

Józef Sławiński

Piotr Jóźwik

# Wstęp do baz danych

Hurtownia części elektrycznych

## Spis treści

1. Zakres i cel projektu .....	4
1.1. Założenia projektu .....	4
2. Definicja systemu .....	4
2.1. Perspektywy użytkowników .....	4
2.1.1. Administrator .....	4
2.1.2. Manager .....	4
2.1.3. Kadry .....	4
2.1.4. Zaopatrzeniowiec.....	4
2.1.5. Koordynator.....	4
2.1.6. Customer Service Representative.....	5
2.1.7. Klient .....	5
3. Model konceptualny .....	5
3.1. Definicja zbiorów encji określonych w projekcie oraz określenie atrybutów i ich dziedzin .....	5
3.2. Ustalenie związków między encjami i ich typów .....	9
3.3. Klucze główne i kandydujące .....	10
3.4. Schemat ER na poziomie konceptualnym.....	11
3.5 Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady .....	12
Pułapka szczelinowa .....	12
Pułapka wachlarzowa .....	12
4. Model logiczny .....	13
4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym – przykłady.....	13
4.2.1 Usunięcie związków wielu do wielu .....	13
4.2.2 Usunięcie dziedziczenia .....	14
4.3. Proces normalizacji – analiza i przykłady.....	15
4.4. Schemat ER na poziomie modelu logicznego .....	16
4.5. Więzy integralności wynikającego z modelu relacyjnego .....	17
4.6. Proces denormalizacji.....	17
5. Faza fizyczna .....	17
5.1. Projekt transakcji i weryfikacja ich wykonalności .....	17
5.2. Strojzenie bazy danych - dobór indeksów.....	18
5.3. Skrypt SQL zakładający bazę danych .....	18
5.4. Przykłady zapytań i poleceń SQL.....	35
5.4.1 Wypisanie ID_Pracownika, imienia oraz nazwisko Magazynierów którzy mają kurs na wózki widłowe .....	35
5.4.2 Wypisanie ID_Klienta, Nazwy firmy klienta oraz NIP, którzy opłacili swoje zamówienie.....	35

5.4.2 Wypisanie imienia oraz nazwiska kierowców posiadających kategorii prawa jazdy 'B' lub 'C', .....	35
5.5. Skrypt uzupełniający bazę danych .....	36

## 1. Zakres i cel projektu

Celem projektu jest zaprojektowanie relacyjnej bazy danych na poziomie konceptualnym i logicznym oraz jej fizyczna implementacja.

### 1.1. Założenia projektu

Projekt dotyczy **hurtowni części elektrycznych**. Posiada ona *klientów* którzy składają *zamówienia* zawierające *produkty*. Jest zaopatrywana poprzez *dostawy* również składające się z *produktów*. *Produkty* są w posiadaniu *hurtowni* – znajdują się w magazynie. Baza zawiera również *pracowników pojazdy*, które są konieczne do poprawnego działania biznesu. *Pracownicy* występują w trzech kategoriach: *Kasjer*, *Magazynier* i *Kierowca*, natomiast pojazdy w dwóch rodzajach: *Wózek widłowy* oraz *Samochód dostawczy*. Dodatkowo w bazie znajdują się informacje na temat *wynagrodzeń* dla *pracowników*.

## 2. Definicja systemu

Założenia funkcjonalności systemu:

1. Dodawanie, usuwanie i edycja zatrudnionych pracowników
2. Przegląd danych pracowników
3. Dodawanie, usuwanie i edycja posiadanych pojazdów
4. Przegląd danych pojazdów
5. Dodawanie, usuwanie i edycja posiadanych klientów
6. Przegląd danych klientów
7. Dodawanie, usuwanie i edycja otrzymywanych dostaw
8. Przegląd danych dostaw
9. Dodawanie, usuwanie i edycja zamówień składanych przez klientów
10. Przegląd danych zamówień
11. Dodawanie, usuwanie i edycja posiadanych produktów
12. Przegląd danych produktów

### 2.1. Perspektywy użytkowników

#### 2.1.1. Administrator

Posiada dostęp do wszystkich funkcjonalności systemu i modyfikacji bazy danych. Ma uprawnienia administratora bazy danych Oracle.

#### 2.1.2. Manager

Ma dostęp do danych pracowników, pojazdów, dostaw oraz klientów.

#### 2.1.3. Kadry

Ma zagwarantowany dostęp do danych pracowników i ich wynagrodzeń.

#### 2.1.4. Zaopatrzeniowiec

Ma dostęp do danych magazynu, dostaw oraz zamówień.

#### 2.1.5. Koordynator

Posiada dostęp do danych pracowników oraz możliwość modyfikowania danych klientów.

### 2.1.6. Customer Service Representative

Ma dostęp do danych klientów, zamówień, dostaw oraz produktów.

### 2.1.7. Klient

Posiada możliwość przeglądania danych produktów oraz zamówień w zawężonym zakresie.

**Tablica funkcjonalności:**

	Admin	Manager	Kadry	Zaopatrzenie	Koordynator	CSR	Klient
1	+	+	+				
2	+	+	+		+		
3	+	+		+			
4	+	+	+	+	+		
5	+	+			+		
6	+	+			+	+	
7	+	+		+	+		
8	+	+		+	+	+	
9	+				+		
10	+	+	+	+	+		
11	+			+	+	+	
12	+	+			+	+	+

## 3. Model konceptualny

### 3.1. Definicja zbiorów encji określonych w projekcie oraz określenie atrybutów i ich dziedzin

Hurtownia części elektrycznych

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
idHurtowni	NUMBER	TAK	Numer identyfikujący hurtownię
dataPowstania	DATE	TAK	Data powstania hurtowni
nazwa	VARCHAR(20)	TAK	Nazwa hurtowni
adres	VARCHAR(200)	TAK	Adres pod którym znajduje się hurtownia
właściciel	VARCHAR(200)	TAK	Imię i Nazwisko właściciela hurtowni
NIP	VARCHAR(10)	TAK	NIP hurtowni

## Pracownik

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
idPracownika	NUMBER	TAK	Numer identyfikacyjny pracownika
imię	VARCHAR(20)	TAK	Imię pracownika
nazwisko	VARCHAR(20)	TAK	Nazwisko pracownika
Plec	Plec	TAK	Płeć pracownika
dataUrodzenia	Date	TAK	Data urodzenia pracownika
stanowisko	VARCHAR(20)	TAK	Stanowisko, zajmowane przez pracownika
dataZatrudnienia	Date	TAK	Data zatrudnienia pracownika
PESEL	VARCHAR(11)		PESEL pracownika
numerKonta	VARCHAR(20)	TAK	Numer konta, na które ma zostać przesłane wynagrodzenie
adres	VARCHAR(20)	TAK	Adres zamieszkania pracownika
numerTelefonu	VARCHAR(11)	TAK	Numer telefonu kontaktowego pracownika
formaZatrudnienia	Forma_zatrudnienia	TAK	Forma zatrudnienia na jaką pracownik jest zarejestrowany. Możliwe: UMOWA O DZIEŁO, UMOWA O PRACĘ, UMOWA ZLECENIE, UMOWA AGENCYJNA, SAMOZATRUDNIENIE, STAZ

## Magazynier

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Czy_kurs_na_wozki_widlowe	BOOLEAN	TAK	Informacja o możliwości prowadzenia wózka widłowego przez magazyniera
Data_waznosci_kursu	DATE	NIE	Data ważności kursu na prawo jazdy

## Kierowca

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Numer_prawa_jazdy	VARCHAR(14)	TAK	Numer prawa jazdy, którym legitymuje się kierowca
Data_wydania	DATE	TAK	Data wydania prawa jazdy
Data_waznosci	DATE	TAK	Data ważności prawa jazdy

## Wynagrodzenie

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
idWynagrodzenia	NUMBER	TAK	Numre identyfikujący dane wynagrodzenia
dataWypłaty	DATE	TAK	Data wystania przelewu – wynagrodzenia
kwotaBrutto	MONEY	TAK	Kwata BRUTTO wynagrodzenia
kwotaNetto	MONEY	TAK	Kwota NETTO wynagrodzenia

## Klient

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
idKlienta	NUMBER	TAK	Numer identyfikujący danego klienta
NIP	NUMBER	TAK	Numer NIP danego klienta
Nazwa	VARCHAR(30)	TAK	Nazwa firmy klienta
numerTelefonu	VARCHAR(11)	TAK	Numer kontaktowy do klienta
adresMailowy	VARCHAR(40)	NIE	Adres mail klienta
adres	VARCHAR(100)	TAK	Adres wysyłki klienta
upust	DECIMAL(10,0)	TAK	Upust jaki jest przypisany do klienta

## Zamówienie

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
idZamowienia	NUMBER	TAK	Numer identyfikujący dane zamówienie
dataZamówienia	DATE	TAK	Data złożenia zamówienia
terminPłatności	DATE	TAK	Termin płatności za zamówienie
sposóbPłatności	Sposób_platnosci	TAK	Sposob_platnosci spośród 'GOTOWKA', 'PRZELEW', 'BLIK', 'TALON'
numerFaktury	VARCHAR(15)	TAK	Numer identyfikujący fakturę za zamówienie
statusOpłacone	BOOLEAN	TAK	Status określający czy zamówienie zostało już opłacone

## Produkt

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
ID_Projektu	NUMBER	TAK	Numer identyfikujący dany produkt
nazwaTowaru	VARCHAR(20)	TAK	Nazwa pod którą dany produkt jest zamawiany
pakowanie	VARCHAR(15)	TAK	Sposób w jaki pakowany jest dany produkt
cenaNETTO	MONEY	TAK	Cena netto za dany produkt
stawkaVAT	DECIMAL(10,0)	TAK	Stawka VAT za produkt
upust	DECIMAL(10,0)	NIE	Upust jaki obowiązuje na dany produkt
dataWejsciaDoMagazynu	DATE	NIE	Data kiedy produkt został zarejestrowany w magazynie
dataWyjsciaZMagazynu	DATE	NIE	Data kiedy produkt został wyrejestrowany z magazynu

## Dostawa

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
ID_Dostawy	NUMBER	TAK	Numer identyfikujący daną dostawę
dataDostawy	DATE	TAK	Data na jaką jest umówiona dostawa
numerUmowy	INTEGER	TAK	Numer umowy w ramach której realizowana jest dana dostawa
numerAwizoDostawy	INTEGER	TAK	Numer Awiza do Dostawy
numerPotwierdzeniaOdbioru	INTEGER	TAK	Numer Potwierdzenia Odbioru dostawy

## Pojazd

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
ID_Pojazdu	NUMBER	TAK	Numer identyfikujący dany pojazd
dataProdukcji	DATE	TAK	Data produkcji pojazdu
typSilnika	INTEGER	NIE	Rodzaj silnika ('BENZyna', 'DIESEL')
dataPrzeglądu	INTEGER	TAK	Data ważności przeglądu pojazdu



## Wozek widlowy

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Maksymalny_ładunek	NUMBER	TAK	Maksymalny możliwy ładunek (w kg)
Maksymalna_wysokosc_podnosnika	DATE	TAK	Maksymalna możliwa do osiągnięcia wysokość podnośnika ( w metrach)

## Samochod dostawczy

Nazwa atrybutu	Typ i dziedzina	Wymagania dostępności	Opis
Marka	VARCHAR(20)	TAK	Marka samochodu
Numer rejestracyjny	VARCHAR(10)	TAK	Numer rejestracyjny
Numer VIN	VARCHAR(30)	TAK	Numer VIN

### 3.2. Ustalenie związków między encjami i ich typów

- HURTOWNIA – PRACOWNIK

Encja hurtownie zatrudnia wiele pracowników. W sytuacji gdy hurtownia została dopiero utworzona może nie zatrudniać żadnego pracownika. Każdy pracownik przypisany jest do hurtowni.

- HURTOWNIA – KLIENT

Hurtownia posiada wielu klientów, jednak możliwa też jest sytuacja w której nie posiada żadnego klienta.

- HURTOWNIA – DOSTAWA

Hurtownia otrzymuje dostawy, jednak możliwa też jest sytuacja w której nie otrzymuje żadnej dostawy.

- HURTOWNIA – PRODUKT

Hurtownia posiada wiele produktów, jednak możliwa też jest sytuacja w której nie posiada żadnego produktu.

- KLIENT – ZAMÓWIENIE

Klient może posiadać wiele zamówień, jednak jest możliwe że nie będzie posiadał żadnego zamówienia np. w sytuacji kiedy dopiero został wpisany do rejestru

- PRACOWNIK – POJAZD

Pracownik obsługuje samochód i jest to relacja jeden do wielu

- PRACOWNIK – WYNAGRODZENIE

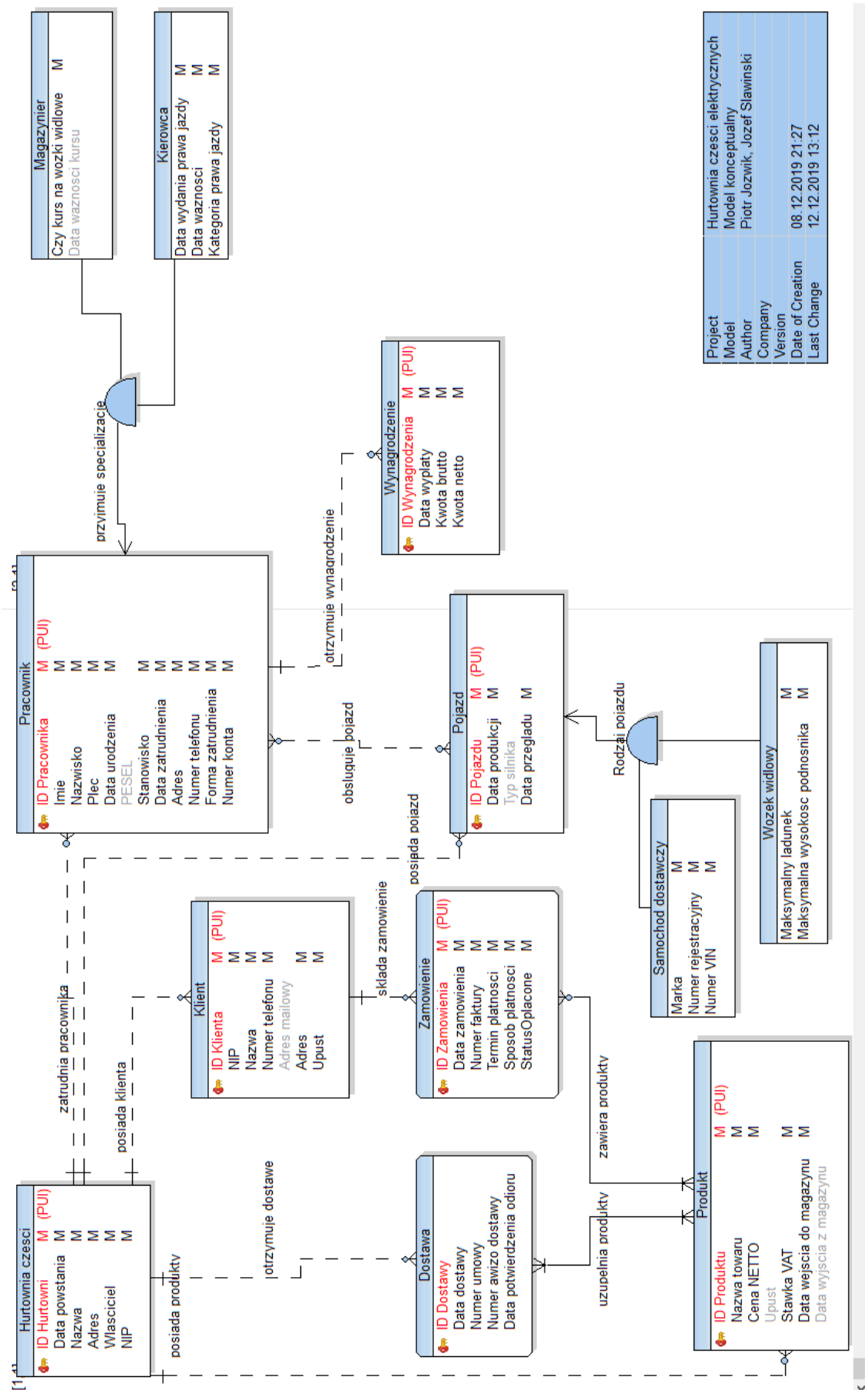
pracownik otrzymuje wynagrodzenie. Możliwa jest sytuacja w której pracownik nie otrzymuje wynagrodzenia, jeżeli dopiero co został zatrudniony i nie minął termin wypłaty wynagrodzenia. Pracownik otrzymuje wynagrodzenie wielokrotnie dlatego to relacja jeden do wielu.

### 3.3. Klucze główne i kandydujące

Na potrzeby naszego projektu stworzyliśmy unikatowe numery ID, które jednoznacznie identyfikują obiekt w bazie danych.

Encja	Klucz główny	Klucz kandydujący
Hurtownia części elektrycznych	ID Hurtowni	Nazwa
Klient	ID Klienta	NIP
Dostawa	ID Dostawy	
Zamówienie	ID Zamówienia	Numer faktury
Produkt	ID Produktu	
Pracownik	ID Pracownika	PESEL
Magazynier		
Kierowca		
Pojazd	ID Pojazdu	
Samochod dostawczy		
Wozek widlowy		
Wynagrodzenie	ID Wynagrodzenia	

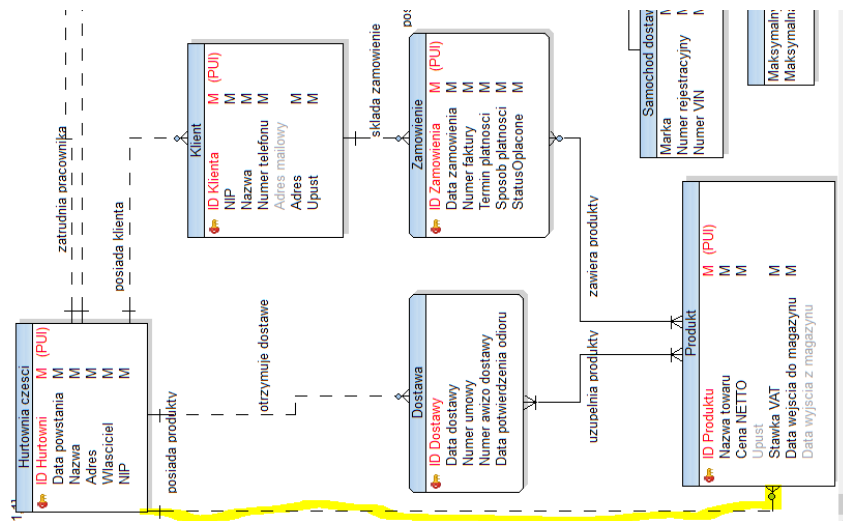
3.4. Schemat ER na poziomie konceptualnym



### 3.5 Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

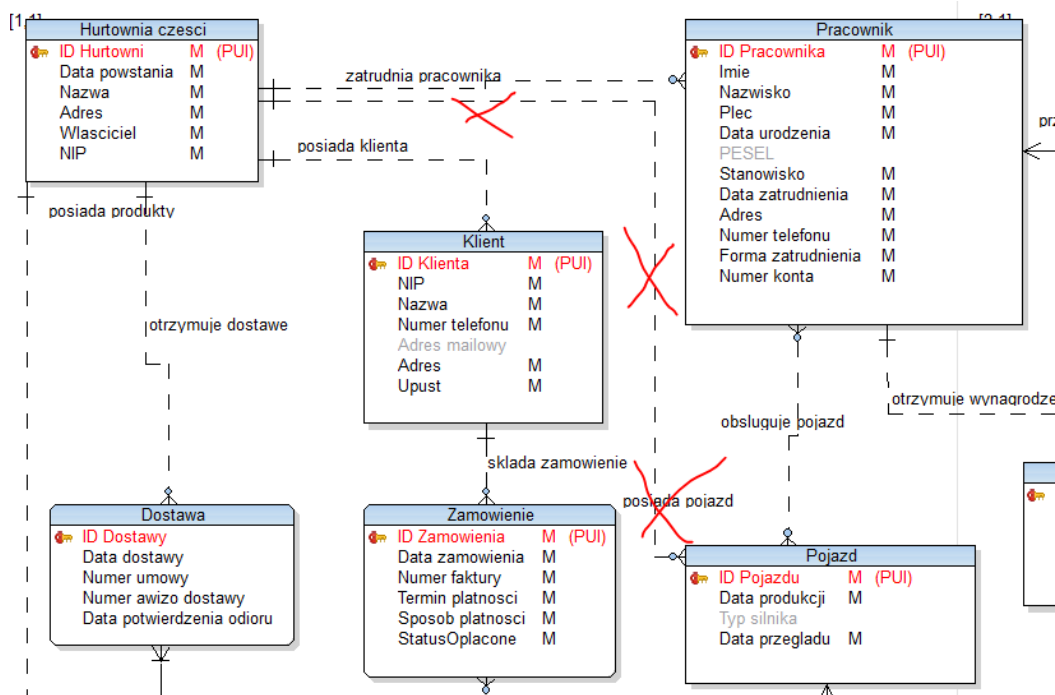
#### Pułapka szczelinowa

Na początku w naszym modelu produkty były związane z zamówieniami i dostawami. To by spowodowało usunięcie wszystkich produktów w przypadku anulacji zamówienia lub dostawy, czyli szczelina. Rozwiązaniem tego problemu było dodanie relacji **Hurtownia części elektrycznych** → **Produkt**. Więc usunięcie produktu mogłoby wystąpić tylko w razie usunięcia całej hurtowni, co jest logiczne.



#### Pułapka wachlarzowa

W przypadku gdyby pojazd mógłby być obsługiwany przez wiele pracowników usunięto by związek łączący Hurtownię z Pojazdem, z punktu widzenia pracownika nie można byłoby stwierdzić z której hurtowni jest obsługiwany pojazd.



## 4. Model logiczny

### 4.1. Charakterystyka modelu relacyjnego

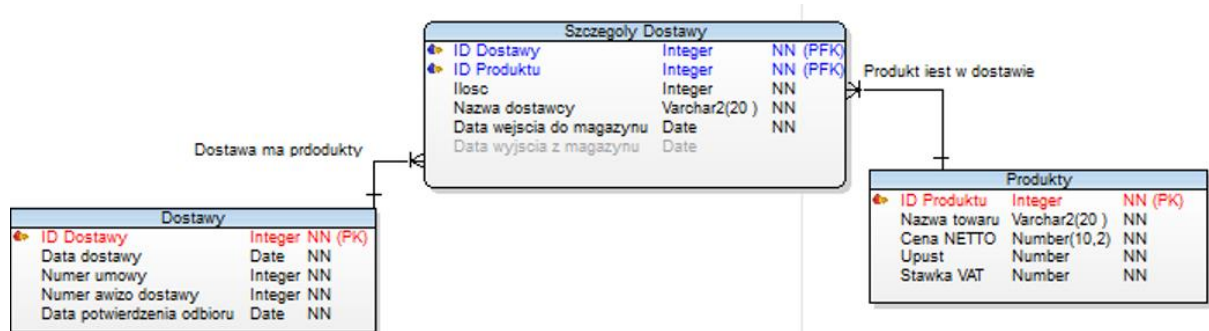
W celu uzyskania modelu logicznego, usunęliśmy niekompatybilności z modelem relacyjnym, między innymi usunęliśmy związki wielu do wielu, używając tablic łączących. Nazwy wszystkich encji zostały zmienione na liczbę mnogą. Pozbycie się atrybutów wielowartościowych oraz segmentowych wiązało się ze stworzeniem nowych encji, lub tzw.: słowników. W naszej bazie danych takimi atrybutami w modelu koncepcyjnym są: właściciel(encja Hurtownia), adres(encja właściciel, klient, pracownik) oraz kategoria prawa jazdy(encja Kierowca).

### 4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym – przykłady

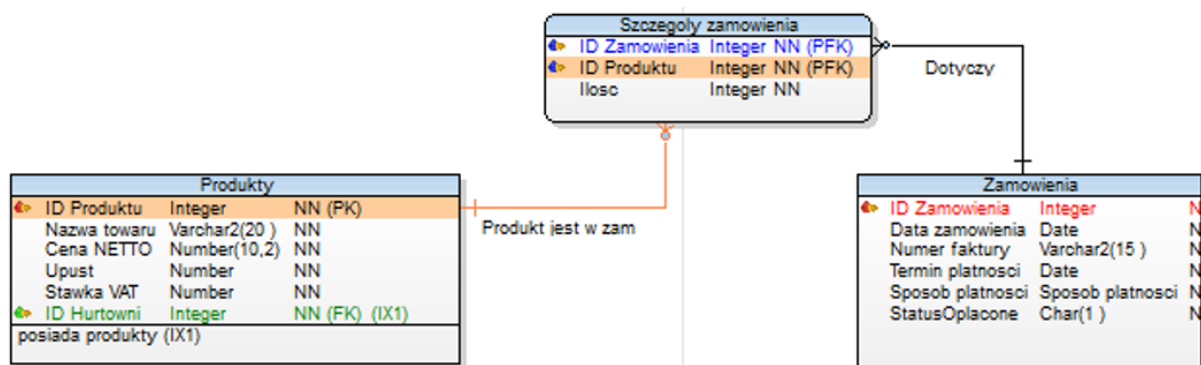
#### 4.2.1 Usunięcie związków wielu do wielu

Główną niekompatybilnością z modelem relacyjnym była obecność związków wielu do wielu na etapie koncepcyjnym. Zostały również zmienione nazwy encji na liczne mnogą w celu odróżnienia ich od relacji.

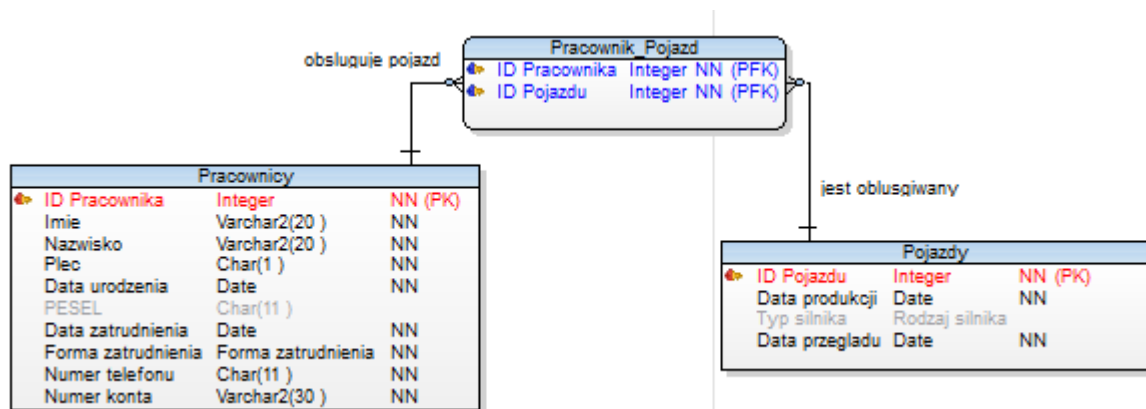
Jednym z tego typu związku był związek Dostawa-Produkt. Dostawa może zawierać wiele produktów, a dany produkt może znajdować się w różnych dostawach. Problem też został rozwiązany dodając tabele łączące 'Szczegóły dostawy'.



Drugim związkiem jest związek Zamówienie-Produkt. Produkt może znajdować się w różnych zamówieniach wiele razy i analogicznie zamówienie może zawierać wiele produktów. Tutaj również została zastosowana tabela brydgująca – 'Szczegóły zamówienia':



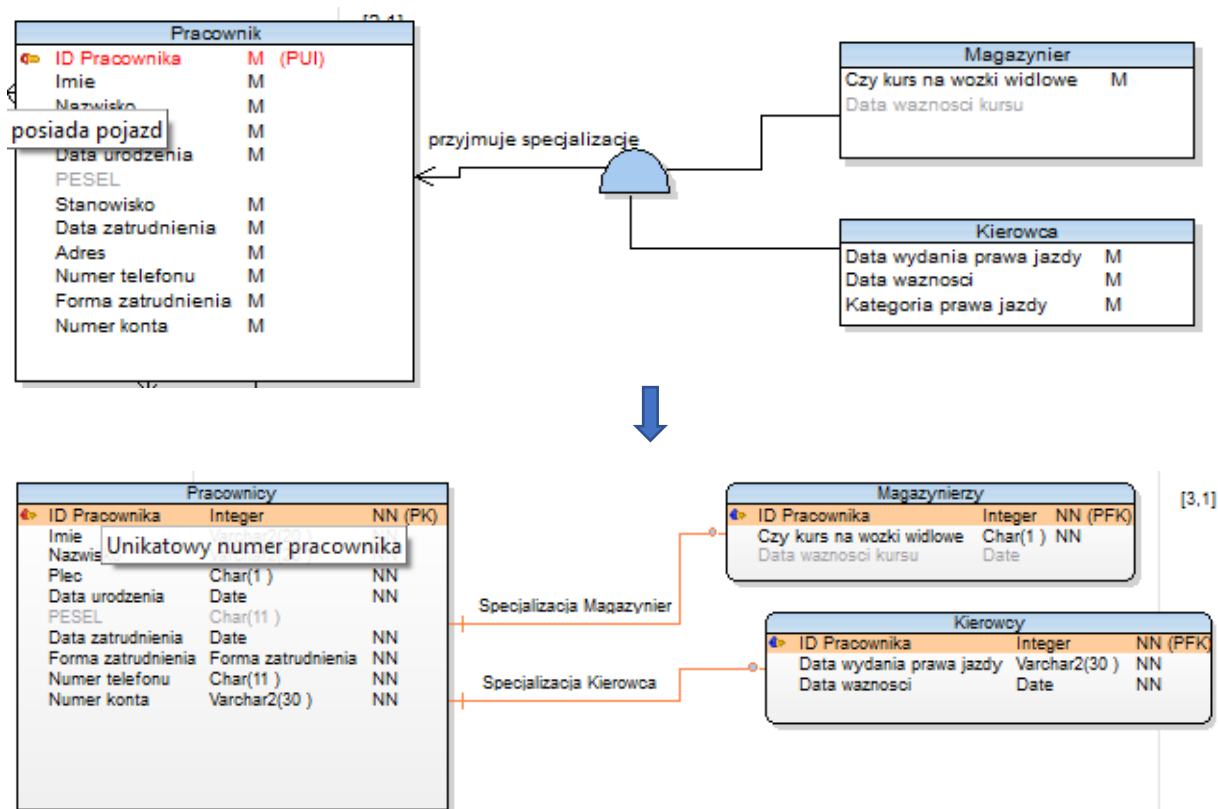
Następnym związkiem wielu do wielu jest związek Pracownik-Pojazd. Pracownik może obsługiwać kilka pojazdów, a pojazd może być obsługiwany przez wielu pracowników.



#### 4.2.2 Usunięcie dziedziczenia

Dziedziczenie na poziomie konceptualnym zostało zastąpione relacjami typu 1:1:

Przykład:



#### 4.3. Proces normalizacji – analiza i przykłady

Proces normalizacji polegał na sprowadzeniu naszej bazy danych do 3 Postaci Normalnej, oraz sprawdzeniu bazy pod względem redundancji danych.

- 1 postać normalna:

- wartości są atomowe (nierozkładalne)
- brak powtarzających się grup

W tym celu atrybuty segmentowe zostały zamienione na atrybuty atomowe:

Adres -> Miasto, Numer budynku, Numer lokalu, Kod pocztowy

Właściciel -> Imię, Nazwisko, Numer telefonu, Adres mailowy

W celu uniknięcia powtarzających się grup oraz atrybutów wielowartościowych stworzyliśmy osobne encje (słowniki).

Grupą powtarzającą się był Adres, zawierający się w encjach: 'Hurtownia części elektrycznych', 'Pracownik', 'Klient' na etapie konceptualnym, więc stworzyliśmy nową encję 'Adresy'

Adresy			
ID Adresu	Integer	NN (PK)	
Miasto	Varchar2(20 )	NN	
Numer budynku	Varchar2(5 )	NN	
Numer lokalu	Varchar2(4 )	NN	
Kod pocztowy	Char(8 )	NN	

Atrybutem wielowartościowym natomiast był atrybut 'Właściciel' oraz 'Kategoria prawa jazdy'. Aby pozbyć się tego problemu stworzyliśmy dwie nowe encje:

Kategorie			
ID kategorii	Integer	NN (PK)	
Kod kategorii	Varchar2(6 )	NN	
Opis kategorii	Varchar2(300 )	NN	

Wlasciciele			
ID Wlasciciela	Integer	NN (PK)	
Imie	Varchar2(20 )	NN	
Nazwisko	Varchar2(20 )	NN	
Numer telefonu	Varchar2(15 )	NN	
Adres mailowy	Varchar2(30 )		
ID Hurtowni	Integer	NN (FK) (IX1)	
ID Adresu	Integer	NN (FK) (IX2)	
IX Relationship1 (IX1)			

Stworzyliśmy również encję 'Stanowiska' gdyż atrybut ten na etapie konceptualnym w encji 'Pracownicy' był elementem powtarzającym się dla każdego pracownika:

Stanowiska			
ID stanowisko	Integer	NN (PK)	
Nazwa	Varchar2(30 )	NN	(AK1)
Opis	Varchar2(400 )		

- 2 postać normalna:

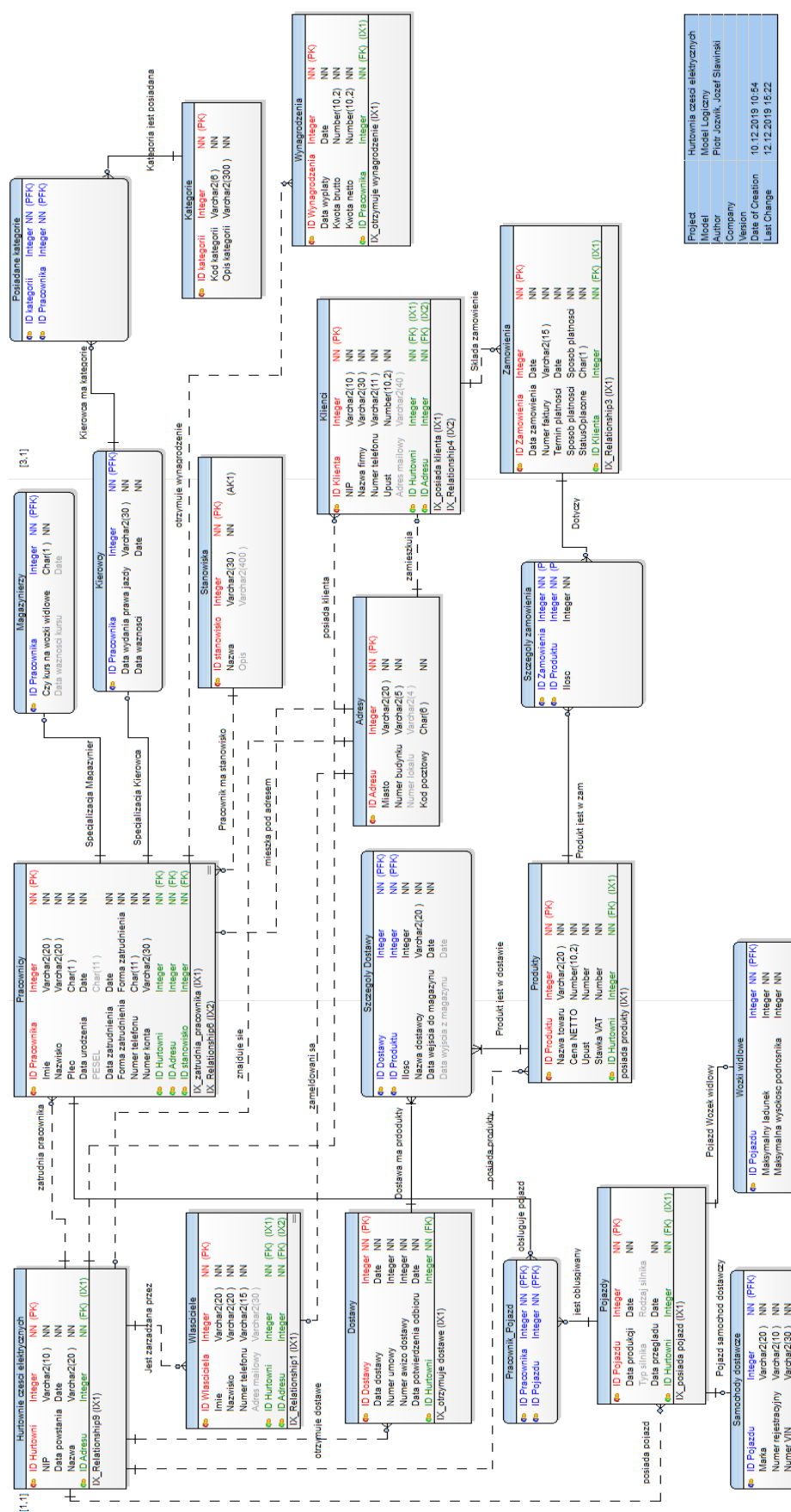
- zawiera się w 1 postaci normalnej
- jest spełniona, gdyż wszystkie klucze w naszej bazie danych są kluczami prostymi

- 3 postać normalna:

- zawiera się w 2 postaci normalnej
- brak zależności przechodnich

Postać ta również została spełniona dla naszej bazy danych.

#### 4.4. Schemat ER na poziomie modelu logicznego





#### 4.5. Więzy integralności wynikającego z modelu relacyjnego

Dzięki węzłom integralności jesteśmy w stanie zabezpieczyć dane przed straceniem spójności, podczas operacji na nich.

W celu zabezpieczenia bazy większość atrybutów jest obowiązkowa (nie może przyjąć wartości NULL). Co więcej każdy klucz główny jest unikatowy aby jednoznacznie identyfikował każdy rekord relacji. Zastosowaliśmy również wiele dziedzin i reguł, aby dokładniej określić zakres danych pól. Dodatkowo w bazie nie występują wartości segmentowe, każdy atrybut ma swoją własną nazwę, a zbiory encji mają atrybuty o wartościach atomowych.

#### 4.6. Proces denormalizacji

Proces denormalizacji polega na wprowadzeniu kontrolowanej nadmierności w celu przyspieszenia operacji w bazie. Nasza baza jest dość mała i jej szybkość jest zadawalająca dlatego zrezygnowaliśmy z tego procesu.

Przykładem tego procesu mogłoby być scalenie relacji Pracownik-Stanowisko, co wprowadziło by pewną redundancję.

Kolejnym przykładem mogłoby być zrezygnowanie ze zbioru encji Adresy, co spowodowałoby powstanie pola segmentowego, składającego się z: miasta, ulicy, kodu pocztowego oraz numeru budynku i lokalu. Aktualizacja każdego z tych atrybutów wiązałaby się z aktualizacją całego segmentu danych co spowodowałoby znaczne obniżenie elastyczności bazy.

Podobna sytuacja wystąpiłaby przy zrezygnowanie ze zbioru encji Właściciele i zastąpieniu jej polem wielosegmentowym.

### 5. Faza fizyczna

#### 5.1. Projekt transakcji i weryfikacja ich wykonalności

Numer	Transakcja	Wykonalna?
1	Dodawanie, usuwanie i edycja zatrudnionych pracowników	TAK
2	Przegląd danych pracowników	TAK
3	Dodawanie, usuwanie i edycja posiadanych pojazdów	TAK
4	Przegląd danych pojazdów	TAK
5	Dodawanie, usuwanie i edycja posiadanych klientów	TAK
6	Przegląd danych klientów	TAK
7	Dodawanie, usuwanie i edycja otrzymywanych dostaw	TAK
8	Przegląd danych dostaw	TAK
9	Dodawanie, usuwanie i edycja zamówień składanych przez klientów	TAK
10	Przegląd danych zamówień	TAK
11	Dodawanie, usuwanie i edycja posiadanych produktów	TAK
12	Przegląd danych produktów	TAK

## 5.2. Strojenie bazy danych - dobór indeksów

Index pozwala uzyskać szybszy dostęp do danych. Dobrane przez nas indexy:

- Pod jakim adresem znajduje się hurtownia  
`CREATE INDEX ZNAJDUJE_SIE ON HURTOWNIE_CZESCI_ELEKTRYCZNYCH (ID_ADRESU)`
- jakich pracowników zatrudnia hurtownia  
`CREATE INDEX ZATRUDNIA_PRACOWNIKA ON PRACOWNICY (ID_HURTOWNI)`
- Pod jakim adresem mieszka pracownik:  
`CREATE INDEX MIESZKA_POPD_ADRESEM ON PRACOWNICY (ID_ADRESU)`
- Jakie stanowisko ma pracownik:  
`CREATE INDEX PRACOWNIK_MA_STANOWISKO ON PRACOWNICY (ID_STANOWISKO)`
- Jakie wynagrodzenie otrzymuje pracownik:  
`CREATE INDEX OTRZYMUJE_WYNAGRODZENIE ON WYNAGRODZENIA (ID_PRACOWNIKA)`
- Zamowienie składane przez klienta:  
`CREATE INDEX SKLADA_ZAMOWIENIE ON ZAMOWIENIA (ID_KLIENTA)`
- Hurtownia otrzymuje dostawę:  
`CREATE INDEX OTRZYMUJE_DOSTAWĘ ON DOSTAWY (ID_HURTOWNI)`
- Posiadanie klienta przez hurtownię:  
`CREATE INDEX POSIADA_KLIENTA ON KLIENCI (ID_HURTOWNI)`
- Pod jakim adresem znajduje się firma klienta  
`CREATE INDEX ZAMIESZKUJA ON KLIENCI (ID_ADRESU)`
- Jakie pojazdy posiada hurtownia:  
`CREATE INDEX POSIADA_POJAZD ON POJAZDY (ID_HURTOWNI)`
- Przez kogo zarządzana jest hurtownia:  
`CREATE INDEX JEST_ZARZADZANA_PRZEZ ON WLASCICIELE (ID_HURTOWNI)`
- Pod jakim adresem zameldowani są właściciele:  
`CREATE INDEX ZAMELDOWANI_SA ON WLASCICIELE (ID_ADRESU)`
- Jakie produkty posiada hurtownia:  
`CREATE INDEX POSIADA_PRODUKTY ON PRODUKTY`

## 5.3. Skrypt SQL zakładający bazę danych

```
/*  
CREATED: 10.12.2019  
MODIFIED: 12.12.2019  
PROJECT: HURTOWNIA CZESCI ELEKTRYCZNYCH  
MODEL: MODEL LOGICZNY  
AUTHOR: PIOTR JOZWIK, JOZEF SLAWINSKI  
DATABASE: ORACLE 19C  
*/  
-- CREATE SEQUENCES SECTION -----
```

```
CREATE SEQUENCE HURTSEQ
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/
```

```
CREATE SEQUENCE WLASCICIELSEQ
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/
```

```
CREATE SEQUENCE DOSTAWASEQ
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/
```

```
CREATE SEQUENCE ADRESYSEQ
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/
```

```
CREATE SEQUENCE KLIENCISEQ
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/
```

```
CREATE SEQUENCE WYNAGRODZENIASIQ
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
```

/

```
CREATE SEQUENCE POJAZDYSEQ  
  INCREMENT BY 1  
  START WITH 1  
  NOMAXVALUE  
  NOMINVALUE  
  CACHE 20
```

/

```
CREATE SEQUENCE PRODUKTYSEQ  
  INCREMENT BY 1  
  START WITH 1  
  NOMAXVALUE  
  NOMINVALUE  
  CACHE 20
```

/

```
CREATE SEQUENCE ZAMOWIENIASIQ  
  INCREMENT BY 1  
  START WITH 1  
  NOMAXVALUE  
  NOMINVALUE  
  CACHE 20
```

/

```
CREATE SEQUENCE KATEGORIASEQ  
  INCREMENT BY 1  
  START WITH 1  
  NOMAXVALUE  
  NOMINVALUE  
  CACHE 20
```

/

```
CREATE SEQUENCE PRACOWNICYSEQ  
  INCREMENT BY 1  
  START WITH 1  
  NOMAXVALUE  
  NOMINVALUE  
  CACHE 20
```

/

```
CREATE SEQUENCE STANOWISKASEQ  
  INCREMENT BY 1  
  START WITH 1  
  NOMAXVALUE  
  NOMINVALUE  
  CACHE 20
```

/

```
-- CREATE TABLES SECTION -----
```

```
-- TABLE HURTOWNIE_CZESCI_ELEKTRYCZNYCH
```

```
CREATE TABLE HURTOWNIE_CZESCI_ELEKTRYCZNYCH(  
    ID_HURTOWNI INTEGER NOT NULL,  
    NIP VARCHAR2(10 ) NOT NULL,  
    DATA_POWSTANIA DATE NOT NULL,  
    NAZWA VARCHAR2(20 ) NOT NULL,  
    ID_ADRESU INTEGER NOT NULL  
)  
/
```

```
-- CREATE INDEXES FOR TABLE HURTOWNIE_CZESCI_ELEKTRYCZNYCH
```

```
CREATE INDEX ZNAJDUJE_SIE ON HURTOWNIE_CZESCI_ELEKTRYCZNYCH (ID_ADRESU)  
/
```

```
-- ADD KEYS FOR TABLE HURTOWNIE_CZESCI_ELEKTRYCZNYCH
```

```
ALTER TABLE HURTOWNIE_CZESCI_ELEKTRYCZNYCH ADD CONSTRAINT UNIQUE_IDENTIFIER1 P  
RIMARY KEY (ID_HURTOWNI)  
/
```

```
-- TABLE PRACOWNICY
```

```
CREATE TABLE PRACOWNICY(  
    ID_PRACOWNIKA INTEGER NOT NULL,  
    IMIE VARCHAR2(20 ) NOT NULL,  
    NAZWISKO VARCHAR2(20 ) NOT NULL,  
    PLEC CHAR(1 ) NOT NULL  
        CONSTRAINT JAKA_PLEC CHECK (PLEC IN ('K','M')),  
    DATA_URODZENIA DATE NOT NULL,  
    PESEL CHAR(11 ),  
    DATA_ZATRUDNIENIA DATE NOT NULL,  
    FORMA_ZATRUDNIENIA VARCHAR2(50 ) NOT NULL  
        CHECK (FORMA_ZATRUDNIENIA IN ('UMOWA O DZIELO','UMOWA O PRACE','UMOWA  
ZLECENIE','UMOWA AGENCYJNA','SAMOZATRUDNIENIE','STAZ')),  
    NUMER_TELEFONU CHAR(11 ) NOT NULL,  
    NUMER_KONTA VARCHAR2(30 ) NOT NULL,  
    ID_HURTOWNI INTEGER NOT NULL,  
    ID_ADRESU INTEGER NOT NULL,  
    ID_STANOWISKO INTEGER NOT NULL  
)  
/
```

```
-- CREATE INDEXES FOR TABLE PRACOWNICY
```

```
CREATE INDEX ZATRUDNIA_PRACOWNIKA ON PRACOWNICY (ID_HURTOWNI)
/
```

```
CREATE INDEX MIESZKA_POPD_ADRESEM ON PRACOWNICY (ID_ADRESU)
/
```

```
CREATE INDEX PRACOWNIK_MA_STANOWISKO ON PRACOWNICY (ID_STANOWISKO)
/
```

```
-- ADD KEYS FOR TABLE PRACOWNICY
```

```
ALTER TABLE PRACOWNICY ADD CONSTRAINT UNIQUE_IDENTIFIER2 PRIMARY KEY (ID_PRACOWNIKA)
/
```

```
-- TABLE WYNAGRODZENIA
```

```
CREATE TABLE WYNAGRODZENIA(
    ID_WYNAGRODZENIA INTEGER NOT NULL,
    DATA_WYPLATY DATE NOT NULL,
    KWOTA_BRUTTO NUMBER(10,2) NOT NULL,
    KWOTA_NETTO NUMBER(10,2) NOT NULL,
    ID_PRACOWNIKA INTEGER NOT NULL
)
/
```

```
-- CREATE INDEXES FOR TABLE WYNAGRODZENIA
```

```
CREATE INDEX OTRZYMUJE_WYNAGRODZENIE ON WYNAGRODZENIA (ID_PRACOWNIKA)
/
```

```
-- ADD KEYS FOR TABLE WYNAGRODZENIA
```

```
ALTER TABLE WYNAGRODZENIA ADD CONSTRAINT UNIQUE_IDENTIFIER3 PRIMARY KEY (ID_WYNAGRODZENIA)
/
```

```
-- TABLE ZAMOWIENIA
```

```
CREATE TABLE ZAMOWIENIA(
    ID_ZAMOWIENIA INTEGER NOT NULL,
    DATA_ZAMOWIENIA DATE NOT NULL,
    NUMER_FAKTURY VARCHAR2(15 ) NOT NULL,
    TERMIN_PLATNOSCI DATE NOT NULL,
    SPOSOB_PLATNOSCI VARCHAR2(30 ) NOT NULL
        CHECK (SPOSOB_PLATNOSCI IN ('GOTOWKA', 'PRZELEW', 'BLIK', 'TALON')),
    STATUSOPLACONE CHAR(1 ) NOT NULL,
    ID_KLIENTA INTEGER NOT NULL
)
/
```

```
/
```

```
-- CREATE INDEXES FOR TABLE ZAMOWIENIA
```

```
CREATE INDEX SKLADA_ZAMOWIENIE ON ZAMOWIENIA (ID_KLIENTA)
```

```
/
```

```
-- ADD KEYS FOR TABLE ZAMOWIENIA
```

```
ALTER TABLE ZAMOWIENIA ADD CONSTRAINT UNIQUE_IDENTIFIER4 PRIMARY KEY (ID_ZAMOWIENIA)
```

```
/
```

```
-- TABLE KIEROWCY
```

```
CREATE TABLE KIEROWCY(  
    ID_PRACOWNIKA INTEGER NOT NULL,  
    DATA_WYDANIA_PRAWA_JAZDY VARCHAR2(30 ) NOT NULL,  
    DATA_WAZNOSCI DATE NOT NULL
```

```
)
```

```
/
```

```
-- ADD KEYS FOR TABLE KIEROWCY
```

```
ALTER TABLE KIEROWCY ADD CONSTRAINT UNIQUE_IDENTIFIER5 PRIMARY KEY (ID_PRACOWNIKA)
```

```
/
```

```
-- TABLE MAGAZYNIERZY
```

```
CREATE TABLE MAGAZYNIERZY(  
    ID_PRACOWNIKA INTEGER NOT NULL,  
    CZY_KURS_NA_WOZKI_WIDLOWE CHAR(1 ) NOT NULL,  
    DATA_WAZNOSCI_KURSU DATE
```

```
)
```

```
/
```

```
-- ADD KEYS FOR TABLE MAGAZYNIERZY
```

```
ALTER TABLE MAGAZYNIERZY ADD CONSTRAINT UNIQUE_IDENTIFIER6 PRIMARY KEY (ID_PRACOWNIKA)
```

```
/
```

```
-- TABLE DOSTAWY
```

```
CREATE TABLE DOSTAWY(  
    ID_DOSTAWY INTEGER NOT NULL,  
    DATA_DOSTAWY DATE NOT NULL,  
    NUMER_UMOWY INTEGER NOT NULL,
```

```

    NUMER_AWIZO_DOSTAWY INTEGER NOT NULL,
    DATA_POTWIERDZENIA_ODBIORU DATE NOT NULL,
    ID_HURTOWNI INTEGER NOT NULL
)
/

-- CREATE INDEXES FOR TABLE DOSTAWY

CREATE INDEX OTRZYMUJE_DOSTAWY ON DOSTAWY (ID_HURTOWNI)
/

-- ADD KEYS FOR TABLE DOSTAWY

ALTER TABLE DOSTAWY ADD CONSTRAINT UNIQUE_IDENTIFIER8 PRIMARY KEY (ID_DOSTAWY)
/

-- TABLE KLIENCI

CREATE TABLE KLIENCI(
    ID_KLIENTA INTEGER NOT NULL,
    NIP VARCHAR2(10 ) NOT NULL,
    NAZWA_FIRMY VARCHAR2(30 ) NOT NULL,
    NUMER_TELEFONU VARCHAR2(11 ) NOT NULL,
    UPUST NUMBER(10,2) NOT NULL,
    ADRES_MAILOWY VARCHAR2(40 ),
    ID_HURTOWNI INTEGER NOT NULL,
    ID_ADRESU INTEGER NOT NULL
)
/

-- CREATE INDEXES FOR TABLE KLIENCI

CREATE INDEX POSIADA_KLIENTA ON KLIENCI (ID_HURTOWNI)
/

CREATE INDEX ZAMIESZKUJA ON KLIENCI (ID_ADRESU)
/

-- ADD KEYS FOR TABLE KLIENCI

ALTER TABLE KLIENCI ADD CONSTRAINT UNIQUE_IDENTIFIER9 PRIMARY KEY (ID_KLIENTA)
/

-- TABLE POJAZDY

CREATE TABLE POJAZDY(
    ID_POJAZDU INTEGER NOT NULL,
    DATA_PRODUKCJI DATE NOT NULL,
    TYP_SILNIKA VARCHAR2(10 )
        CHECK (TYP_SILNIKA IN ('DIESEL', 'BENZYNNA')),

```



```

    DATA_PRZEGLADU DATE NOT NULL,
    ID_HURTOWNI INTEGER NOT NULL
)
/

-- CREATE INDEXES FOR TABLE POJAZDY

CREATE INDEX POSIADA_POJAZD ON POJAZDY (ID_HURTOWNI)
/

-- ADD KEYS FOR TABLE POJAZDY

ALTER TABLE POJAZDY ADD CONSTRAINT UNIQUE_IDENTIFIER10 PRIMARY KEY (ID_POJAZDU)
/

-- TABLE WOZKI_WIDLOWE

CREATE TABLE WOZKI_WIDLOWE(
    ID_POJAZDU INTEGER NOT NULL,
    MAKSYMALNY_LADUNEK INTEGER NOT NULL,
    MAKSYMALNA_WYSOKOSC_PODNOSNIKA INTEGER NOT NULL
)
/

-- ADD KEYS FOR TABLE WOZKI_WIDLOWE

ALTER TABLE WOZKI_WIDLOWE ADD CONSTRAINT UNIQUE_IDENTIFIER11 PRIMARY KEY (ID_POJAZDU)
/

-- TABLE SAMOCHODY_DOSTAWCZE

CREATE TABLE SAMOCHODY_DOSTAWCZE(
    ID_POJAZDU INTEGER NOT NULL,
    MARKA VARCHAR2(20) NOT NULL,
    NUMER_REJESTRACYJNY VARCHAR2(10) NOT NULL,
    NUMER_VIN VARCHAR2(30) NOT NULL
)
/

-- ADD KEYS FOR TABLE SAMOCHODY_DOSTAWCZE

```

```
ALTER TABLE SAMOCHODY_DOSTAWCZE ADD CONSTRAINT UNIQUE_IDENTIFIER12 PRIMARY KEY
(ID_POJAZDU)
/
```

```
-- TABLE PRACOWNIK_POJAZD
```

```
CREATE TABLE PRACOWNIK_POJAZD(
    ID_PRACOWNIKA INTEGER NOT NULL,
    ID_POJAZDU INTEGER NOT NULL
)
/
```

```
-- TABLE WLASCICIELE
```

```
CREATE TABLE WLASCICIELE(
    ID_WLASCICIELA INTEGER NOT NULL,
    IMIE VARCHAR2(20 ) NOT NULL,
    NAZWISKO VARCHAR2(20 ) NOT NULL,
    NUMER_TELEFONU VARCHAR2(15 ) NOT NULL,
    ADRES_MAILOWY VARCHAR2(30 ),
    ID_HURTOWNI INTEGER NOT NULL,
    ID_ADRESU INTEGER NOT NULL
)
/
```

```
-- CREATE INDEXES FOR TABLE WLASCICIELE
```

```
CREATE INDEX JEST_ZARZADZANA_PRZEZ ON WLASCICIELE (ID_HURTOWNI)
/
```

```
CREATE INDEX ZAMELDOWANI_SA ON WLASCICIELE (ID_ADRESU)
/
```

```
-- ADD KEYS FOR TABLE WLASCICIELE
```

```
ALTER TABLE WLASCICIELE ADD CONSTRAINT PK_WLASCICIELE PRIMARY KEY (ID_WLASCICIELA)
/
```

```
-- TABLE ADRESY
```

```
CREATE TABLE ADRESY(
    ID_ADRESU INTEGER NOT NULL,
    MIASTO VARCHAR2(20 ) NOT NULL,
    NUMER_BUDYNKU VARCHAR2(5 ) NOT NULL,
    NUMER_LOKALU VARCHAR2(4 ),
    KOD_POCZTOWY CHAR(6 ) NOT NULL,
    ULICA VARCHAR2(20 ) NOT NULL
)
/
```

```
-- ADD KEYS FOR TABLE ADRESY
```

```
ALTER TABLE ADRESY ADD CONSTRAINT PK_ADRESY PRIMARY KEY (ID_ADRESU)  
/
```

```
-- TABLE PRODUKTY
```

```
CREATE TABLE PRODUKTY(  
    ID_PRODUKTU INTEGER NOT NULL,  
    NAZWA_TOWARU VARCHAR2(20 ) NOT NULL,  
    CENA_NETTO NUMBER(10,2) NOT NULL,  
    UPUST NUMBER NOT NULL,  
    STAWKA_VAT NUMBER NOT NULL,  
    ID_HURTOWNI INTEGER NOT NULL  
)  
/
```

```
-- CREATE INDEXES FOR TABLE PRODUKTY
```

```
CREATE INDEX POSIADA_PRODUKTY ON PRODUKTY (ID_HURTOWNI)  
/
```

```
-- ADD KEYS FOR TABLE PRODUKTY
```

```
ALTER TABLE PRODUKTY ADD CONSTRAINT PK_PRODUKTY PRIMARY KEY (ID_PRODUKTU)  
/
```

```
-- TABLE SZCZEGOLY_DOSTAWY
```

```
CREATE TABLE SZCZEGOLY_DOSTAWY(  
    ID_DOSTAWY INTEGER NOT NULL,  
    ID_PRODUKTU INTEGER NOT NULL,  
    ILOSC INTEGER NOT NULL,  
    NAZWA_DOSTAWCY VARCHAR2(20 ) NOT NULL,  
    DATA_WEJSCIA_DO_MAGAZYNU DATE NOT NULL,  
    DATA_WYJSCIA_Z_MAGAZYNU DATE  
)  
/
```

```
-- ADD KEYS FOR TABLE SZCZEGOLY_DOSTAWY
```

```
ALTER TABLE SZCZEGOLY_DOSTAWY ADD CONSTRAINT PK_SZCZEGOLY_DOSTAWY PRIMARY KEY  
(ID_DOSTAWY, ID_PRODUKTU)  
/
```

```
-- TABLE SZCZEGOLY_ZAMOWIENIA
```

```
CREATE TABLE SZCZEGOLY_ZAMOWIENIA(  
    ID_ZAMOWIENIA INTEGER NOT NULL,  
    ID_PRODUKTU INTEGER NOT NULL,  
    ILOSC INTEGER NOT NULL
```

```

)
/

-- ADD KEYS FOR TABLE SZCZEGOLY_ZAMOWIENIA

ALTER TABLE SZCZEGOLY_ZAMOWIENIA ADD CONSTRAINT PK_SZCZEGOLY_ZAMOWIENIA PRIMARY KEY (ID_ZAMOWIENIA, ID_PRODUKTU)
/

-- TABLE KATEGORIE

CREATE TABLE KATEGORIE(
    ID_KATEGORII INTEGER NOT NULL,
    KOD_KATEGORII VARCHAR2(6 ) NOT NULL,
    OPIS_KATEGORII VARCHAR2(300 ) NOT NULL
)
/

-- ADD KEYS FOR TABLE KATEGORIE

ALTER TABLE KATEGORIE ADD CONSTRAINT PK_KATEGORIE PRIMARY KEY (ID_KATEGORII)
/

-- TABLE POSIADANE_KATEGORIE

CREATE TABLE POSIADANE_KATEGORIE(
    ID_KATEGORII INTEGER NOT NULL,
    ID_PRACOWNIKA INTEGER NOT NULL
)
/

-- ADD KEYS FOR TABLE POSIADANE_KATEGORIE

ALTER TABLE POSIADANE_KATEGORIE ADD CONSTRAINT PK_POSIADANE_KATEGORIE PRIMARY KEY (NUMER_KATEGORII, ID_PRACOWNIKA)
/

-- TABLE STANOWISKA

CREATE TABLE STANOWISKA(
    ID_STANOWISKO INTEGER NOT NULL,
    NAZWA VARCHAR2(30 ) NOT NULL,
    OPIS VARCHAR2(400 )
)
/

-- ADD KEYS FOR TABLE STANOWISKA

```

```
ALTER TABLE STANOWISKA ADD CONSTRAINT PK_STANOWISKA PRIMARY KEY (ID_STANOWISKO
)
/
```

```
ALTER TABLE STANOWISKA ADD CONSTRAINT NAZWA UNIQUE (NAZWA)
/
```

```
-
- TRIGGER FOR SEQUENCE HURTSEQ FOR COLUMN ID_HURTOWNI IN TABLE HURTOWNIE_CZESC
I_ELEKTRYCZNYCH -----
```

```
CREATE OR REPLACE TRIGGER TS_HURTOWNIE_HURTSEQ BEFORE INSERT
ON HURTOWNIE_CZESCI_ELEKTRYCZNYCH FOR EACH ROW
BEGIN
```

```
    :NEW.ID_HURTOWNI := HURTSEQ.NEXTVAL;
END;
/
```

```
CREATE OR REPLACE TRIGGER TSU_HURTOWNIE_HURTSEQ AFTER UPDATE OF ID_HURTOWNI
ON HURTOWNIE_CZESCI_ELEKTRYCZNYCH FOR EACH ROW
BEGIN
```

```
    RAISE_APPLICATION_ERROR(-
20010, 'CANNOT UPDATE COLUMN ID_HURTOWNI IN TABLE HURTOWNIE_CZESCI_ELEKTRYCZNYC
H AS IT USES SEQUENCE.');
```

```
END;
/
```

```
-
- TRIGGER FOR SEQUENCE PRACOWNICYSEQ FOR COLUMN ID_PRACOWNIKA IN TABLE PRACOWN
ICY -----
```

```
CREATE OR REPLACE TRIGGER TS_PRACOWNICY_PRACOWNICYSEQ BEFORE INSERT
ON PRACOWNICY FOR EACH ROW
BEGIN
```

```
    :NEW.ID_PRACOWNIKA := PRACOWNICYSEQ.NEXTVAL;
END;
/
```

```
CREATE OR REPLACE TRIGGER TSU_PRACOWNICY_PRACOWNICYSEQ AFTER UPDATE OF ID_PRAC
OWNIKA
ON PRACOWNICY FOR EACH ROW
BEGIN
```

```
    RAISE_APPLICATION_ERROR(-
20010, 'CANNOT UPDATE COLUMN ID_PRACOWNIKA IN TABLE PRACOWNICY AS IT USES SEQUE
NCE.');
```

```
END;
/
```

```
- TRIGGER FOR SEQUENCE WYNAGRODZENIASEQ FOR COLUMN ID_WYNAGRODZENIA IN TABLE W
YNAGRODZENIA -----
```

```
CREATE OR REPLACE TRIGGER TS_W_WYNAGRODZENIASEQ BEFORE INSERT
```

```

ON WYNAGRODZENIA FOR EACH ROW
BEGIN
    :NEW.ID_WYNAGRODZENIA := WYNAGRODZENIASEQ.NEXTVAL;
END;
/
CREATE OR REPLACE TRIGGER TSU_W_WYNAGRODZENIASEQ AFTER UPDATE OF ID_WYNAGRODZENIA
ON WYNAGRODZENIA FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-
20010, 'CANNOT UPDATE COLUMN ID_WYNAGRODZENIA IN TABLE WYNAGRODZENIA AS IT USES
SEQUENCE. ');
END;
/

-
- TRIGGER FOR SEQUENCE ZAMOWIENIASEQ FOR COLUMN ID_ZAMOWIENIA IN TABLE ZAMOWIENIA -----
CREATE OR REPLACE TRIGGER TS_ZAMOWIENIA_ZAMOWIENIASEQ BEFORE INSERT
ON ZAMOWIENIA FOR EACH ROW
BEGIN
    :NEW.ID_ZAMOWIENIA := ZAMOWIENIASEQ.NEXTVAL;
END;
/
CREATE OR REPLACE TRIGGER TSU_ZAMOWIENIA_ZAMOWIENIASEQ AFTER UPDATE OF ID_ZAMOWIENIA
ON ZAMOWIENIA FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-
20010, 'CANNOT UPDATE COLUMN ID_ZAMOWIENIA IN TABLE ZAMOWIENIA AS IT USES SEQUENCE. ');
END;
/

-- TRIGGER FOR SEQUENCE DOSTAWASEQ FOR COLUMN ID_DOSTAWY IN TABLE DOSTAWY -----
CREATE OR REPLACE TRIGGER TS_DOSTAWY_DOSTAWASEQ BEFORE INSERT
ON DOSTAWY FOR EACH ROW
BEGIN
    :NEW.ID_DOSTAWY := DOSTAWASEQ.NEXTVAL;
END;
/
CREATE OR REPLACE TRIGGER TSU_DOSTAWY_DOSTAWASEQ AFTER UPDATE OF ID_DOSTAWY
ON DOSTAWY FOR EACH ROW
BEGIN

```

```

        RAISE_APPLICATION_ERROR(-
20010,'CANNOT UPDATE COLUMN ID_DOSTAWY IN TABLE DOSTAWY AS IT USES SEQUENCE.')
    ;
END;
/

-- TRIGGER FOR SEQUENCE KLIENCISEQ FOR COLUMN ID_KLIENTA IN TABLE KLIENCI ----
-----
CREATE OR REPLACE TRIGGER TS_KLIENCI_KLIENCISEQ BEFORE INSERT
ON KLIENCI FOR EACH ROW
BEGIN
    :NEW.ID_KLIENTA := KLIENCISEQ.NEXTVAL;
END;
/
CREATE OR REPLACE TRIGGER TSU_KLIENCI_KLIENCISEQ AFTER UPDATE OF ID_KLIENTA
ON KLIENCI FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-
20010,'CANNOT UPDATE COLUMN ID_KLIENTA IN TABLE KLIENCI AS IT USES SEQUENCE.')
    ;
END;
/

-- TRIGGER FOR SEQUENCE POJAZDYSEQ FOR COLUMN ID_POJAZDU IN TABLE POJAZDY ----
-----
CREATE OR REPLACE TRIGGER TS_POJAZDY_POJAZDYSEQ BEFORE INSERT
ON POJAZDY FOR EACH ROW
BEGIN
    :NEW.ID_POJAZDU := POJAZDYSEQ.NEXTVAL;
END;
/
CREATE OR REPLACE TRIGGER TSU_POJAZDY_POJAZDYSEQ AFTER UPDATE OF ID_POJAZDU
ON POJAZDY FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-
20010,'CANNOT UPDATE COLUMN ID_POJAZDU IN TABLE POJAZDY AS IT USES SEQUENCE.')
    ;
END;/

-
- TRIGGER FOR SEQUENCE WLASCICIELSEQ FOR COLUMN ID_WLASCICIELA IN TABLE WLASCI
CIELE -----
CREATE OR REPLACE TRIGGER TS_WLASCICIELE_WLASCICIELSEQ BEFORE INSERT
ON WLASCICIELE FOR EACH ROW
BEGIN
    :NEW.ID_WLASCICIELA := WLASCICIELSEQ.NEXTVAL;
END;
/

```

```

CREATE OR REPLACE TRIGGER TSU_WLASCICIELE_WLASCICIELSEQ AFTER UPDATE OF ID_WLA
SCICIELA
ON WLASCICIELE FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-
20010,'CANNOT UPDATE COLUMN ID_WLASCICIELA IN TABLE WLASCICIELE AS IT USES SEQ
UENCE.');
```

END;

/

```

-- TRIGGER FOR SEQUENCE ADRESYSEQ FOR COLUMN ID_ADRESU IN TABLE ADRESY -----
--
CREATE OR REPLACE TRIGGER TS_ADRESY_ADRESYSEQ BEFORE INSERT
ON ADRESY FOR EACH ROW
BEGIN
    :NEW.ID_ADRESU := ADRESYSEQ.NEXTVAL;
END;
```

/

```

CREATE OR REPLACE TRIGGER TSU_ADRESY_ADRESYSEQ AFTER UPDATE OF ID_ADRESU
ON ADRESY FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-
20010,'CANNOT UPDATE COLUMN ID_ADRESU IN TABLE ADRESY AS IT USES SEQUENCE.');
```

END;

/

```

-- TRIGGER FOR SEQUENCE PRODUKTYSEQ FOR COLUMN ID_PRODUKTU IN TABLE PRODUKTY -
-----
CREATE OR REPLACE TRIGGER TS_PRODUKTY_PRODUKTYSEQ BEFORE INSERT
ON PRODUKTY FOR EACH ROW
BEGIN
    :NEW.ID_PRODUKTU := PRODUKTYSEQ.NEXTVAL;
END;
```

/

```

CREATE OR REPLACE TRIGGER TSU_PRODUKTY_PRODUKTYSEQ AFTER UPDATE OF ID_PRODUKTU
ON PRODUKTY FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-
20010,'CANNOT UPDATE COLUMN ID_PRODUKTU IN TABLE PRODUKTY AS IT USES SEQUENCE.
');
```

END;

/

-

```

- TRIGGER FOR SEQUENCE KATEGORIASEQ FOR COLUMN ID_KATEGORII IN TABLE KATEGORIE
-----
CREATE OR REPLACE TRIGGER TS_KATEGORIE_KATEGORIASEQ BEFORE INSERT
ON KATEGORIE FOR EACH ROW
BEGIN
```



```

        :NEW.ID_KATEGORII := KATEGORIASEQ.NEXTVAL;
END;
/
CREATE OR REPLACE TRIGGER TSU_KATEGORIE_KATEGORIASEQ AFTER UPDATE OF ID_KATEGORII
ON KATEGORIE FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-
20010, 'CANNOT UPDATE COLUMN ID_KATEGORII IN TABLE KATEGORIE AS IT USES SEQUENCE. ');
END;
/

```

```

-
- TRIGGER FOR SEQUENCE STANOWISKASEQ FOR COLUMN ID_STANOWISKO IN TABLE STANOWISKA -----
CREATE OR REPLACE TRIGGER TS_STANOWISKA_STANOWISKASEQ BEFORE INSERT
ON STANOWISKA FOR EACH ROW
BEGIN
    :NEW.ID_STANOWISKO := STANOWISKASEQ.NEXTVAL;
END;
/
CREATE OR REPLACE TRIGGER TSU_STANOWISKA_STANOWISKASEQ AFTER UPDATE OF ID_STANOWISKO
ON STANOWISKA FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-
20010, 'CANNOT UPDATE COLUMN ID_STANOWISKO IN TABLE STANOWISKA AS IT USES SEQUENCE. ');
END;
/

```

```

-- CREATE FOREIGN KEYS (RELATIONSHIPS) SECTION -----
-----

```

```

ALTER TABLE PRACOWNICY ADD CONSTRAINT ZATRUDNIA_PRACOWNIKA FOREIGN KEY (ID_HURTOWNI)
REFERENCES HURTOWNIE_CZESCI_ELEKTRYCZNYCH (ID_HURTOWNI)
/

```

```

ALTER TABLE WYNAGRODZENIA ADD CONSTRAINT OTRZYMUJE_WYNAGRODZENIE FOREIGN KEY (ID_PRACOWNIKA)
REFERENCES PRACOWNICY (ID_PRACOWNIKA)
/

```

```
ALTER TABLE KLIENCI ADD CONSTRAINT POSIADA_KLIENTA FOREIGN KEY (ID_HURTOWNI) REFERENCES HURTOWNIE_CZESCI_ELEKTRYCZNYCH (ID_HURTOWNI)
/
```

```
ALTER TABLE DOSTAWY ADD CONSTRAINT OTRZYMUJE_DOSTAWIE FOREIGN KEY (ID_HURTOWNI) REFERENCES HURTOWNIE_CZESCI_ELEKTRYCZNYCH (ID_HURTOWNI)
/
```

```
ALTER TABLE POJAZDY ADD CONSTRAINT POSIADA_POJAZD FOREIGN KEY (ID_HURTOWNI) REFERENCES HURTOWNIE_CZESCI_ELEKTRYCZNYCH (ID_HURTOWNI)
/
```

```
ALTER TABLE WLASCICIELE ADD CONSTRAINT JEST_ZARZADZANA_PRZEZ FOREIGN KEY (ID_HURTOWNI) REFERENCES HURTOWNIE_CZESCI_ELEKTRYCZNYCH (ID_HURTOWNI)
/
```

```
ALTER TABLE KLIENCI ADD CONSTRAINT ZAMIESZKUJA FOREIGN KEY (ID_ADRESU) REFERENCES ADRESY (ID_ADRESU)
/
```

```
ALTER TABLE PRACOWNICY ADD CONSTRAINT MIESZKA_POD_ADRESEM FOREIGN KEY (ID_ADRESU) REFERENCES ADRESY (ID_ADRESU)
/
```

```
ALTER TABLE WLASCICIELE ADD CONSTRAINT ZAMELDOWANI_SA FOREIGN KEY (ID_ADRESU) REFERENCES ADRESY (ID_ADRESU)
/
```

```
ALTER TABLE HURTOWNIE_CZESCI_ELEKTRYCZNYCH ADD CONSTRAINT ZNAJDUJE_SIE FOREIGN KEY (ID_ADRESU) REFERENCES ADRESY (ID_ADRESU)
/
```

```
ALTER TABLE PRODUKTY ADD CONSTRAINT POSIADA_PRODUKTY FOREIGN KEY (ID_HURTOWNI) REFERENCES HURTOWNIE_CZESCI_ELEKTRYCZNYCH (ID_HURTOWNI)
/
```

```
ALTER TABLE SZCZEGOLY_DOSTAWY ADD CONSTRAINT DOSTAWA_MA_PRODUKTY FOREIGN KEY (ID_DOSTAWY) REFERENCES DOSTAWY (ID_DOSTAWY)
/
```

```
ALTER TABLE SZCZEGOLY_DOSTAWY ADD CONSTRAINT PRODUKT_JEST_W_DOSTAWIE FOREIGN KEY (ID_PRODUKTU) REFERENCES PRODUKTY (ID_PRODUKTU)
/
```

```
ALTER TABLE ZAMOWIENIA ADD CONSTRAINT SKLADA_ZAMOWIENIE FOREIGN KEY (ID_KLIENTA) REFERENCES KLIENCI (ID_KLIENTA)
/
```

```
ALTER TABLE SZCZEGOLY_ZAMOWIENIA ADD CONSTRAINT DOTYCZY FOREIGN KEY (ID_ZAMOWIENIA) REFERENCES ZAMOWIENIA (ID_ZAMOWIENIA)
/
```

```
ALTER TABLE SZCZEGOLY_ZAMOWIENIA ADD CONSTRAINT PRODUKT_JEST_W_ZAM FOREIGN KEY (ID_PRODUKTU) REFERENCES PRODUKTY (ID_PRODUKTU)
/
```

```
ALTER TABLE POSIADANE_KATEGORIE ADD CONSTRAINT KATEGORIA_JEST_POSIADANA FOREIGN KEY (NUMER_KATEGORII) REFERENCES KATEGORIE (ID_KATEGORII)
/
```

```
ALTER TABLE POSIADANE_KATEGORIE ADD CONSTRAINT KIEROWCA_MA_KATEGORIE FOREIGN KEY (ID_PRACOWNIKA) REFERENCES KIEROWCY (ID_PRACOWNIKA)
/
```

```
ALTER TABLE PRACOWNICY ADD CONSTRAINT PRACOWNIK_MA_STANOWISKO FOREIGN KEY (ID_STANOWISKO) REFERENCES STANOWISKA (ID_STANOWISKO)
/
```

## 5.4. Przykłady zapytań i poleceń SQL

5.4.1 Wypisanie ID\_Pracownika, imienia oraz nazwisko Magazynierów którzy mają kurs na wózki widłowe

```
SELECT p.ID_Pracownika, p.Imie, p.Nazwisko FROM Pracownicy p
JOIN Magazynierzy m ON m.ID_Pracownika = p.ID_Pracownika
WHERE m.czy_kurs_na_wozki_widlowe = 'T';
```

5.4.2 Wypisanie ID\_Klienta, Nazwy firmy klienta oraz NIP, którzy opłacili swoje zamówienie

```
SELECT k.ID_Klienta, k.Nazwa_Firmy, k.NIP FROM Klienci k
JOIN Zamowienia z ON z.ID_Klienta = k.ID_Klienta
Where z.StatusOpłaczone = 'T';
```

5.4.2 Wypisanie imienia oraz nazwiska kierowców posiadających kategorii prawa jazdy 'B' lub 'C'

```
SELECT p.Imie, p.Nazwisko FROM Pracownicy p
JOIN Posiadane_Kategorie m ON m.ID_Pracownika = p.ID_Pracownika
JOIN Kategorie k ON k.id_kategorii = m.ID_kategorii
WHERE Kod_Kategorii IN ('B', 'C');
```

## 5.5. Skrypt uzupełniający bazę danych

```
INSERT INTO Adresy  
VALUES(null, 'Warszawa', 189, '34C', '00-791', 'Chocimska');
```

```
INSERT INTO Adresy  
VALUES(null, 'Warszawa', 28, '167', '01-475', 'Radziejowicka');
```

```
INSERT INTO Adresy  
VALUES(null, 'Warszawa', 45, '21A', '01-376', 'Dzwigowa');
```

```
INSERT INTO Adresy  
VALUES(null, 'Warszawa', 89, '78C', '01-934', 'Opalin');
```

```
INSERT INTO Adresy  
VALUES(null, 'Warszawa', 56, 78, '03-085', 'Lidzbarska');
```

```
INSERT INTO Adresy  
VALUES(null, 'Warszawa', 11, 13, '04-413', 'Gorna');
```

```
INSERT INTO Adresy  
VALUES(null, 'Warszawa', 12, 193, '03-144', 'Swiatowida');
```

```
INSERT INTO Hurtownie_czesci_elektrycznych  
VALUES(null, 9823567411, TO_DATE('1990-04-23'), 'ElectroTest', 1);
```

```
INSERT INTO Stanowiska  
VALUES(null, 'Kierowca', 'Dosarcza zamowienia klientom');
```

```
INSERT INTO Stanowiska  
VALUES(null, 'Magazynier', 'Odpowiedzialny jest za odbieranie dostaw oraz kompletowanie zamowien');
```

```
INSERT INTO Pracownicy  
VALUES (null, 'Joanna', 'Kowalczyk', 'K', TO_DATE('1998-08-14'), 98081437890, TO_DATE('2019-11-19'), 'UMOWA O DZIELO', 693456234, 48114020040000367201355387, 1, 2, 1);
```

```
INSERT INTO Pracownicy  
VALUES (null, 'Dominik', 'Atlasner', 'M', TO_DATE('1998-09-29'), 98092937890, TO_DATE('2019-03-10'), 'UMOWA O PRACE', 693856123, 27114020040000367201355387, 1, 3, 1);
```

```
INSERT INTO Pracownicy
VALUES (null, 'Mateusz', 'Jaczewski', 'M', TO_DATE('1997-12-
28'), 97122838564, TO_DATE('2017-03-
11'), 'UMOWA O DZIELO', 693456345, 29114020002400367201355387, 1, 4, 2);
```

```
INSERT INTO Pracownicy
VALUES (null, 'Mateusz', 'Jasinski', 'M', TO_DATE('1996-03-
28'), 96032811938, TO_DATE('2019-01-
06'), 'UMOWA O PRACE', 723567987, 42114029940000367201355387, 1, 5, 2);
```

```
INSERT INTO Magazynierzy
VALUES(3, 'T', TO_DATE('2018-10-03'));
```

```
INSERT INTO Magazynierzy
VALUES(4, 'T', TO_DATE('2018-07-11'));
```

```
INSERT INTO Produkty
VALUES (null, 'tranzystor MOS', 28, 1.5, 23, 1);
```

```
INSERT INTO Produkty
VALUES (null, 'czujnik Temperatury', 56, 5, 23, 1);
```

```
INSERT INTO KLIENCI
VALUES (null, 5623513411, 'ELTOM', 734893412, 15, 'eltom@gmail.com', 1, 6);
```

```
INSERT INTO KLIENCI
VALUES (null, 5637903411, 'Ronson', 732145709, 10, 'ronson@o2.com', 1, 7);
```

```
INSERT INTO Zamowienia
VALUES (null, TO_DATE('2019-12-12'), '2019/12/12/A456', TO_DATE('2019-12-
11'), 'PRZELEW', 'T', 1);
```

```
INSERT INTO Zamowienia
VALUES (null, TO_DATE('2019-07-15'), '2019/07/15/B345', TO_DATE('2019-07-
17'), 'GOTOWKA', 'T', 2);
```

```
INSERT INTO Kierowcy
VALUES (1, TO_DATE('2017-01-19'), TO_DATE('2022-01-19'));
```

```
INSERT INTO Kierowcy
VALUES (2, TO_DATE('2017-06-11'), TO_DATE('2022-06-11'));
```

```
INSERT INTO Kategorie
VALUES (null, 'B', 'Kategoria umożliwia prowadzenie samochodów osobowych nieprze-
kraczających 3,5 tony');
```

```
INSERT INTO Kategorie
VALUES (null, 'C', 'Kategoria umożliwia prowadzenie samochodów ciężarowych');
```

```
INSERT INTO Posiadane_kategorie  
VALUES (1,1);
```

```
INSERT INTO Posiadane_kategorie  
VALUES (2,2);
```