

Wstęp do Baz Danych
Projekt 1.

Salon Samochodowy

Krzysztof Kobyliński i Rafał Szczepanik

Spis treści

1.	Zakres i cel projektu	4
	Plan biznesowy:	4
	Założenia przyjęte w projekcie:	4
2.	Definicja systemu	4
2.1.	Perspektywy użytkowników	4
	Właściciel/ Manager	4
	Klient	4
	Sprzedawca	4
	Księgowy	4
2.2.	Perspektywy użytkowników	5
3.	Model konceptualny	5
3.1.	Definicja zbiorów encji określonych w projekcie	5
	Encja 'Salon_samochodowy'	5
	Encja 'Samochod'	5
	Encja 'Klient'	5
	Encja 'Punkt_sprzedaży'	5
	Encja 'Wynagrodzenie'	5
	Encja 'Wyposazenie'	5
	Encja 'Pracownik'	5
	Encja 'Sprzedawca'	5
3.2.	Ustalenie związków między encjami i ich typów	6
3.3.	Określenie atrybutów i ich dziedzin	7
	Encja 'Salon_samochodowy'	7
	Encja 'Samochod'	7
	Encja 'Klient'	8
	Encja 'Punkt_sprzedaży'	8
	Encja 'Wyposazenie'	8
	Encja 'Pracownik'	9
	Encja 'Sprzedawca'	9
3.4.	Dodatkowe reguły integralnościowe	10
3.5.	Klucze kandydujące i główne	10
3.6.	Schemat ER na poziomie konceptualnym	11
3.7.	Problem pułapek szczelinowych i wachlarzowych	11
4.	Model logiczny	14
4.1.	Charakterystyka modelu relacyjnego	14

4.2.	Usunięcie właściwości niekompatybilnych z modelem relacyjnym	14
4.2.1.	Schemat po konwersji – przed procesem normalizacji	15
4.3.	Proces normalizacji – analiza i przykłady	16
4.4.	Schemat ER na poziomie modelu logicznego	17
4.5.	Więzy integralności	18
	Encja „Salony Samochodowe”	18
	Encja „Punkty sprzedaży”	18
	Encja „Klienci”	19
	Encja „Pracownicy”	19
	Encja „Miejsca_pracy”	20
	Encja „Księgowi”	20
4.6.	Proces denormalizacji – analiza i przykłady	23
5.	Faza fizyczna	24
5.1.	Projekt transakcji i weryfikacja ich wykonalności	24
5.2.	Strojenie bazy danych – dobór indeksów.....	25
5.3.	Skrypt SQL zakładający bazę.....	25
5.4.	Przykłady zapytań i poleceń SQL odnoszących się do bazy danych	Błąd! Nie zdefiniowano zakładki.
	Bibliografia.....	34

1. Zakres i cel projektu

Celem projektu jest projekt relacyjnej bazy danych na poziomie koncepcyjnym i logicznym. Końcowym etapem jest fizyczne założenie bazy danych.

Wykorzystano oprogramowanie:

- Oracle Database 19c
- Toad Data Modeler 7.1

Plan biznesowy:

Baza danych ma reprezentować salon samochodowy. Salon samochodowy posiada wiele punktów sprzedaży, zlokalizowanych w różnych miejscach. W punktach sprzedaży pracują sprzedawcy. Jeden sprzedawca może pracować w wielu punktach sprzedaży. Salon samochodowy oferuje samochody różnych marek i modeli. Klienci mogą dokonać zakupu samochodu lub przeprowadzić jazdę testową, jednym z dostępnych samochodów. Przed zakupem klient może dokonać wyboru wyposażenia, które ma posiadać jego samochód.

Oprócz sprzedawców salon zatrudnia również księgowych. Zarządzają oni danymi dotyczącymi pracowników (wynagrodzenia, historia zatrudnienia, kwalifikacje), dokonanych transakcji i posiadanych samochodów.

Założenia przyjęte w projekcie:

- Salon samochodowy posiada wiele lokalnych placówek.
- Istnieją dwie kategorie pracowników zatrudnianych przez salon – sprzedawcy oraz księgowi.
- Klienci oprócz zakupu samochodu mogą również przeprowadzać jazdy testowe.
- Salon oferuje modele samochodów różnych marek.
- Samochody mogą posiadać różne wyposażenie, które jest dobierane przez klienta przed zakupem.
- W bazie znajduje się dane historyczne dotyczące dokonanych transakcji (sprzedaży samochodów), jazd testowych oraz wypłaconych wynagrodzeń pracowniczych.

2. Definicja systemu

2.1. Perspektywy użytkowników

Właściciel/ Manager

Ma dostęp do wszystkich funkcjonalności systemu w pełnym wymiarze – możliwość modyfikacji.

Klient

Ma możliwość podglądu interesujących go aspektów systemu – dostępnych samochodów i ich wyposażeni oraz dostępnych jazd próbnych.

Sprzedawca

Ma dostęp do funkcji systemowych związanych z klientami (modyfikowanie ich danych), możliwość modyfikowania danych dotyczących transakcji, dostępnych i przeprowadzonych jazd próbnych. Ma możliwość podglądu dostępnych wyposażań.

Księgowy

Ma możliwość podglądu danych dotyczących pracowników, klientów, sprzedaży, wypłaconych wynagrodzeń, przeprowadzonych jazd próbnych w celu prowadzenia księgowości.

2.2. Perspektywy użytkowników

Dostęp:	Właściciel/ Manager	Klient	Sprzedawca	Księgowy
Dane placówek	M	-	P	P
Dane pracowników	M	-	-	P
Wypłacone wynagrodzenia	M	-	-	M
Dane klientów	M	-	M	P
Dostępne samochody	M	P	M	P
Dokonane transakcje - sprzedane samochody	M	-	M	P
Przeprowadzone jazdy próbne	M	-	M	P
Dostępne wyposażenia	M	P	P	P

M - modyfikowanie/dodawanie/usuwanie

P – podgląd

3. Model konceptualny

3.1. Definicja zbiorów encji określonych w projekcie

Encja 'Salon_samochodowy'

Encja salonu samochodowego - główna encja, przedstawia podstawowe dane firmy

Encja 'Samochod'

Encja Samochodu - reprezentuje konkretny samochód

Encja 'Klient'

Encja klienta - gromadzi podstawowe dane o kliencie

Encja 'Punkt_sprzedazy'

Encja punktu sprzedaży - reprezentuje jedną placówkę firmy

Encja 'Wynagrodzenie'

Encja wynagrodzenia - encja gromadząca poprzednie wypłaty dla pracowników

Encja 'Wyposazenie'

Wyposażenie samochodu - encja reprezentująca konkretne elementy wyposażenia samochodu

Encja 'Pracownik'

Encja pracownika - reprezentuje dane personalne pracownika firmy

Encja 'Ksiegowy'

Uszczegółowienie encji pracownika

Encja 'Sprzedawca'

Uszczegółowienie encji pracownika

3.2. Ustalenie związków między encjami i ich typów

Relacja pomiędzy	Krotność	Opis
Salon_samochodowy - Pracownik	1..1 - 0..m	Salon samochodowy jest zawsze jeden i może zatrudniać od zera do wielu pracowników.
Salon_samochodowy - Punkt_sprzedazy	1..1 - 0..m	Salon samochodowy może mieć od zera do wielu punktów sprzedaży.
Salon_samochodowy - Samochod	1..1 - 0..m	Samochód zawsze należy do jednego konkretnego salonu samochodowego. Salon w skrajnym przypadku może jeszcze nie mieć żadnego samochodu.
Salon_samochodowy - Klient	1..1 - 0..m	Salon samochodowy jest zawsze jeden i może mieć od zera do wielu klientów.
Pracownik - Wynagrodzenie	1..1 - 0..m	Pracownik mógł mieć wiele wynagrodzeń w przeszłości, w skrajnym przypadku nie dostał jeszcze żadnego. Dane wynagrodzenie należy zawsze do konkretnego pracownika.
Pracownik - Samochod	0..n - 0..m	Pod kontrolą pracownika może być zero samochodów (księgowy) lub wiele. Samochód może nie być jeszcze przypisany do żadnego pracownika, do jednego lub być przepisywany do innych pracowników.
Pracownik - Punkt_sprzedazy	0..n - 0..m	Pracownik może nie pracować w żadnym punkcie sprzedaży (księgowy), ale mógł również zmieniać punkty sprzedaży do których był przypisany w przeszłości. Dany punkt sprzedaży może mieć od zera do wielu pracowników.
Samochod – Klient (jazda testowa)	0..n - 0..m	Relacja dotycząca jazdy testowej. Samochód mógł być testowany przez wielu klientów, albo nie testowany przez nikogo. Tak samo klient mógł testować wiele samochodów, w skrajnym przypadku nie testował żadnego.
Punkt_sprzedazy - Samochod	0..1 - 0..m	Samochód może należeć maksymalnie do jednego punktu sprzedaży. Punkt sprzedaży w skrajnym przypadku może jeszcze nie mieć żadnego samochodu.
Samochod - Wyposazenie	0..n - 0..m	Samochód może mieć od zera do wielu elementów wyposażenia dodatkowego. Tak samo dane wyposażenie może znajdować się w wielu samochodach lub w skrajnym przypadku w żadnym.
Samochod – Klient	1..n - 0..m	Samochód mógł zmieniać właścicieli lub nie miał jeszcze żadnego. Klient posiada od jednego do wielu samochodów.

3.3. Określenie atrybutów i ich dziedzin

Przyjęto następujące oznaczenia:

- PUI - czy atrybut wchodzi w skład klucza głównego
- M (Mandatory) – pole obowiązkowe
- D (Default) – pole domyślne
- R (Rule) - reguła

Encja 'Salon_samochodowy'

PUI	Nazwa atrybutu	Domena	Typ danych	M	D	R	Opis
TAK	ID_Salonu		Integer	TAK	NIE	NIE	Unikalny numer identyfikacyjny
NIE	Nazwa		VarChar(30)	TAK	NIE	NIE	Nazwa Salonu
NIE	Adres		VarChar(400)	TAK	NIE	NIE	Adres salonu, pole segmentowe

Encja 'Samochod'

PUI	Nazwa atrybutu	Domena	Typ danych	M	D	R	Opis
TAK	ID_samochodu		Integer	TAK	NIE	NIE	Unikalny numer identyfikacyjny samochodu
NIE	Marka		VarChar(30)	TAK	NIE	NIE	Marka samochodu
NIE	Model		VarChar(30)	TAK	NIE	NIE	Model samochodu
NIE	Typ_nadwozia		VarChar(15)	TAK	NIE	NIE	Typ nadwozia samochodu
NIE	Pojemnosc		Money	TAK	NIE	NIE	Pojemność silnika samochodu
NIE	Cena_bazowa		Money	TAK	NIE	NIE	Cena bazowa samochodu

Encja 'Klient'

PUI	Nazwa atrybutu	Domena	Typ danych	M	D	R	Opis
TAK	ID_klienta		Integer	TAK	NIE	NIE	Unikalny numer identyfikacyjny klienta
NIE	Imie		VarChar(30)	TAK	NIE	NIE	Imie klienta
NIE	Nazwisko		VarChar(30)	TAK	NIE	NIE	Nazwisko klienta
NIE	PESEL		Character(11)	TAK	NIE	NIE	Numer PESEL klienta
NIE	Plec	PlecD	Character(1)	TAK	NIE	NIE	Płeć klienta
NIE	Adres		VarChar(400)	NIE	NIE	NIE	Adres klienta, pole segmentowe
NIE	Nr_telefonu		VarChar(15)	NIE	NIE	NIE	Numer telefonu klienta
NIE	Email		VarChar(30)	NIE	NIE	NIE	Adres email klienta

Encja 'Punkt_sprzedazy'

PUI	Nazwa atrybutu	Domena	Typ danych	M	D	R	Opis
TAK	ID_punktu		Bigint	TAK	NIE	NIE	Unikalny numer identyfikacyjny punktu sprzedaży
NIE	Adres		VarChar(400)	TAK	NIE	NIE	Adres punktu sprzedaży, pole segmentowe
NIE	Godziny_otwarcia		VarChar(400)	TAK	NIE	NIE	Godziny otwarcia punktu sprzedaży, pole segmentowe
NIE	Email		VarChar(30)	NIE	NIE	NIE	Email kontaktowy punktu sprzedaży
NIE	Nr_telefonu		VarChar(15)	NIE	NIE	NIE	Kontaktowy numer telefonu punktu sprzedaży

Encja 'Wynagrodzenie'

PUI	Nazwa atrybutu	Domena	Typ danych	M	D	R	Opis
TAK	ID_wynagrodzenia		Bigint	TAK	NIE	NIE	Unikalny numer identyfikacyjny wynagrodzenia
NIE	Data_wypłaty		Date	TAK	NIE	NIE	Data wypłaty
NIE	Kwota		Money	TAK	NIE	NIE	Wysokość wynagrodzenia

Encja 'Wypozazenie'

PUI	Nazwa atrybutu	Domena	Typ danych	M	D	R	Opis
TAK	ID_wyposazenia		Integer	TAK	NIE	NIE	Unikalny numer identyfikacyjny wyposażenia
NIE	Typ		VarChar(30)	TAK	NIE	NIE	Typ wyposażenia
NIE	Nazwa		VarChar(30)	TAK	NIE	NIE	Nazwa wyposażenia
NIE	Cena		Money	TAK	NIE	NIE	Cena wyposażenia
NIE	Opis		VarChar(300)	TAK	NIE	NIE	Opis wyposażenia

Encja 'Pracownik'

PUI	Nazwa atrybutu	Domena	Typ danych	M	D	R	Opis
TAK	ID_pracownika		Integer	TAK	NIE	NIE	Unikalny numer identyfikacyjny pracownika
NIE	Imie		VarChar(30)	TAK	NIE	NIE	Imie pracownika
NIE	Nazwisko		VarChar(30)	TAK	NIE	NIE	Nazwisko pracownika
NIE	PESEL		Character(11)	TAK	NIE	NIE	Numer PESEL pracownika
NIE	Plec	PlecD	Character(1)	TAK	NIE	NIE	Płeć pracownika
NIE	Narodowosc		VarChar(30)	TAK	NIE	NIE	Narodowość pracownika
NIE	Nr_konta		VarChar(30)	TAK	NIE	NIE	Numer konta bankowego pracownika
NIE	Adres		VarChar(400)	TAK	NIE	NIE	Adres pracownika, pole segmentowe
NIE	Nr_telefonu		VarChar(15)	TAK	NIE	NIE	Numer telefonu pracownika
NIE	Email		VarChar(30)	TAK	NIE	NIE	Adres email pracownika

Encja 'Ksiegowy'

PUI	Nazwa atrybutu	Domena	Typ danych	M ₁₎	D ₂₎	R ₃₎	Opis
NIE	Dzial		VarChar(30)	TAK	NIE	NIE	Dział pracownika
NIE	Nr_kadrowy		VarChar(15)	TAK	NIE	NIE	Numer kadrowy pracownika
NIE	Login		VarChar(30)	NIE	NIE	NIE	Login pracownika
NIE	Budynek		VarChar(5)	NIE	NIE	NIE	Budynek w którym pracuje pracownik
NIE	Nr_wew		Character(5)	NIE	NIE	NIE	Numer wewnętrzny pracownika
NIE	Pokoj		VarChar(5)	NIE	NIE	NIE	Pokój pracownika

Encja 'Sprzedawca'

PUI	Nazwa atrybutu	Domena	Typ danych	M ₁₎	D ₂₎	R ₃₎	Opis
NIE	Jezyk		VarChar(15)	TAK	NIE	NIE	Języki obce pracownika, pole segmentowe, wielowartościowe
NIE	Prawo_jazdy	Prawo_Jazdy_D	VarChar(4)	NIE	NIE	NIE	Prawa jazdy pracownika
NIE	Specjalnosc		VarChar(30)	NIE	NIE	NIE	Specjalność w sprzedaży pracownika

3.4. Dodatkowe reguły integralnościowe

Dziedziny w większości atrybutów określone są przez typ danych. W celu poprawy integralności danych stworzono specjalne dziedziny:

PlecD – Przyjmuje wartości 'M' lub 'K', definiuje płeć, wykorzystywaną w encji Pracownik oraz Klient.

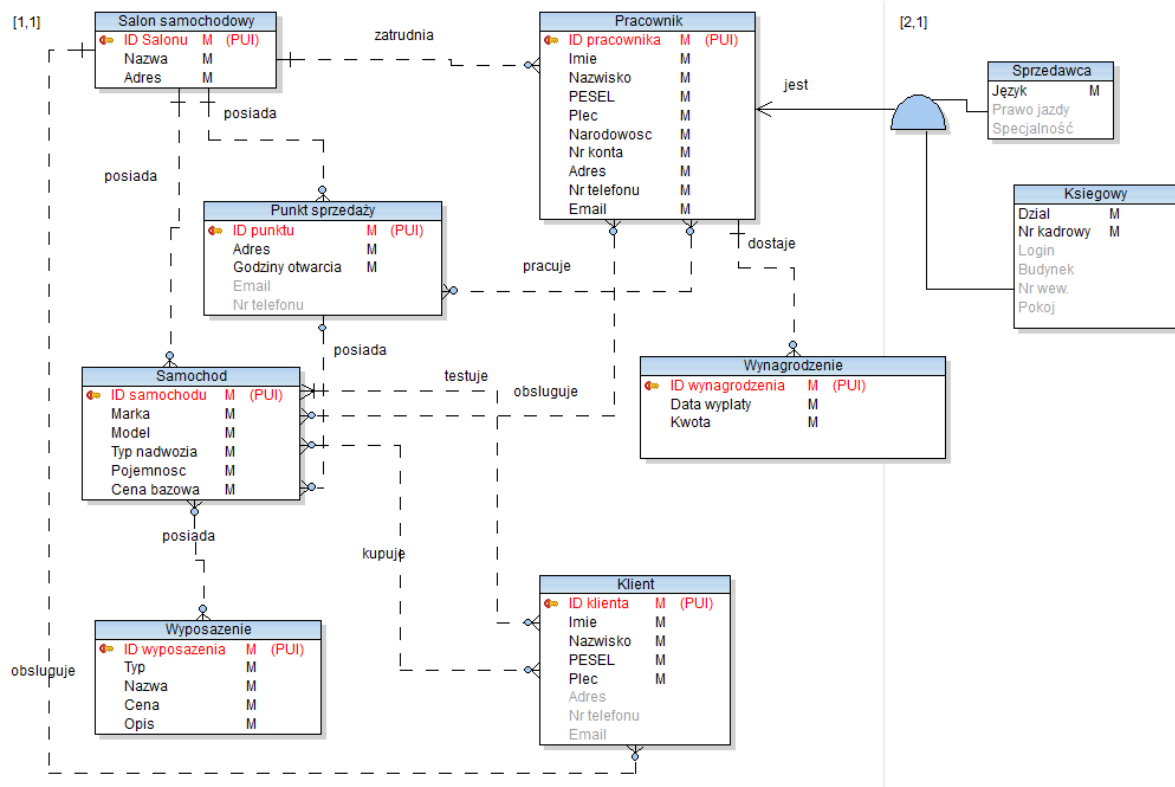
Prawo JazdyD – Przyjmuje wartości 'AM', 'A1', 'A2', 'A', 'B1', 'B', 'B+E', 'C', 'C1', 'C1+E', 'C+E', 'D', 'D1', 'D1+E', 'D+E', 'T'. Definiuje Kategorię prawa jazdy, wykorzystywaną w encji Sprzedawca.

Atrybuty typ nadwozia, marka, model oraz poziom języka również będą miały określone dziedziny. Zostało to określone na poziomie modelu logicznego. Rozważono dwa rozwiązania: encje słownikowe oraz nowe dziedziny (domeny).

3.5. Klucze kandydujące i główne

Nazwa encji	Klucz główny	Klucz potencjalny
Klient	KlientPK	PESEL
Pracownik	PracownikPK	PESEL
Punkt_sprzedazy	PunktPK	Adres
Salon_samochodowy	SalonPK	
Samochod	SamochodPK	
Wynagrodzenie	WynagrodzeniePK	
Wyposazenie	WyposazeniePK	Nazwa

3.6. Schemat ER na poziomie konceptualnym

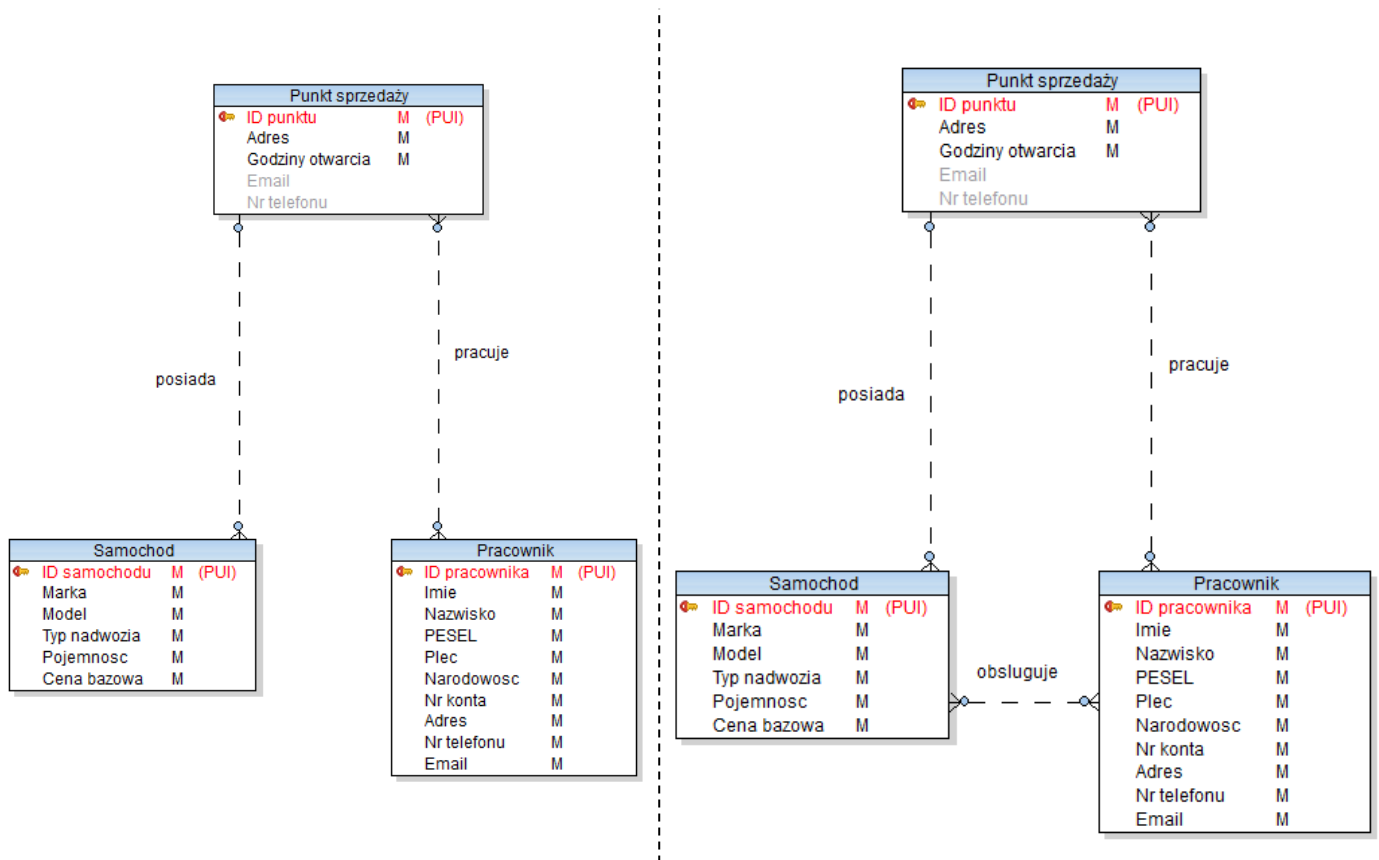


3.7. Problem pułapek szczelinowych i wachlarzowych

W stworzonym modelu nie występowały pułapki wachlarzowe i szczelinowe. W trakcie tworzenia modelu, zauważono jednak, że podjęcie innych decyzji projektowych, spowodowałoby pojawienie się takich problemów. Poniżej przedstawiono problemy, których udało się uniknąć poprzez podjęcie odpowiednich decyzji:

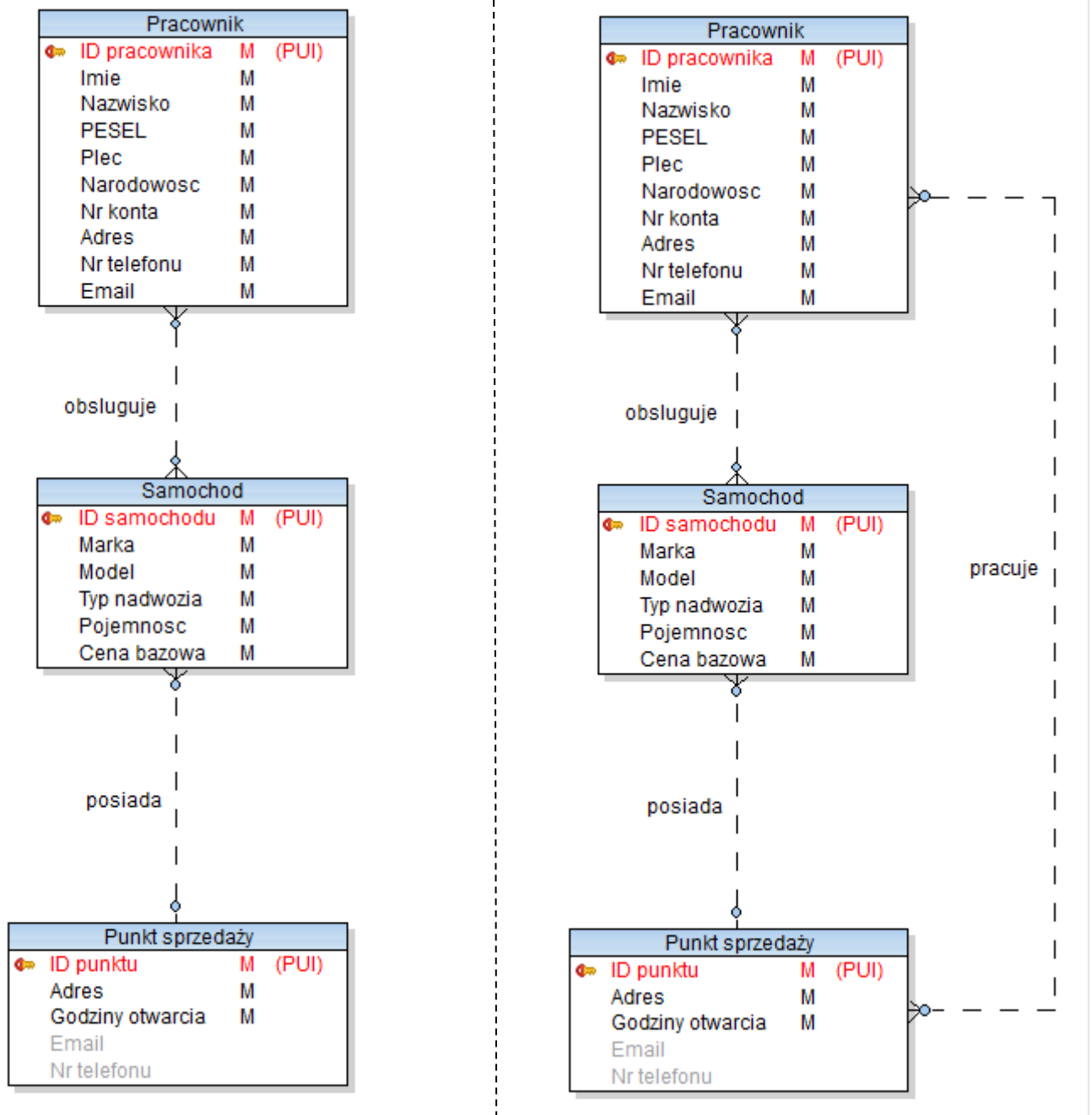
Potencjalna pułapka wachlarzowa: Brak związku Pracownik-Samochód.

W jednej z wersji roboczych modelu, okazało się, że przy połączeniu tabel Samochód – Punkt_sprzedaży - Pracownik nie ma możliwości określenia który pracownik obsługuje, który samochód. Jest to konieczne z punktu widzenia planu biznesowego – trzeba mieć możliwość określenia, który pracownik dokonał transakcji sprzedaży danego samochodu. Problem rozwiązano poprzez stworzeniu dodatkowego związku Samochód-Pracownik rozwiązując ten problem.



Potencjalna pułapka szczelinowa: Brak związku Pracownik-Punkt sprzedaży.

Kontynuując analizę sytuacji Samochód – Punkt_sprzedaży – Pracownik rozważono czy konieczny jest związek Pracownik – Punkt sprzedaży. Okazało się jednak, że powstaje wtedy pułapka szczelinowa - nie ma możliwości, aby pracownik pracował w punkcie sprzedaży bez obsługiwanego samochodu. Z tego powodu konieczne było wprowadzenie powyższego związku.



4. Model logiczny

Z modelu konceptualnego jest możliwość przejścia do dowolnego modelu logicznego bazy danych. Zgodnie z założeniami projektu, dokonano konwersji do modelu relacyjnego.

4.1. Charakterystyka modelu relacyjnego

Zgodnie z założeniami projektowymi, model na poziomie logicznym powinien spełniać założenia modelu bazy relacyjnej oraz być w przynajmniej zgodny z 3PN, a co za tym idzie 2PN oraz 1PN.

Model relacyjny charakteryzuje się:

- Odmiennością nazw relacji
- Odmiennością nazw atrybutów w obrębie danej relacji
- Ustaloną dziedziną atrybutów
- Odmiennością krotek w obrębie relacji
- Posiadaniem klucza (klucze główne i klucze obce) przez każdą relację
- Brakiem związków wiele do wielu

4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym

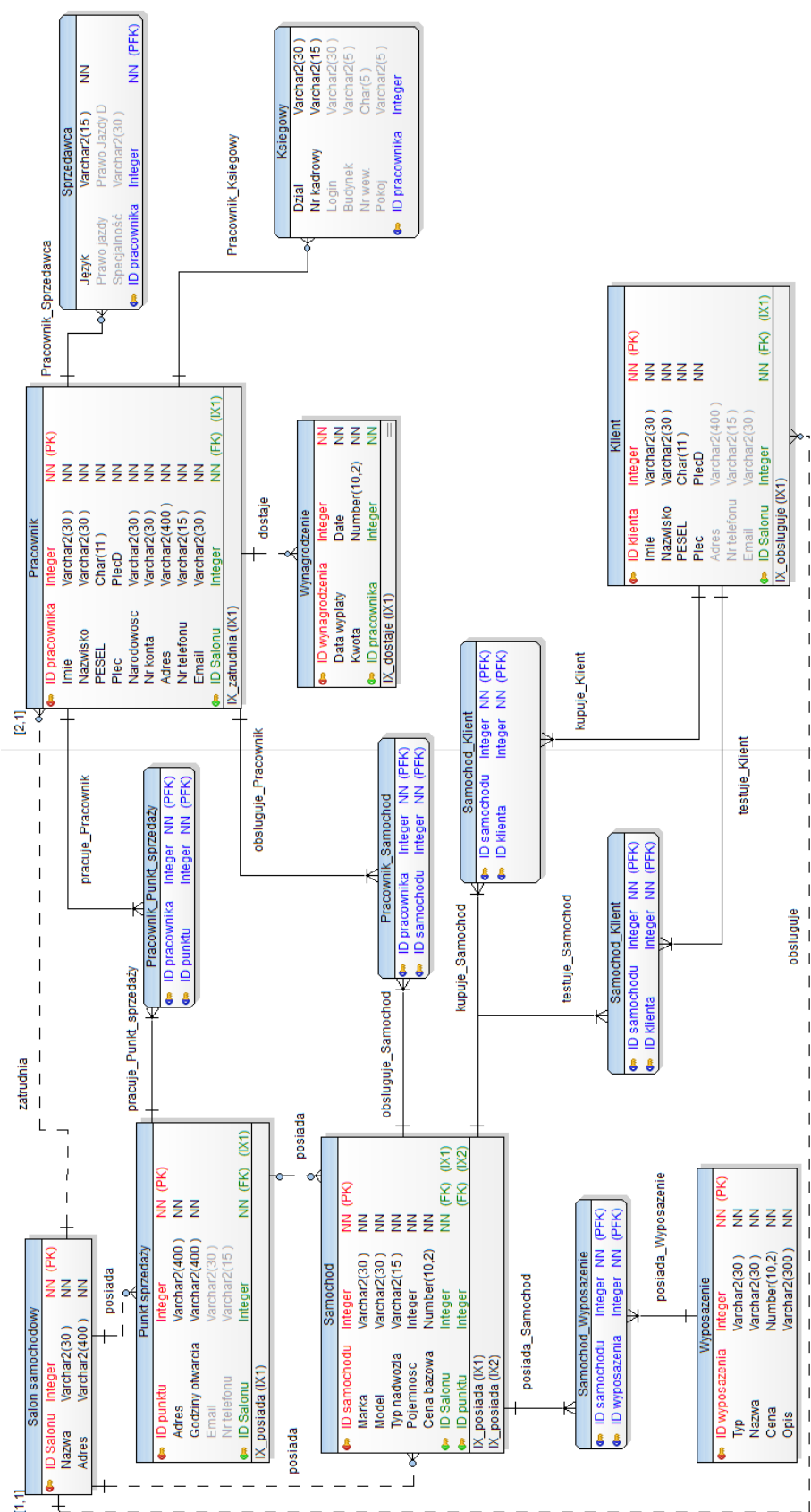
Już na poziomie konwersji przy użyciu narzędzie *Toad Data Modeler* zostały usunięte związki *wiele-do-wielu*. Zostały wygenerowane automatycznie tzw. Tablice brydżujące. Należało je zweryfikować – nadać odpowiednią nazwę, ustalić klucze oraz zastanowić się nad ewentualnymi dodatkowymi argumentami w obrębie tablic. Np. w przypadku tablicy brydżującej „Jazda testowa” dodano atrybuty – *Cena* i *Data*. Trzeba było zmienić obowiązkowość oraz krotność związków łączących tablice brydżujące.

Ta sama konwersja spowodowała zamianę specjalizacji encji na związki *jeden-do-wielu* z wprowadzeniem klucza obcego *ID pracownika* do encji „Sprzedawcy” i „Księgowi”.

Konwersja do modelu logicznego spowodowała utworzenie encji odpowiadającym encjom z modelu konceptualnego zmieniono nazwy ich na liczbę mnogą.

Model logiczny zaraz po wygenerowaniu bez wprowadzanie powyższych zmian oraz normalizacji wyglądał następująco:

4.2.1. Schemat po konwersji – przed procesem normalizacji



4.3. Proces normalizacji – analiza i przykłady

Aby osiągnąć 1PN przeprowadzono następujące operacje:

- Zmodyfikowano pola wielowartościowe w celu uzyskania pól z wartościami atomowymi, zależnymi tylko od klucza i od całości klucza. Np. w encji „Salon samochodowy” pole *adres* (wielowartościowe) zamieniono na pola: *Miasto*, *Ulica*, *Kod pocztowy*, *Numer domu*, *Numer mieszkania*.
- Zmodyfikowano pola segmentowe. Np. w encji „Sprzedawca” pole *język* (segmentowe) zostało zrealizowane przez utworzenie dwóch encji – „Język” (encja słownikowa) oraz „Znajomość języka” – tablica brydząca między „Sprzedawca” i „Język”, zawierająca dodatkowo atrybut informujący o poziomie znajomości języka.
- Stworzone zostały encje słownikowe np. encje „Model” i „Marka”. Nie zdecydowano się na nową dziedzinę, ponieważ nie jest to zbiór skończony i w trakcie wykorzystywania bazy danych mogą być dodawane nowe wartości do zbioru. W przypadku argumentów *typ nadwozia* oraz *poziom języka* zdecydowano się na stworzenie dziedziny.

2 PN wymaga, aby model był w 1 PN oraz aby każdy atrybut tej relacji nie wchodzący w skład żadnego klucza potencjalnego był w pełni funkcyjnie zależny od wszystkich kluczy potencjalnych tej relacji.

- Nasz model charakteryzuje się tym, że każda relacja posiada klucz prosty – wymaganie 2 PN jest spełnione automatycznie

3 PN wymaga, aby model był w 2 PN oraz aby każdy atrybut tej relacji nie wchodzący w skład żadnego klucza potencjalnego nie jest przechodnio funkcyjnie zależny od żadnego klucza potencjalnego tej relacji.

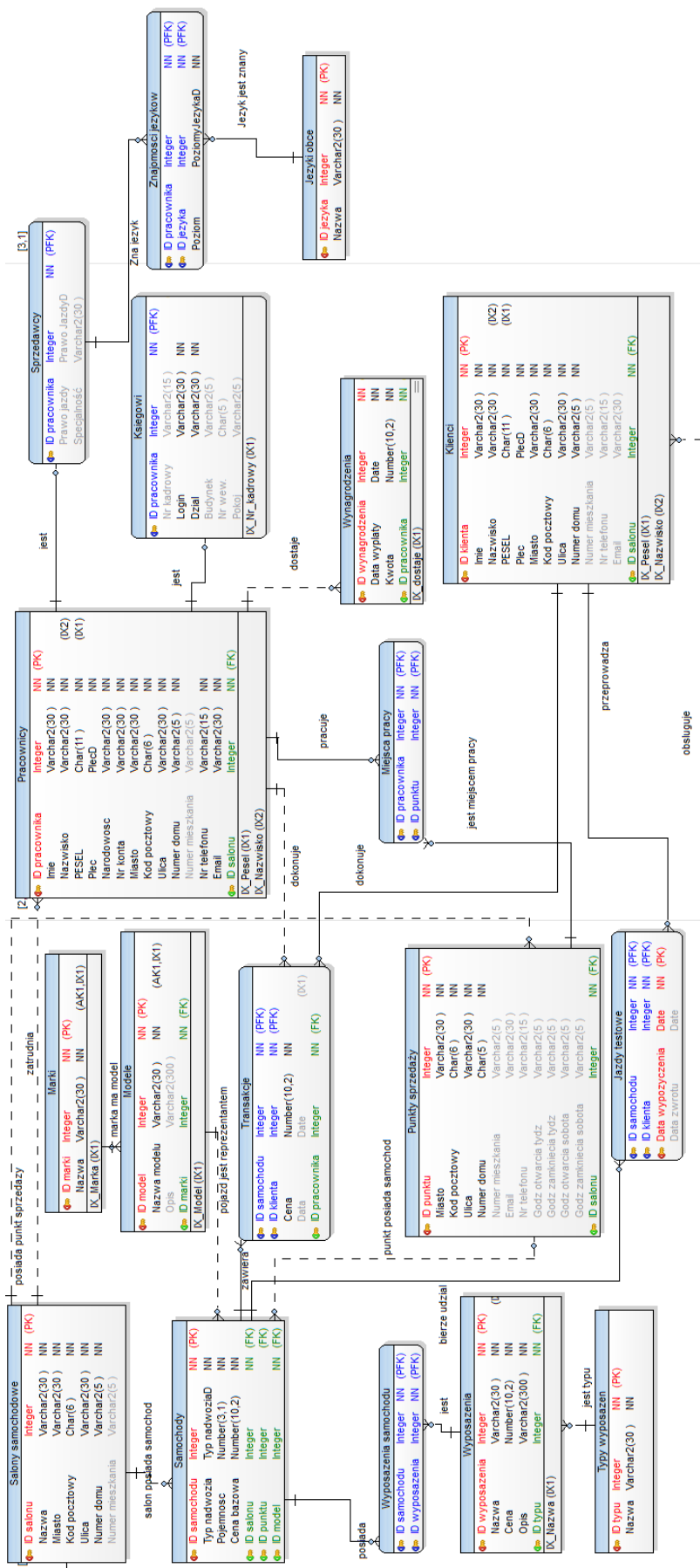
- Nasz model spełnia te wymagania – został doprowadzony do 2 PN. Nie występują w nim pola wyliczalne, a wszystkie klucze są proste.

Tak na przykład encja „Samochody” przed normalizacją (po lewej) i po normalizacji (po prawej):

Samochod		
🔑 ID samochodu	Integer	NN (PK)
Marka	Varchar2(30)	NN
Model	Varchar2(30)	NN
Typ nadwozia	Varchar2(15)	NN
Pojemnosc	Integer	NN
Cena bazowa	Number(10,2)	NN
🔑 ID Salonu	Integer	NN (FK) (IX1)
🔑 ID punktu	Integer	(FK) (IX2)

Samochody		
🔑 ID samochodu	Integer	NN (PK)
Typ nadwozia	Typ nadwoziaD	NN
Pojemnosc	Number(3,1)	NN
Cena bazowa	Number(10,2)	NN
🔑 ID salonu	Integer	NN (FK)
🔑 ID punktu	Integer	(FK)
🔑 ID model	Integer	NN (FK)

4.4. Schemat ER na poziomie modelu logicznego



4.5. Więzy integralności

Stworzono oddzielne klucze dla każdej encji, aby zagwarantować integralność danych. Wszystkie pola niezbędne do prawidłowego działania bazy danych zostały ustawione jako NOT NULL.

Stworzono tabelę Transakcja zamiast dwóch tabel brydżujących – Pracownik-Samochód oraz Klient-Samochód. Tym sposobem dane dotyczące zakupu samochodu obsługiwane przez danego pracownika i kupionego przez klienta przechowywane są w jednym miejscu. Podczas aktualizacji, zmiany przeprowadzane są tylko tam.

Do prezentacji atrybutów poszczególnych tabel w modelu logicznym zostały użyte następujące skróty:

- N (Not Null) – czy pole musi przyjmować jakąś wartość,
- U (Unique) – czy unikatowe wartości atrybutu
- C (Check) – czy wartość pola ograniczona w zakresie dziedziny,
- D (Default) – czy pole ma domyślną wartość

Dodatkowo w kolumnie „Klucz” zostały użyte następujące skróty:

- PK – atrybut należy do klucza głównego,
- FK – atrybut jest kluczem obcym,
- PFK – atrybut należy do klucza głównego oraz jest kluczem obcym

Encja „Salony Samochodowe”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PK	ID_salonu		Integer	Tak	Nie	Nie	Nie	
	Nazwa		Varchar2(30)	Tak	Nie	Nie	Nie	
	Miasto		Varchar2(30)	Tak	Nie	Nie	Nie	
	Kod_pocztowy		Char(6)	Tak	Nie	Nie	Nie	
	Ulica		Varchar2(30)	Tak	Nie	Nie	Nie	
	Numer_domu		Varchar2(5)	Tak	Nie	Nie	Nie	
	Numer_mieszkania		Varchar2(5)	Nie	Nie	Nie	Nie	

Encja „Punkty sprzedaży”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PK	ID_punktu		Integer	Tak	Nie	Nie	Nie	
	Miasto		Varchar2(30)	Tak	Nie	Nie	Nie	
	Kod_pocztowy		Char(6)	Tak	Nie	Nie	Nie	
	Ulica		Varchar2(30)	Tak	Nie	Nie	Nie	
	Numer_domu		Char(5)	Tak	Nie	Nie	Nie	
	Numer_mieszkania		Varchar2(5)	Nie	Nie	Nie	Nie	

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
	Email		Varchar2(30)	Nie	Nie	Nie	Nie	
	Nr_telefonu		Varchar2(15)	Nie	Nie	Nie	Nie	
	Godz_otwarcia_tydz		Varchar2(5)	Nie	Nie	Nie	Nie	
	Godz_zamknienia_tyd z		Varchar2(5)	Nie	Nie	Nie	Nie	
	Godz_otwarcia_sobota		Varchar2(5)	Nie	Nie	Nie	Nie	
	Godz_zamknienia_sob ota		Varchar2(5)	Nie	Nie	Nie	Nie	
FK	ID_Salonu		Integer	Tak	Nie	Nie	Nie	

Encja „Klienci”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PK	ID_klienta		Integer	Tak	Nie	Nie	Nie	
	Imie		Varchar2(30)	Tak	Nie	Nie	Nie	
	Nazwisko		Varchar2(30)	Tak	Nie	Nie	Nie	
	PESEL		Char(11)	Tak	Nie	Nie	Nie	
	Plec	PlecD	Char(1)	Tak	Nie	Nie	Nie	
	Miasto		Varchar2(30)	Tak	Nie	Nie	Nie	
	Kod_pocztowy		Char(6)	Tak	Nie	Nie	Nie	
	Ulica		Varchar2(30)	Tak	Nie	Nie	Nie	
	Numer_domu		Varchar2(5)	Tak	Nie	Nie	Nie	
	Numer_mieszkania		Varchar2(5)	Nie	Nie	Nie	Nie	
	Nr_telefonu		Varchar2(15)	Nie	Nie	Nie	Nie	
	Email		Varchar2(30)	Nie	Nie	Nie	Nie	
FK	ID_Salonu		Integer	Tak	Nie	Nie	Nie	

Encja „Pracownicy”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PK	ID_pracownika		Integer	Tak	Nie	Nie	Nie	
	Imie		Varchar2(30)	Tak	Nie	Nie	Nie	
	Nazwisko		Varchar2(30)	Tak	Nie	Nie	Nie	
	PESEL		Char(11)	Tak	Nie	Nie	Nie	
	Plec	PlecD	Char(1)	Tak	Nie	Nie	Nie	
	Narodowosc		Varchar2(30)	Tak	Nie	Nie	Nie	

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
	Nr_konta		Varchar2(30)	Tak	Nie	Nie	Nie	
	Miasto		Varchar2(30)	Tak	Nie	Nie	Nie	
	Kod_pocztowy		Char(6)	Tak	Nie	Nie	Nie	
	Ulica		Varchar2(30)	Tak	Nie	Nie	Nie	
	Numer_domu		Varchar2(5)	Tak	Nie	Nie	Nie	
	Numer_mieszkania		Varchar2(5)	Nie	Nie	Nie	Nie	
	Nr_telefonu		Varchar2(15)	Tak	Nie	Nie	Nie	
	Email		Varchar2(30)	Tak	Nie	Nie	Nie	
FK	ID_Salonu		Integer	Tak	Nie	Nie	Nie	

Encja „Miejsca_pracy”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PFK	ID_pracownika		Integer	Tak	Nie	Nie	Nie	
PFK	ID_punktu		Integer	Tak	Nie	Nie	Nie	

Encja „Księgowi”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PFK	ID_pracownika		Integer	Tak	Nie	Nie	Nie	
	Nr_kadrowy		Varchar2(15)	Nie	Nie	Nie	Nie	
	Login		Varchar2(30)	Tak	Nie	Nie	Nie	
	Dzial		Varchar2(30)	Tak	Nie	Nie	Nie	
	Budynek		Varchar2(5)	Nie	Nie	Nie	Nie	
	Nr_wew		Char(5)	Nie	Nie	Nie	Nie	
	Pokoj		Varchar2(5)	Nie	Nie	Nie	Nie	

Encja „Sprzedawcy”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PFK	ID_pracownika		Integer	Tak	Nie	Nie	Nie	
	Prawo_jazdy	Prawo_JazdyD	Varchar2(4)	Nie	Nie	Nie	Nie	
	Specjalnosc		Varchar2(30)	Nie	Nie	Nie	Nie	

Encja „Języki obce”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PK	ID_jezyka		Integer	Tak	Nie	Nie	Nie	
	Nazwa		Varchar2(30)	Tak	Nie	Nie	Nie	

Encja „Znajomosci_jezykow”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PFK	ID_pracownika		Integer	Tak	Nie	Nie	Nie	
PFK	ID_jezyka		Integer	Tak	Nie	Nie	Nie	
	Poziom	PoziomyJezykaD	Varchar2(5)	Tak	Nie	Nie	Nie	

Encja „Wynagrodzenia”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PK	ID_wynagrodzenia		Integer	Tak	Nie	Nie	Nie	
	Data_wyplaty		Date	Tak	Nie	Nie	Nie	
	Kwota		Number(10,2)	Tak	Nie	Nie	Nie	
FK	ID_pracownika		Integer	Tak	Nie	Nie	Nie	

Encja „Samochody”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PK	ID_samochodu		Integer	Tak	Nie	Nie	Nie	
	Typ_nadwozia	Typ_nadwoziaD	Varchar2(30)	Tak	Nie	Tak	Nie	
	Pojemnosc		Number(3,1)	Tak	Nie	Nie	Nie	
	Cena_bazowa		Number(10,2)	Tak	Nie	Nie	Nie	
FK	ID_Salonu		Integer	Tak	Nie	Nie	Nie	
FK	ID_punktu		Integer	Nie	Nie	Nie	Nie	
FK	Id_model		Integer	Tak	Nie	Nie	Nie	

Encja „Jazdy testowe”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PFK	ID_samochodu		Integer	Tak	Nie	Nie	Nie	
PFK	ID_klienta		Integer	Tak	Nie	Nie	Nie	

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PK	Data_wypozyczenia		Date	Tak	Nie	Nie	Nie	
	Data_zwrotu		Date	Nie	Nie	Nie	Nie	

Encja „Marki”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PK	ID_marki		Integer	Tak	Nie	Nie	Nie	
	Nazwa		Varchar2(30)	Tak	Tak	Nie	Nie	

Encja „Modele”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PK	ID_model		Integer	Tak	Nie	Nie	Nie	
	Nazwa_modelu		Varchar2(30)	Tak	Tak	Nie	Nie	
	Opis		Varchar2(300)	Nie	Nie	Nie	Nie	
FK	ID_marki		Integer	Tak	Nie	Nie	Nie	

Encja „Transakcje”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PFK	ID_samochodu		Integer	Tak	Nie	Nie	Nie	
PFK	ID_klienta		Integer	Tak	Nie	Nie	Nie	
	Cena		Number(10,2)	Tak	Nie	Nie	Nie	
	Data		Date	Nie	Nie	Nie	Nie	
FK	ID_pracownika		Integer	Tak	Nie	Nie	Nie	

Encja „Wypożyczenia”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PK	ID_wypozyczenia		Integer	Tak	Nie	Nie	Nie	
	Nazwa		Varchar2(30)	Tak	Nie	Nie	Nie	
	Cena		Number(10,2)	Tak	Nie	Nie	Nie	
	Opis		Varchar2(300)	Tak	Nie	Nie	Nie	
FK	ID_typu		Integer	Tak	Nie	Nie	Nie	

Encja „Wypozazenia_samochodu”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PFK	ID_samochodu		Integer	Tak	Nie	Nie	Nie	
PFK	ID_wyposazenia		Integer	Tak	Nie	Nie	Nie	

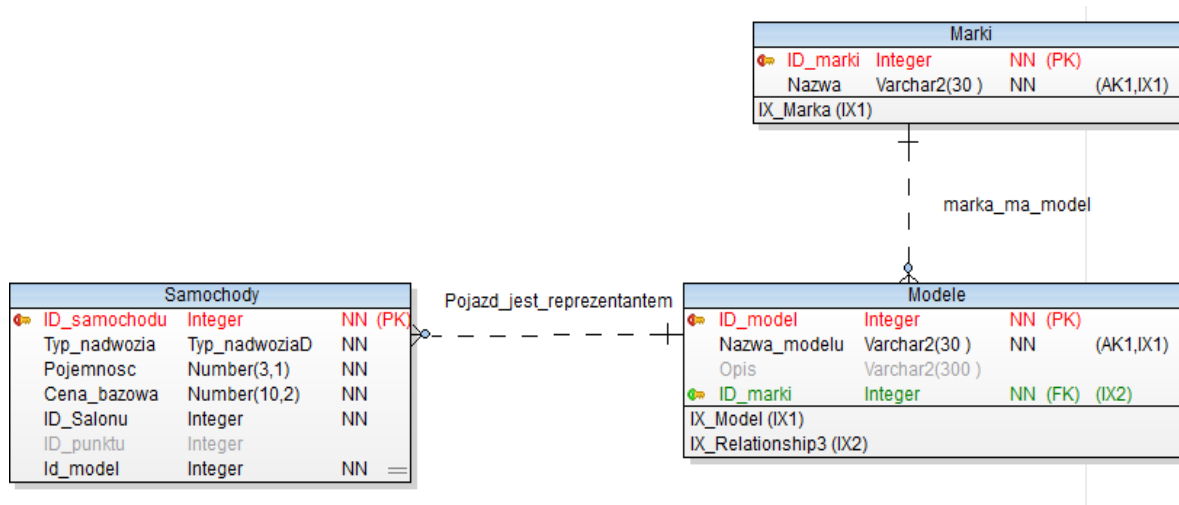
Encja „Typy wyposazeń”

Klucz	Nazwa	Domena	Typ danych	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Komentarze
PK	ID_typu		Integer	Tak	Nie	Nie	Nie	
	Nazwa		Varchar2(30)	Tak	Nie	Nie	Nie	

4.6. Proces denormalizacji – analiza i przykłady

W procesie denormalizacji, aby przyspieszyć wyszukiwanie w bazie można połączyć encje Samochody, Modele oraz Marki. Nie będzie potrzebne wtedy używania funkcji JOIN, aby wyszukać samochody konkretnej marki czy modelu.

Tak na przykład przed denormalizacją encje Samochody, Modele i Marki



Po denormalizacji zamieniły by się w jedną encję Samochody

Samochody		
ID_samochodu	Integer	NN (PK)
Typ_nadwozia	Typ_nadwoziaD	NN
Pojemnosc	Number(3,1)	NN
Cena_bazowa	Number(10,2)	NN
Nazwa_modelu	Varchar2(30)	NN
Nazwa_marki	Varchar2(30)	NN
Opis_modelu	Varchar2(300)	
ID_Salonu	Integer	NN
ID_punktu	Integer	

5. Faza fizyczna

5.1. Projekt transakcji i weryfikacja ich wykonalności

Transakcja	Potrzebne encje	Czy wykonalne?
Podgląd/modyfikacja danych personalnych pracowników	Pracownicy (ewentualnie Księgowi oraz Sprzedawcy)	Tak
Podgląd/modyfikacja znajomości języków obcych pracownika	Pracownicy, Sprzedawcy, Znajomosci jezykow, Jezyki obce	Tak
Podgląd/modyfikacja przeszłych wynagrodzeń pracownika	Pracownicy, Wynagrodzenia	Tak
Podgląd/modyfikacja pracowników pracujących w danym punkcie sprzedaży	Pracownicy, Miejsce pracy, Punkty sprzedaży	Tak
Podgląd/modyfikacja przeprowadzonych transakcji w danym miesiącu	Transakcje, Samochody, Klienci, Pracownicy	Tak
Podgląd/modyfikacja danych personalnych Klientów	Klienci	Tak
Podgląd/modyfikacja przeprowadzonych jazd testowych w danym miesiącu	Klienci, Jazdy testowe	Tak
Podgląd/modyfikacja samochodów danej marki	Samochody, Modele, Marki	Tak
Podgląd/modyfikacja wyposażenia konkretnego samochodu	Samochody, Wyposażenia samochodu, Wyposażenia, Typy wyposażen, Modele, Marki	Tak

5.2. Strojenie bazy danych – dobór indeksów

Indeksy założono na polach, które będą najczęściej wyszukiwane, czyli np:

- Pole 'nazwa' w encji Wyposazenia
- Pole 'nazwa modelu' w encji Modele.

5.3. Skrypt SQL zakładający bazę

```
/*
Created: 05/12/2019
Modified: 04/01/2020
Model: Salon Samochodowy LM
Database: Oracle 19c
*/

-- Create tables section -----
----

-- Table Punkty_sprzedazy

CREATE TABLE Punkty_sprzedazy(
  ID_punktu Integer NOT NULL,
  Miasto Varchar2(30 ) NOT NULL,
  Kod_pocztowy Char(6 ) NOT NULL,
  Ulica Varchar2(30 ) NOT NULL,
  Numer_domu Char(5 ) NOT NULL,
  Numer_mieszkania Varchar2(5 ),
  Email Varchar2(30 ),
  Nr_telefonu Varchar2(15 ),
  Godz_otwarcia_tydz Varchar2(5 ),
  Godz_zamknienia_tydz Varchar2(5 ),
  Godz_otwarcia_sobota Varchar2(5 ),
  Godz_zamknienia_sobota Varchar2(5 ),
  ID_Salonu Integer NOT NULL
)
/

-- Add keys for table Punkty_sprzedazy

ALTER TABLE Punkty_sprzedazy ADD CONSTRAINT PunktPK PRIMARY KEY
(ID_punktu)
/

-- Table Wyposazenia

CREATE TABLE Wyposazenia(
  ID_wyposazenia Integer NOT NULL,
  Nazwa Varchar2(30 ) NOT NULL,
  Cena Number(10,2) NOT NULL,
  Opis Varchar2(300 ) NOT NULL,
  ID_typu Integer NOT NULL
)
/

-- Create indexes for table Wyposazenia

CREATE INDEX IX_Nazwa ON Wyposazenia (Nazwa)
```

```

/

-- Add keys for table Wyposazenia

ALTER TABLE Wyposazenia ADD CONSTRAINT WyposazeniePK PRIMARY KEY
(ID_wyposazenia)
/

-- Table Samochody

CREATE TABLE Samochody(
    ID_samochodu Integer NOT NULL,
    Typ_nadwozia Varchar2(30 ) NOT NULL
CHECK (Typ_nadwozia IN ('SUV', 'Coupe', 'Hatchback', 'Combi',
'Sedan', 'Pickup', 'Minivan', 'Cabriolet',
'Universal')),

    Pojemnosc Number(3,1) NOT NULL,
    Cena_bazowa Number(10,2) NOT NULL,
    ID_Salonu Integer NOT NULL,
    ID_punktu Integer,
    Id_model Integer NOT NULL
)
/

-- Add keys for table Samochody

ALTER TABLE Samochody ADD CONSTRAINT SamochodPK PRIMARY KEY
(ID_samochodu)
/

-- Table Wynagrodzenia

CREATE TABLE Wynagrodzenia(
    ID_wynagrodzenia Integer NOT NULL,
    Data_wypłaty Date NOT NULL,
    Kwota Number(10,2) NOT NULL,
    ID_pracownika Integer NOT NULL
)
/

-- Create indexes for table Wynagrodzenia

CREATE INDEX IX_dostaje ON Wynagrodzenia (ID_pracownika)
/

-- Add keys for table Wynagrodzenia

ALTER TABLE Wynagrodzenia ADD CONSTRAINT WynagrodzeniePK PRIMARY KEY
(ID_wynagrodzenia)
/

-- Table Ksiegowi

CREATE TABLE Ksiegowi(
    ID_pracownika Integer NOT NULL,
    Nr_kadrowy Varchar2(15 ),
    Login Varchar2(30 ) NOT NULL,

```

```

    Dzial Varchar2(30 ) NOT NULL,
    Budynek Varchar2(5 ),
    Nr_wew Char(5 ),
    Pokoj Varchar2(5 )
)
/

-- Create indexes for table Ksiegowi

CREATE INDEX IX_Nr_kadrowy ON Ksiegowi (Nr_kadrowy)
/

-- Add keys for table Ksiegowi

ALTER TABLE Ksiegowi ADD CONSTRAINT KsiegowiPK PRIMARY KEY
(ID_pracownika)
/

-- Table Sprzedawcy

CREATE TABLE Sprzedawcy(
    ID_pracownika Integer NOT NULL,
    Prawo_jazdy Varchar2(4 )
        CHECK (Prawo_jazdy IN ('AM', 'A1', 'A2', 'A', 'B1', 'B',
'B+E', 'C', 'C1', 'C1+E', 'C+E', 'D', 'D1', 'D1+E', 'D+E', 'T')),
    Specjalnosc Varchar2(30 )
)
/

-- Add keys for table Sprzedawcy

ALTER TABLE Sprzedawcy ADD CONSTRAINT SprzedawcyPK PRIMARY KEY
(ID_pracownika)
/

-- Table Pracownicy

CREATE TABLE Pracownicy(
    ID_pracownika Integer NOT NULL,
    Imie Varchar2(30 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    PESEL Char(11 ) NOT NULL,
    Plec Char(1 ) NOT NULL
        CHECK (Plec IN ('M', 'K')),
    Narodowosc Varchar2(30 ) NOT NULL,
    Nr_konta Varchar2(30 ) NOT NULL,
    Miasto Varchar2(30 ) NOT NULL,
    Kod_pocztowy Char(6 ) NOT NULL,
    Ulica Varchar2(30 ) NOT NULL,
    Numer_domu Varchar2(5 ) NOT NULL,
    Numer_mieszkania Varchar2(5 ),
    Nr_telefonu Varchar2(15 ) NOT NULL,
    Email Varchar2(30 ) NOT NULL,
    ID_Salonu Integer NOT NULL
)
/

```

```

-- Create indexes for table Pracownicy

CREATE INDEX IX_Pesel_Pracownik ON Pracownicy (PESEL)
/

CREATE INDEX IX_Nazwisko_Pracownik ON Pracownicy (Nazwisko)
/

-- Add keys for table Pracownicy

ALTER TABLE Pracownicy ADD CONSTRAINT PracownikPK PRIMARY KEY
(ID_pracownika)
/

-- Table Salony_samochodowe

CREATE TABLE Salony_samochodowe(
    ID_salonu Integer NOT NULL,
    Nazwa Varchar2(30 ) NOT NULL,
    Miasto Varchar2(30 ) NOT NULL,
    Kod_pocztowy Char(6 ) NOT NULL,
    Ulica Varchar2(30 ) NOT NULL,
    Numer_domu Varchar2(5 ) NOT NULL,
    Numer_mieszkania Varchar2(5 )
)
/

-- Add keys for table Salony_samochodowe

ALTER TABLE Salony_samochodowe ADD CONSTRAINT SalonPK PRIMARY KEY
(ID_salonu)
/

-- Table Klienci

CREATE TABLE Klienci(
    ID_klienta Integer NOT NULL,
    Imie Varchar2(30 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    PESEL Char(11 ) NOT NULL,
    Plec Char(1 ) NOT NULL
        CHECK (Plec IN ('M', 'K')),
    Miasto Varchar2(30 ) NOT NULL,
    Kod_pocztowy Char(6 ) NOT NULL,
    Ulica Varchar2(30 ) NOT NULL,
    Numer_domu Varchar2(5 ) NOT NULL,
    Numer_mieszkania Varchar2(5 ),
    Nr_telefonu Varchar2(15 ),
    Email Varchar2(30 ),
    ID_Salonu Integer NOT NULL
)
/

-- Create indexes for table Klienci

CREATE INDEX IX_Pesel_Klient ON Klienci (PESEL)
/

```

```

CREATE INDEX IX_Nazwisko_Klient ON Klienci (Nazwisko)
/

-- Add keys for table Klienci

ALTER TABLE Klienci ADD CONSTRAINT KlientPK PRIMARY KEY (ID_klienta)
/

-- Table Miejsca_pracy

CREATE TABLE Miejsca_pracy(
    ID_pracownika Integer NOT NULL,
    ID_punktu Integer NOT NULL
)
/

-- Table Wyposazenia_samochodu

CREATE TABLE Wyposazenia_samochodu(
    ID_samochodu Integer NOT NULL,
    ID_wyposazenia Integer NOT NULL
)
/

-- Table Transakcje

CREATE TABLE Transakcje(
    ID_samochodu Integer NOT NULL,
    ID_klienta Integer NOT NULL,
    Cena Number(10,2) NOT NULL,
    Data Date,
    ID_pracownika Integer NOT NULL
)
/

-- Create indexes for table Transakcje

CREATE INDEX IX_Data ON Transakcje (Data)
/

-- Table Jazdy_testowe

CREATE TABLE Jazdy_testowe(
    ID_samochodu Integer NOT NULL,
    ID_klienta Integer NOT NULL,
    Data_wypozyczenia Date NOT NULL,
    Data_zwrotu Date
)
/

-- Table Jezyki_obce

CREATE TABLE Jezyki_obce(
    ID_jezyka Integer NOT NULL,
    Nazwa Varchar2(30 ) NOT NULL
)
/

```

```

-- Add keys for table Jezyki_obce

ALTER TABLE Jezyki_obce ADD CONSTRAINT Jezyk_obcyPK PRIMARY KEY
(ID_jezyka)
/

-- Table Znajomosci_jezykow

CREATE TABLE Znajomosci_jezykow(
    ID_pracownika Integer NOT NULL,
    ID_jezyka Integer NOT NULL,
    Poziom Varchar2(5 ) NOT NULL
        CHECK (Poziom IN ('A1', 'A2','B1', 'B2','C1', 'C2', 'Native'))
)
/

-- Add keys for table Znajomosci_jezykow

ALTER TABLE Znajomosci_jezykow ADD CONSTRAINT PK_Znajomosci_jezykow
PRIMARY KEY (ID_pracownika,ID_jezyka)
/

-- Table Marki

CREATE TABLE Marki(
    ID_marki Integer NOT NULL,
    Nazwa Varchar2(30 ) NOT NULL
)
/

-- Create indexes for table Marki

CREATE INDEX IX_Marka ON Marki (Nazwa)
/

-- Add keys for table Marki

ALTER TABLE Marki ADD CONSTRAINT PK_Marki PRIMARY KEY (ID_marki)
/

ALTER TABLE Marki ADD CONSTRAINT Nazwa UNIQUE (Nazwa)
/

-- Table Modele

CREATE TABLE Modele(
    ID_model Integer NOT NULL,
    Nazwa_modelu Varchar2(30 ) NOT NULL,
    Opis Varchar2(300 ),
    ID_marki Integer NOT NULL
)
/

-- Create indexes for table Modele

CREATE INDEX IX_Model ON Modele (Nazwa_modelu)
/

```

```

-- Add keys for table Modele

ALTER TABLE Modele ADD CONSTRAINT PK_Modele PRIMARY KEY (ID_model)
/

ALTER TABLE Modele ADD CONSTRAINT Nazwa_modelu UNIQUE (Nazwa_modelu)
/

-- Table Typy_wyposazen

CREATE TABLE Typy_wyposazen(
    ID_typu Integer NOT NULL,
    Nazwa Varchar2(30 ) NOT NULL
)
/

-- Add keys for table Typy_wyposazen

ALTER TABLE Typy_wyposazen ADD CONSTRAINT PK_Typy_wyposazen PRIMARY
KEY (ID_typu)
/

-- Create foreign keys (relationships) section -----
-----

ALTER TABLE Pracownicy ADD CONSTRAINT zatrudnia FOREIGN KEY
(ID_Salonu) REFERENCES Salony_samochodowe (ID_salonu)
/

ALTER TABLE Samochody ADD CONSTRAINT salon_posiada_samochod FOREIGN
KEY (ID_Salonu) REFERENCES Salony_samochodowe (ID_salonu)
/

ALTER TABLE Punkty_sprzedazy ADD CONSTRAINT posiada_punkt_sprzedazy
FOREIGN KEY (ID_Salonu) REFERENCES Salony_samochodowe (ID_salonu)
/

ALTER TABLE Wynagrodzenia ADD CONSTRAINT dostaje FOREIGN KEY
(ID_pracownika) REFERENCES Pracownicy (ID_pracownika)
/

ALTER TABLE Samochody ADD CONSTRAINT punkt_posiada_samochod FOREIGN
KEY (ID_punktu) REFERENCES Punkty_sprzedazy (ID_punktu)
/

```

```

ALTER TABLE Klienci ADD CONSTRAINT obsluguje FOREIGN KEY (ID_Salonu)
REFERENCES Salony_samochodowe (ID_salonu)
/

ALTER TABLE Znajomosci_jezykow ADD CONSTRAINT Jezyk_jest_znany FOREIGN
KEY (ID_jezyka) REFERENCES Jezyki_obce (ID_jezyka)
/

ALTER TABLE Znajomosci_jezykow ADD CONSTRAINT Zna_jezyk FOREIGN KEY
(ID_pracownika) REFERENCES Sprzedawcy (ID_pracownika)
/

ALTER TABLE Transakcje ADD CONSTRAINT dokonuje FOREIGN KEY
(ID_pracownika) REFERENCES Pracownicy (ID_pracownika)
/

ALTER TABLE Modele ADD CONSTRAINT marka_ma_model FOREIGN KEY
(ID_marki) REFERENCES Marki (ID_marki)
/

ALTER TABLE Samochody ADD CONSTRAINT pojazd_jest_reprezentantem
FOREIGN KEY (Id_model) REFERENCES Modele (ID_model)
/

ALTER TABLE Wyposazenia ADD CONSTRAINT jest_typu FOREIGN KEY (ID_typu)
REFERENCES Typy_wyposazen (ID_typu)
/

```

5.4. Skrypt wpisujący przykładowe dane do bazy

```

INSERT INTO JAZDY_TESTOWE (ID_SAMOCODU, ID_KLIENTA, DATA_WYPOZYCZENIA,
DATA_ZWROTU)
VALUES (1, 1, TO_DATE('01/08/2019', 'DD/MM/YYYY'), TO_DATE('10/08/2019',
'DD/MM/YYYY'));

INSERT INTO JEZYKI_OBCE (ID_JEZYKA, NAZWA)
VALUES (1, 'ANGIELSKI');

```



```
INSERT INTO KSIEGOWI (id_pracownika, nr_kadrowy, login, dzial, budynek, nr_wew, pokoj)
VALUES (1, 'ASDF-1244-GGHJ', 'j_nowak', 'Ubezpieczenia', 'A', '01567', '519');
```

```
INSERT INTO MARKI (id_marki, NAZWA)
VALUES (1, 'Mercedes');
```

```
INSERT INTO MIEJSCA_PRACY (id_pracownika, id_punktu)
VALUES (1, 1);
```

```
INSERT INTO MODELE (id_model, nazwa_modelu, opis, id_marki)
VALUES (1, 'Mercedes-AMG GT', 'Mercedes-AMG GT 4-drzwiowe Coupé to najlepsza wersja
wygody, wszechstronności i wydajności. To także niezwykle połączenie dynamiki z zadziwiającą
efektywnością.',
1);
```

```
INSERT INTO SALONY_SAMOCODOWE (id_salonu, nazwa, miasto, kod_pocztowy, ulica,
numer_domu, numer_mieszkania)
VALUES (1, 'Salon Samochodowy', 'Warszawa', '02-210', 'Pulawska', '15', '2');
```

```
INSERT INTO PUNKTY_SPRZEDAZY (id_punktu, miasto, kod_pocztowy, ulica, numer_domu,
numer_mieszkania, email, nr_telefonu, godz_otwarcia_tydz, godz_zamknienia_tydz,
godz_otwarcia_sobota,
godz_zamknienia_sobota, id_salonu)
VALUES (1, 'Warszawa', '02-211', 'Al. Niepodlesci', '1', '1', 'email2@gmail.com', '777888999',
'09:00', '18:00', '10:00', '16:00', 1);
```

```
INSERT INTO PRACOWNICY (id_pracownika, imie, nazwisko, pesel, plec, narodowosc,
nr_konta, miasto, kod_pocztowy, ulica, numer_domu, numer_mieszkania, nr_telefonu, email,
id_salonu)
VALUES (1, 'Marek', 'Kowalski', '98765432112', 'M', 'Polska', '82102033520000130201814011',
'Warszawa', '02-211', 'Chmielna', '11', '2', '789456132', 'email@gmail.com', 1);
```

```
INSERT INTO KLIENCI (ID_KLIENTA, IMIE, NAZWISKO, PESEL, PLEC, MIASTO, KOD_POCZTOWY,
ULICA, NUMER_DOMU, numer_mieszkania, nr_telefonu, email, id_salonu)
VALUES (1, 'Jan', 'Nowak', '12345678998', 'M', 'Warszawa', '02-211', 'Nowowiejska', '10', '22',
'123456789', 'example@gmail.com', 1);
```

```
INSERT INTO SAMOCHODY (id_samochodu, typ_nadwozia, pojemnosc, cena_bazowa,
id_salonu, id_punktu, id_model)
VALUES (1, 'Cabriolet', '2.0', '160000', 1, 1, 1);
```

```
INSERT INTO SPRZEDAWCY (id_pracownika, prawo_jazdy, specjalnosc)
VALUES (1, 'B', 'Mercedes');
```

```
INSERT INTO Znajomosci_jezykow (ID_pracownika, ID_jezyka, Poziom)
VALUES (1, 1, 'B2');
```

```
INSERT INTO Wyposazenia_samochodu (ID_samochodu, ID_wyposazenia)
VALUES (1, 1);
```

```
INSERT INTO Wyposazenia( ID_wyposazenia, Nazwa, Cena, Opis, ID_typu)
```

```
VALUES (1, 'Tapicerka skórzana biała', 800.00, 'Skórzana tapicerka najbardziej efektownie  
wygląda w jasnych kolorach: białym, kremowym, jasnoszarym. Takie wnętrze auta jest  
prestiżowe, jasne, jakby prześwietlone słońcem.', 1);
```

```
INSERT INTO Wynagrodzenia(ID_wynagrodzenia, Data_wypłaty, Kwota, ID_pracownika)  
VALUES (1, TO_DATE('10/08/2019', 'DD/MM/YYYY'), 8200.00, 1);
```

```
INSERT INTO Typy_wyposazen(ID_typu, Nazwa)  
VALUES (1, 'Tapicerka');
```

```
INSERT INTO Transakcje(ID_samochodu, ID_klienta, cena, data, id_pracownika )  
VALUES (1,1, 180000.00, TO_DATE('20/10/2019', 'DD/MM/YYYY'), 1);
```

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

Podgląd znajomości języków obcych pracownika o numerze PESEL 12345678912

```
Select p.Imie, p.Nazwisko, j.Nazwa, z.Poziom from Pracownicy p  
join Sprzedawcy s on p.ID_pracownika = s.ID_pracownika  
join Znajomosci_jezykow z on s.ID_pracownika = z.ID_pracownika  
join Języki_obce j on z.ID_jezyka = j.ID_jezyka  
where p.PESEL = '12345678912';
```

Podgląd pracowników pracujących w punktach w Warszawie

```
Select p.Imie, p.Nazwisko from Pracownicy p  
join Miejsca_pracy m on p.ID_pracownika = m.ID_pracownika  
join Punkty_sprzedazy ps on m.ID_punktu = ps.ID_punktu  
where p.miasto = 'Warszawa';
```

Podgląd wszystkich samochodów marki Ford i cenie bazowej pomiędzy 150 000 – 200 000 zł w bazie

```
Select * from Samochody s  
join Modele mo on s.ID_model = mo.ID_model  
join Marki ma on mo.ID_marka = ma.ID_marka  
where ma.Nazwa = 'Mercedes' and s.Cena_bazowa between 150000 and 200000;
```

Bibliografia

- <http://jsystems.pl/blog> - blog poświęcony Oracle
- <https://www.w3schools.com> – tutorial SQL
- Wykłady przedmiotu WBD