki)
/

-- table warsztaty

create table warsztaty(
  id_placowki integer not null,
  liczba_stanowisk integer not null,
  czy_wymiana_opon char(1 ) not null
```

```
)
/

-- add keys for table warsztaty

alter table warsztaty add constraint unique_identifier14 primary key (id_placowki)
/

-- table przydzialy_serwisowe

create table przydzialy_serwisowe(
  id_pracownika integer not null,
  id_serwisu integer not null
)
/

-- table klienci_placowki

create table klienci_placowki(
  id_klienta integer not null,
  id_placowki integer not null
)
/

-- table transakcje

create table transakcje(
  id_samochodu integer not null,
  id_klienta integer not null,
  typ_transakcji varchar2(8 ) not null
        check (typ_transakcji in ('wynajem','kupno')),
  data_transakcji date not null,
  data_konca_wynajmu date
)
/

-- table adresy

create table adresy(
  id_adresu integer generated always as identity(
    start with 1
    increment by 1
    nomaxvalue
    nominvalue
    cache 20) not null,
  miasto varchar2(30 ) not null,
  ulica varchar2(30 ) not null,
  numer_budynku varchar2(5 ) not null,
  numer_lokalu varchar2(5 ),
  kod_pocztowy char(6 ) not null
)
/

-- add keys for table adresy

alter table adresy add constraint pk_adresy primary key (id_adresu)
/

-- table modele

create table modele(
```

```
  id_modelu integer generated always as identity(
    start with 1
    increment by 1
    nomaxvalue
    nominvalue
    cache 20) not null,
  nazwa_modelu varchar2(10 ) not null,
  id_marki integer not null,
  id_salonu integer not null
)
/

-- create indexes for table modele

create index ix_model_jest_danej_marki on modele (id_marki)
/

create index ix_model_jest_obslugiwany_przez_salon on modele (id_salonu)
/

-- add keys for table modele

alter table modele add constraint pk_modele primary key (id_modelu)
/

alter table modele add constraint nazwa_modelu unique (nazwa_modelu)
/

-- table marki

create table marki(
  id_marki integer generated always as identity(
    start with 1
    increment by 1
    nomaxvalue
    nominvalue
    cache 20) not null,
  nazwa_marki varchar2(10 ) not null,
  id_salonu integer not null
)
/

-- create indexes for table marki

create index ix_salon_obsluguje_marki on marki (id_salonu)
/

-- add keys for table marki

alter table marki add constraint pk_marki primary key (id_marki)
/

alter table marki add constraint nazwa_marki unique (nazwa_marki)
/

-- table serwisowane_marki

create table serwisowane_marki(
  id_placowki integer not null,
  id_marki integer not null
)
```

```
/

-- add keys for table serwisowane_marki

alter table serwisowane_marki add constraint pk_serwisowane_marki primary key
(id_placowki,id_marki)
/

-- table stanowiska

create table stanowiska(
  id_stanowiska integer generated always as identity(
    start with 1
    increment by 1
    nomaxvalue
    nominvalue
    cache 20) not null,
  nazwa_stanowiska varchar2(20 ) not null,
  opis varchar2(300 ) not null
)
/

-- add keys for table stanowiska

alter table stanowiska add constraint pk_stanowiska primary key (id_stanowiska)
/

alter table stanowiska add constraint nazwa_stanowiska unique (nazwa_stanowiska)
/

-- table wynagrodzenia

create table wynagrodzenia(
  id_wynagrodzenia integer generated always as identity(
    start with 1
    increment by 1
    nomaxvalue
    nominvalue
    cache 20) not null,
  data date not null,
  kwota number(10,2) not null,
  kwota_dod number(8,2),
  id_pracownika integer not null
)
/

-- create indexes for table wynagrodzenia

create index ix_otrzymuje_wynagrodzenie on wynagrodzenia (id_pracownika)
/

-- add keys for table wynagrodzenia

alter table wynagrodzenia add constraint pk_wynagrodzenia primary key (id_wynagrodzenia)
/


-- create foreign keys (relationships) section ------------------------------------------------

alter table pracownicy add constraint jest_pracownikiem_salonu foreign key (id_salonu)
references salony_samochodowe (id_salonu)
```

```
/


alter table pracownicy add constraint pracuje_w_placowce foreign key (id_placowki)
references placowki (id_placowki)
/


alter table placowki add constraint salon_posiada_placowke foreign key (id_salonu)
references salony_samochodowe (id_salonu)
/


alter table serwisy add constraint samochod_jest_serwisowany foreign key (id_samochodu)
references samochody (id_samochodu)
/


alter table klienci add constraint jest_klientem_salonu foreign key (id_salonu)
references salony_samochodowe (id_salonu)
/


alter table samochody add constraint klient_posiada_samochod foreign key (id_klienta)
references klienci (id_klienta)
/


alter table samochody add constraint salon_posiada_samochody foreign key (id_salonu)
references salony_samochodowe (id_salonu)
/


alter table serwisy add constraint ma_serwisowany_samochod foreign key (id_klienta)
references klienci (id_klienta)
/


alter table modele add constraint model_jest_danej_marki foreign key (id_marki)
references marki (id_marki)
/


alter table samochody add constraint samochod_jest_dango_modelu foreign key (id_modelu)
references modele (id_modelu)
/


alter table serwisowane_marki add constraint warsztat_serwisuje_marki foreign key (id_placowki)
references warsztaty (id_placowki)
/
```

```
alter table serwisowane_marki add constraint marka_jest_serwisowana_przez_warsztat foreign key
(id_marki) references marki (id_marki)
/


alter table modele add constraint model_jest_obslugiwany_przez_salon foreign key (id_salonu)
references salony_samochodowe (id_salonu)
/


alter table marki add constraint marki_obslugiwane_przez_salon foreign key (id_salonu)
references salony_samochodowe (id_salonu)
/


alter table pracownicy add constraint ma_stanowisko foreign key (id_stanowiska)
references stanowiska (id_stanowiska)
/


alter table wynagrodzenia add constraint otrzymuje_wynagrodzenie foreign key (id_pracownika)
references pracownicy (id_pracownika)
/


alter table serwisy add constraint serwis_odbywa_sie_w_warsztacie foreign key (id_placowki)
references warsztaty (id_placowki)
/


alter table samochody add constraint placowka_posiada_samochody foreign key (id_placowki)
references punkty_handlowe (id_placowki)
/


alter table klienci_placowki add constraint placowka_ma_klientow foreign key (id_placowki)
references punkty_handlowe (id_placowki)
/


alter table salony_samochodowe add constraint salon_ma_adres foreign key (id_adresu)
references adresy (id_adresu)
/


alter table pracownicy add constraint pracownik_ma_adres foreign key (id_adresu)
references adresy (id_adresu)
/
```

```
alter table placowki add constraint placowka_ma_adrs foreign key (id_adresu)
references adresy (id_adresu)
/


alter table klienci add constraint klient_ma_adres foreign key (id_adresu)
references adresy (id_adresu)
/
```

5.4. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

5.4.1. Zapytania tworzące przykładowy zbiór danych

5.5. Skrypt SQL zakładający bazę danych

Listing 2. Skrypt SQL dodający przykładowe dane

```
-- Table: adresy
insert into adresy (miasto, ulica, numer_budynku, kod_pocztowy) values
('Warszawa', 'Nowowiejska', '15', '00-665');
insert into adresy (miasto, ulica, numer_budynku, numer_lokalu, kod_pocztowy) values
('Warszawa', 'plac Defilad', '1', '2716' ,'00-901');
insert into adresy (miasto, ulica, numer_budynku, kod_pocztowy) values
('Warszawa', 'Krajewskiego', '1A', '01-532');
insert into adresy (miasto, ulica, numer_budynku, kod_pocztowy) values
('Studzianki Pancerne', 'Studzianki Pancerne', '14', '26-903');
insert into adresy (miasto, ulica, numer_budynku, numer_lokalu, kod_pocztowy) values
('Zielonka', '1 Maja', '1', '12B', '05-220');
insert into adresy (miasto, ulica, numer_budynku, kod_pocztowy) values
('Warszawa', 'Krakowskie Przedmiescie', '46/48', '00-071');
insert into adresy (miasto, ulica, numer_budynku, numer_lokalu, kod_pocztowy) values
('Siedlce', 'Jozefa Pilsudskiego', '47', '8', '08-110');
insert into adresy (miasto, ulica, numer_budynku, numer_lokalu, kod_pocztowy) values
('Warszawa', 'Sylwestra Kaliskiego', '2', '65', '01-485');
insert into adresy (miasto, ulica, numer_budynku, kod_pocztowy) values
('Sulejowek', 'Jozefa Pilsudskiego', '29', '05-070');

-- Table: salony_samochodowe
insert into salony_samochodowe (nazwa_salonu, id_adresu) values
('AutoPolska', 1);

-- Table: placowki
insert into placowki (nazwa, id_salonu, id_adresu) values
('AutoSprzedaz', 1, 1);
insert into placowki (nazwa, id_salonu, id_adresu) values
('AutoSerwis', 1, 2);

-- Table: marki
insert into marki (nazwa_marki, id_salonu) values
('FSO', 1);
insert into marki (nazwa_marki, id_salonu) values
('IFA', 1);

-- Table: modele
insert into modele (nazwa_modelu, id_marki, id_salonu) values
('Warszawa', 1, 1);
```

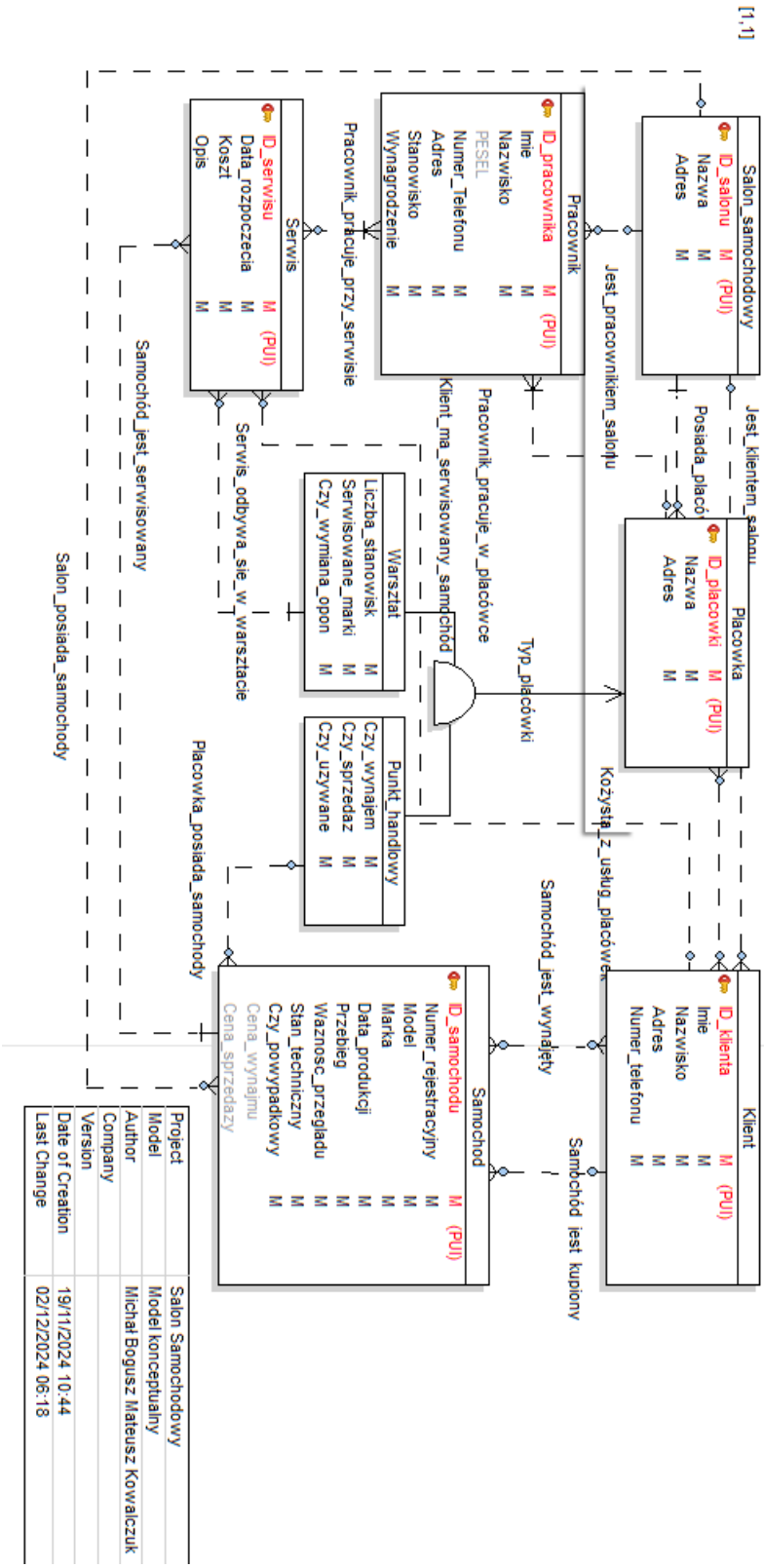
```
insert into modele (nazwa_modelu, id_marki, id_salonu) values
('Wartburg', 2, 1);


-- Table: klienci
insert into klienci (imie, nazwisko, numer_telefonu, id_adresu) values
('Katarzyna', 'Wisniewska', '503456789', 9);
insert into klienci (imie, nazwisko, numer_telefonu, id_adresu) values
('Wojciech', 'Siwicki', '230706890', 6);
insert into klienci (imie, nazwisko, numer_telefonu, id_adresu) values
('Andrzej', 'Nowotko', '123098456', 8);

-- Table: punkty_handlowe
insert into punkty_handlowe (id_placowki, czy_wynajem, czy_sprzedaz, czy_uzywane) values
(1,'Y','Y','Y');

-- Table: warsztaty:
insert into warsztaty (id_placowki, liczba_stanowisk, czy_wymiana_opon) values
(2,8,'Y');

--Table: Klienci_placowki
insert into klienci_placowki (id_klienta,id_placowki) values
(1,1);
insert into klienci_placowki (id_klienta,id_placowki) values
(3,1);
insert into klienci_placowki (id_klienta,id_placowki) values
(2,1);

-- Table: samochody
insert into samochody (numer_rejestracyjny, data_produkcji, przebieg, waznosc_przegladu,
stan_techniczny, czy_powypadkowy, cena_wynajmu, id_placowki, id_modelu) values
('WX12345', to_date('1965-06-15', 'YYYY-MM-DD'), 150000, to_date('2025-06-15', 'YYYY-MM-DD'),
'dobry', 'N', 120.50, 1, 1);
insert into samochody (numer_rejestracyjny, data_produkcji, przebieg, waznosc_przegladu,
stan_techniczny, czy_powypadkowy, cena_sprzedazy, id_placowki, id_modelu) values
('UB3456', to_date('1981-12-13', 'YYYY-MM-DD'), 300000, to_date('2024-05-10', 'YYYY-MM-DD'),
'akceptowalny', 'Y', 80000, 1, 2);
insert into samochody (numer_rejestracyjny, data_produkcji, przebieg, waznosc_przegladu,
stan_techniczny, czy_powypadkowy, id_klienta, id_modelu) values
('HB1111', to_date('1966-12-24', 'YYYY-MM-DD'), 1000, to_date('2028-12-10', 'YYYY-MM-DD'),
'dobry', 'N', 2, 1);

--Table: transakcje
insert into transakcje (id_samochodu, id_klienta, typ_transakcji, data_transakcji) values
(3,2,'Kupno',to_date('2005-12-24', 'YYYY-MM-DD'));
insert into transakcje (id_samochodu, id_klienta, typ_transakcji, data_transakcji,
data_konca_wynajmu) values
(1,3,'Wynajem',to_date('2005-6-24', 'YYYY-MM-DD'),to_date('2010-6-24', 'YYYY-MM-DD'));

-- Table: stanowiska
insert into stanowiska (nazwa_stanowiska, opis) values
('Sprzedawca', 'Odpowiedzialny za sprzedaz pojazd w');
insert into stanowiska (nazwa_stanowiska, opis) values
('Mechanik', 'Przeprowadza serwisy pojazd w');
insert into stanowiska (nazwa_stanowiska, opis) values
('Kierownik', 'Zarzadza przedsiebiorstwem');
insert into stanowiska (nazwa_stanowiska, opis) values
('Kierownik warsztatu', 'zarzadza serwisami');

-- Table: pracownicy
insert into pracownicy (imie, nazwisko, pesel, numer_telefonu, id_salonu, id_stanowiska,
```

```
id_adresu) values
('Jan', 'Kowalski', '89012345678', '501234567', 1, 3, 3);
insert into pracownicy (imie, nazwisko, pesel, numer_telefonu, id_placowki, id_stanowiska,
id_adresu) values
('Anna', 'Nowak', '92040578901', '502345678', 1,1, 5);
insert into pracownicy (imie, nazwisko, pesel, numer_telefonu, id_placowki, id_stanowiska,
id_adresu) values
('Jan', 'Bogdan', '88888888888', '888888888', 2,4, 4);
insert into pracownicy (imie, nazwisko, pesel, numer_telefonu, id_placowki, id_stanowiska,
id_adresu) values
('Gustaw', 'Husak', '92041118912', '423867421', 2,2, 9);


-- Table: serwisy
insert into serwisy (data_rozpoczecia, koszt, opis, id_samochodu, id_placowki) values
(to_date('2024-11-25', 'YYYY-MM-DD'), 500.00, 'Wymiana klock w hamulcowych', 1, 2);
insert into serwisy (data_rozpoczecia, koszt, opis, id_samochodu, id_placowki) values
(to_date('2024-12-01', 'YYYY-MM-DD'), 800.00, 'Przeglad techniczny', 2, 2);

-- Table: Przydzialy_serwisowe
insert into przydzialy_serwisowe (id_pracownika, id_serwisu) values
(4,1);
insert into przydzialy_serwisowe (id_pracownika, id_serwisu) values
(4,2);

--Table: serwisowane_marki
insert into serwisowane_marki (id_placowki, id_marki) values
(2,1);
insert into serwisowane_marki (id_placowki, id_marki) values
(2,2);

-- Table: wynagrodzenia
insert into wynagrodzenia (data, kwota, kwota_dod, id_pracownika) values
(to_date('2024-11-30', 'YYYY-MM-DD'), 4000.00, 500.00, 1);
insert into wynagrodzenia (data, kwota, kwota_dod, id_pracownika) values
(to_date('2024-11-30', 'YYYY-MM-DD'), 4500.00, 300.00, 2);
insert into wynagrodzenia (data, kwota, kwota_dod, id_pracownika) values
(to_date('2024-11-30', 'YYYY-MM-DD'), 6000.00, 500.00, 3);
insert into wynagrodzenia (data, kwota, kwota_dod, id_pracownika) values
(to_date('2024-11-30', 'YYYY-MM-DD'), 700.00, 300.00, 4);
```

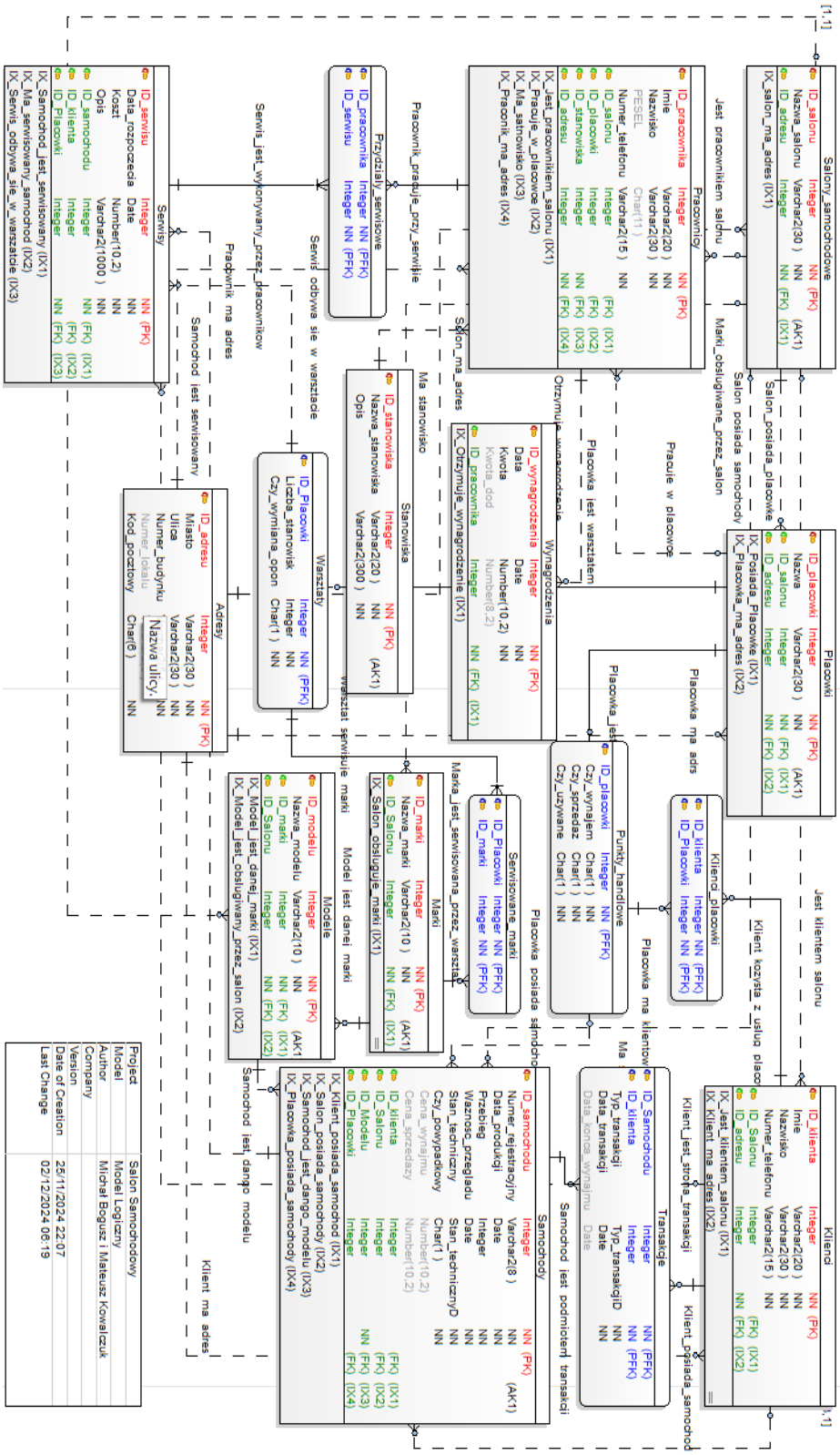
5.5.1. przykładowe zapytania

```
-- Wyszukamnie wszystkich autb w plac wkach w Warszawie
select * from samochody S join modele M on (M.id_modelu = S.id_modelu)
join marki MA on (MA.id_marki = M.id_marki)
where S.ID_placowki IN (Select ID_placowki from placowki P join adresy A on
(P.id_adresu = A.id_adresu) where A.miasto Like 'Warszawa');

-- Sprawdzanie ile razy by Ć serwisowany dany samoch d
Select A.id_samochodu ,count(*) from serwisy S join samochody A on
(S.id_samochodu = A.id_samochodu)
Group by A.id_samochodu;

-- Dodanie samochodu istniejacego typu (np. Warszawty):
insert into samochody (Numer_rejestracyjny, data_produkcji, przebieg,
waznosc_przegladu,stan_techniczny,czy_powypadkowy,cena_sprzedazy,id_placowki,id_modelu) values
('WD2137', to_date('1984-11-30', 'YYYY-MM-DD'), 70, to_date('2025-11-30', 'YYYY-MM-DD'),'wymaga
serwisu','T',90000,1,(Select id_modelu from modele where nazwa_modelu Like 'Warszawa'));
```

17



Rys. 3. Schemat konceptualny



Rys. 4. Model Logiczny w Toad Data Modeler