



FIRMA KURIERSKA

DELIVERX

FAST CHEAP PRECISELY

autor: Łukasz Czepielik
WBMil Informatyka sem.III

Spis treści

1) Wymagania sprzętowe:	3
2) Opis mikroświata:	4
3) Dokumentacja modelu ERD:	6
4) Opis poszczególnych tabel:	7
Klient	7
Pracownik	9
Przesyłka	11
Typ_Przesyłki	13
Typ_Płatności	13
Status	14
Historia	15
Kurs	16
Pojazd	17
Typ_Pojazdu	18
Uprawnienie	18
Ubezpieczenie	19
Typ_Ubezpieczenia	20
Firma_Ubezpieczeniowa	21
5) Opis poszczególnych wiązań pomiędzy tabelami:	23
Historia ↔ Status	23
Przesyłka ↔ Klient ↔ Historia ↔ Typ_Przesyłki ↔ Typ_Płatności	24
Przesyłka ↔ Kurs	25
Kurs ↔ Pracownik ↔ Pojazd	26
Pracownik ↔ Uprawnienie ↔ Typ_Pojazdu	27
Pojazd ↔ Ubezpieczenie ↔ Typ_Pojazdu	28
Ubezpieczenie ↔ Firma_Ubezpieczeniowa ↔ Typ_Ubezpieczenia	29
6) Wybrane zapytania:	30
Zapytania „INSERT”	30
Zapytania „SELECT”	33
Widoki	39

1) Wymagania sprzętowe:

Niezbędnym narzędziem pozwalającym zarządzać bazą danych jest zainstalowane oprogramowanie Microsoft SQL Server Management Studio (MS SSMS) o wymaganiach systemowych podanych poniżej.

<u>Komponent:</u>	<u>Wymagania:</u>
Pamięć RAM	Minimum: MS SSMS Express: 256 MB Zalecane: MS SSMS Express: 1024 MB
Zegar procesora	Minimum: MS SSMS Express 32-bit: 1.0 GHz MS SSMS Express 64-bit: 1.4 GHz Zalecane: 2.0 GHz lub szybszy
Dysk twardy	MS SSMS wymaga przynajmniej 2 GB wolnej przestrzeni na dysku.
System operacyjny	MS SSMS Express 32-bit: system operacyjny Windows XP SP2 lub nowszy. MS SSMS Express 64-bit: system operacyjny Windows Vista lub nowszy.
Monitor	MS SSMS wymaga monitora obsługującego rozdzielczość 800x600 lub wyższą.

2) Opis mikroświata:

Baza danych została stworzona w celu ułatwienia zarządzania firmą kurierską. Głównym zadaniem systemu bazodanowego jest przechowywanie w uporządkowany sposób informacji dotyczących firmy, aby w razie potrzeby szybko znaleźć niezbędne informacje.

Firma kurierska jest przedsiębiorstwem świadczącym pewne konkretne usługi swoim odbiorcom, w konsekwencji baza dotycząca takiej działalności musi pozwalać odwzorowywać zachodzące realne procesy biznesowe.

System bazodanowy musi potrafić obsłużyć proces przekazania przesyłki pomiędzy nadawcą a odbiorcą, który wygląda następująco:

- Klient chcąc nadać paczkę udaje się do siedziby firmy wraz z obiektem, który podlegać będzie transportowi. W siedzibie firmy klient kontaktuje się z pracownikiem odpowiedzialnym za przyjmowanie zleceń. Dane personalne klienta oraz informacje dotyczące paczki zostają zapisane przez pracownika do systemu bazodanowego. Ważnymi informacjami dla przedsiębiorcy są dane dotyczące typu przesyłki oraz typu płatności za nią, a także informacje dotyczące wymaganej daty dostarczenia pakunku.
- Klient zadowolony z nadania przesyłki opuszcza siedzibę firmy, natomiast paczka otrzymuje swój pierwszy status w drodze do swojego odbiorcy. To daje początek historii, która gromadzi wszystkie statusy, przez które paczka będzie przechodzić.
- Paczka klienta zostaje przydzielona do konkretnego kursu zrzeszającego również inne przesyłki, który będzie prowadzony przez konkretnego pracownika odpowiedzialnego za ich transport.
- Pracownik odpowiedzialny za dany kurs pobiera pojazd, który aktualnie nie jest w trasie, przy pomocy którego dokona transportu, po czym wyrusza w drogę, aby dostarczyć przesyłki. Po dostarczeniu każdej pojedynczej przesyłki odnotowuje w systemie dokładną datę jej doręczenia oraz w przypadku pewnych form płatności pobiera od odbierającego stosowną opłatę.

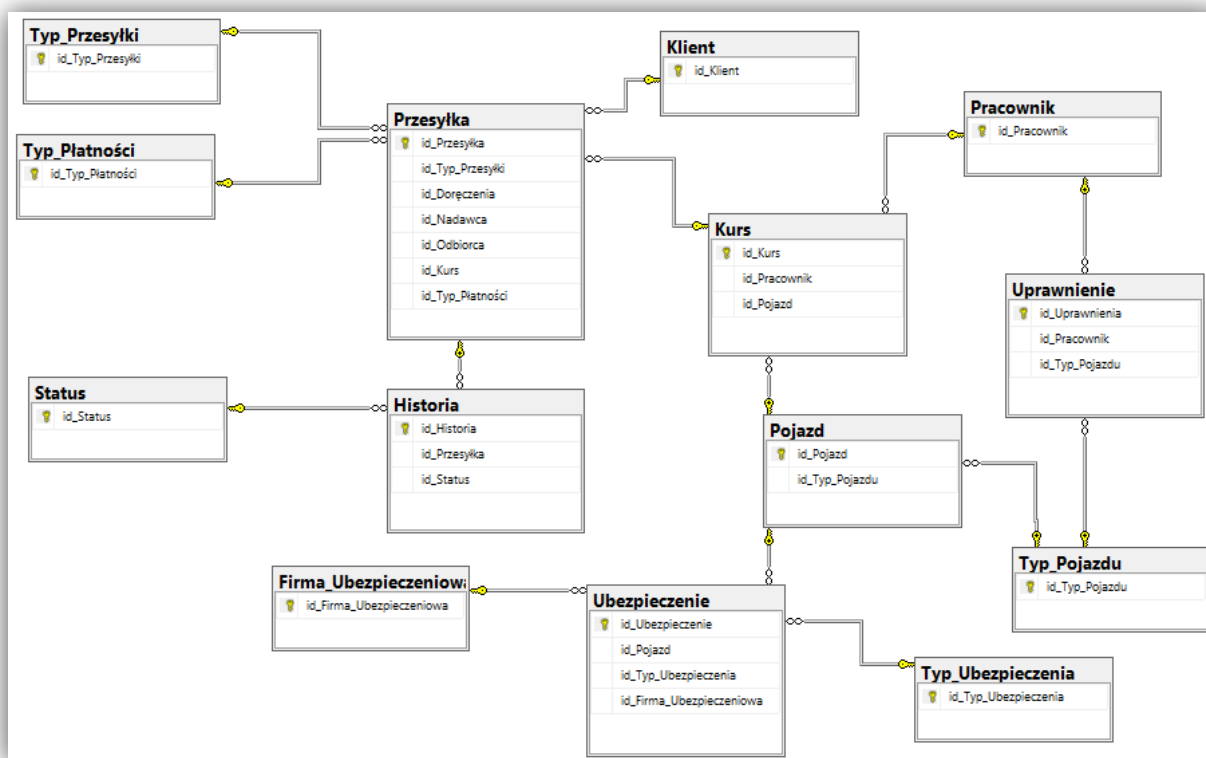
Uwagi

- Pojazd musi zostać dopasowany do wymagań kursu, czyli wymagań dotyczących możliwości transportu ładunku o pewnej wadze.
- W bazie danych muszą znajdować się informacje na temat uprawnień posiadanych przez pracowników. Pracownik nie może wykonywać kursu, który wymaga wykorzystania pojazdu, na który nie posiada uprawnień.
- W bazie danych znajdują się informacje dotyczące floty pojazdów, którą dysponuje firma. Co za tym idzie, każdy ze znajdujących się tam środków transportu musi posiadać ważne ubezpieczenie OC oraz może posiadać inne opcjonalne ubezpieczenia, np. zawartości ładunku. W konsekwencji, w bazie przedsiębiorstwa powinna znajdować się informacja dotycząca ubezpieczeń wykupionych u ubezpieczeniodawców, a konkretniej mówiąc, informacja dotycząca firm ubezpieczeniowych, które świadczą naszemu przedsiębiorstwu usługi ubezpieczenia, a także dokładne dane dotyczące tego jakie pojazdy posiadają konkretne typy ubezpieczeń.



3) Dokumentacja modelu ERD:

Baza danych przedstawiona za pomocą uproszczonego schematu, na którym znajdują się klucze główne, klucze obce oraz wiązania pomiędzy tabelami.



Schemat 1 - Diagram ogólny, uwzględnione wyłącznie klucze główne oraz obce

4) Opis poszczególnych tabel:

Klient

Key	Column Name	Data Type	Allow Nulls	Default Value	Constraint
PK	id_Klient	int	NO		
	Imię	nchar(30)	NO		
	Nazwisko	nchar(30)	NO		
	Państwo	nchar(20)	NO		
	Miasto	nchar(30)	NO		
	Kod_pocztowy	nchar(6)	NO		00-000
	Ulica	nchar(30)	NO		
	[Nr_domu/mieszkania]	nchar(10)	NO		
	Nr_telefonu	nchar(13)	NO		+00 000000000
	[Adres_e-mail]	nchar(50)	YES		

Tabela „**Klient**” przechowuje istotne informacje dotyczące osób, które korzystają z usług firmy. Znajdują się tutaj pola, które opisują dane personalne oraz dane dotyczące miejsca zamieszkania. Z punktu widzenia klienta dużą rolę odgrywa numer telefonu kontaktowego („*Nr_telefonu*”), który pozwala na komunikację z klientem.

Kluczem głównym tabeli jest numer identyfikacyjny klienta: „*id_Klient*”.

Brak kluczy obcych.

Na tabeli zdefiniowane są następujące maski wprowadzania:

a) Maska wprowadzania dla pola „Nr_telefonu”:

▼ (General)	
Expression	(([Nr_telefonu] like '+[0-9][0-9] [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))
▼ Identity	
(Name)	Numer_telefonu
Description	Format numeru telefonu +99 999999999

```
(([Nr_telefonu] like '+[0-9][0-9] [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))
```

b) Maska wprowadzania dla pola „Kod_pocztowy”:

▼ (General)	
Expression	(([Kod_pocztowy] like '[0-9][0-9]-[0-9][0-9][0-9]'))
▼ Identity	
(Name)	Kod_pocztowy
Description	Format kodu pocztowego 99-999

Pracownik

Key	Column Name	Data Type	Allow Nulls	Default Value	Constraint
PK	id_Pracownik	int	NO		
	Imię	nchar(30)	NO		
	Nazwisko	nchar(30)	NO		
	PESEL	nchar(11)	NO		
	Data_urodzenia	date	NO		
	Data_zatrudnienia	date	NO		>[Data_urodzenia]
	Państwo	nchar(20)	NO		
	Miasto	nchar(30)	NO		
	Kod_pocztowy	nchar(6)	NO		00-000
	Ulica	nchar(20)	NO		
	[Nr_domu/mieszkania]	nchar(10)	NO		
	Nr_telefonu	nchar(13)	NO		+00 000000000
	[Adres_e-mail]	nchar(50)	YES		

Tabela „**Pracownik**” przechowuje istotne informacje dotyczące osób, które są zatrudnione w firmie. Znajdują się tutaj pola, które opisują dane personalne oraz dane dotyczące miejsca zamieszkania, czyli wszystko to, co pracodawca powinien wiedzieć o zatrudnianej przez niego osobie.

Kluczem głównym tabeli jest numer identyfikacyjny klienta: „id_Pracownik”.

Brak kluczy obcych.

Na tabeli zdefiniowane są następujące maski wprowadzania:

a) Maska wprowadzania dla pola „PESEL”:

▼ (General)	
Expression	((PESEL] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
▼ Identity	
(Name)	PESEL
Description	Format numeru PESEL

```
((PESEL] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
```

b) Maska wprowadzania dla pola „Kod_pocztowy”:

▼ (General)	
Expression	((Kod_pocztowy] like '[0-9][0-9]-[0-9][0-9][0-9]')
▼ Identity	
(Name)	Kod_pocztowy2
Description	Format kodu pocztowego 99-999

c) Maska wprowadzania dla pola „Nr_telefonu”:

▼ (General)	
Expression	((Nr_telefonu] like '+[0-9][0-9] [0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
▼ Identity	
(Name)	Numer_telefonu2
Description	Format numeru telefonu +99 999999999

```
((Nr_telefonu] like '+[0-9][0-9] [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
```

d) Maska wprowadzania dla pola „Data_zatrudnienia”:

▼ (General)	
Expression	[[Data_zatrudnienia]>[Data_urodzenia]]
▼ Identity	
(Name)	Data4
Description	Data zatrudnienia musi być większa niż data urodzenia

Przesyłka

Key	Column Name	Data Type	Allow Nulls	Default Value	Constraint
PK	id_Przesyłka	int	NO		
FK	id_Typ_Przesyłki	int	NO		
FK	id_Doręczenia	int	NO		
FK	id_Nadawca	int	NO		
FK	id_Odbiorca	int	NO		
FK	id_Kurs	int	NO		
	Data_nadania	smalldate time	NO	GETDATE()	
	Data_wymaganego_dostarczenia	smalldate time	NO		>[Data_nadania]
	Data_rzeczywistego_dostarczenia	smalldate time	YES		>[Data_nadania]
	Waga	float	NO		
	Jednostka_wagi	nchar(5)	NO	„KG”	
	Opłata	small money	NO		>=0
	Waluta	nchar(5)	NO	„PLN”	
	Nr_dokumentu_przewozowego	nchar(30)	NO		
FK	id_Typ_Płatności	int	NO		

Tabela „**Przesyłka**” przechowuje informacje dotyczące pojedynczej konkretnej paczki w czasie całego procesu jej transportu z punktu A do punktu B. Znajdują się tutaj pole identyfikujące odbiorcę(„*id_Odbiorca*”), nadawcę(„*id_Nadawca*”) oraz miejsce doręczenia paczki(„*id_Doręczenia*”). W tabeli widnieją również pola określające wagę przesyłki(„*Waga*”) oraz opłatę za transport(„*Opłata*”) w przypadku, gdy transport nie jest opłacony z góry. Ważnym polem relacji jest numer dokumentu przewozowego(„*Nr_dokumentu_przewozowego*”) oraz rodzaj typu płatności(„*id_Typ_Płatności*”).

Kluczem głównym tabeli jest numer identyfikacyjny przesyłki: „*id_Przesyłka*”.

W tabeli występują następujące klucze obce:

- a) „*id_Typ_Przesyłki*” tworzy wiązanie z tabelą „**Typ_Przesyłki**”.
- b) „*id_Doręczenia*” tworzy wiązanie z tabelą „**Klient**”.
- c) „*id_Nadawca*” tworzy wiązanie z tabelą „**Klient**”.
- d) „*id_Odbiorca*” tworzy wiązanie z tabelą „**Klient**”.
- e) „*id_Kurs*” tworzy wiązanie z tabelą „**Kurs**”.
- f) „*id_Typ_Płatności*” tworzy wiązanie z tabelą „**Typ_Płatności**”.

Na tabeli zdefiniowane są następujące maski wprowadzania:

- a) Maską wprowadzania dla pola „*Opłata*”:

▼ (General)	
Expression	[[Opłata] >= 0]
▼ Identity	
(Name)	Opłata
Description	Wartość nieujemna

- b) Maską wprowadzania dla pola „*Data_wymaganego_dostarczenia*”:

▼ (General)	
Expression	[[Data_wymaganego_dostarczenia]>[Data_nadania]]
▼ Identity	
(Name)	Data2
Description	Czas na dostarczenie

c) Maska wprowadzania dla pola „Data_rzeczywistego_dostarczenia”:

▼ (General)	
Expression	[[Data_rzeczywistego_dostarczenia]>[Data_nadania]]
▼ Identity	
(Name)	Data3
Description	Realny czas dostarczenia paczki

Typ_Przesyłki

Key	Column Name	Data Type	Allow Nulls	Default Value	Constraint
PK	id_Typ_Przesyłki	int	NO		
	Typ	nchar(20)	NO		

Tabela słownikowa „**Typ_Przesyłki**” przechowuje typy przesyłek, jakie mogą być transportowane przez firmę. Przykładowo, czy jest to przesyłka priorytetowa czy ekonomiczna oraz wiele innych wariantów.

Kluczem głównym jest numer identyfikacyjny przesyłki: „id_Typ_Przesyłki”.

Brak kluczy obcych.

Brak zdefiniowanych masek wprowadzania.

Typ_Płatności

Key	Column Name	Data Type	Allow Nulls	Default Value	Constraint
PK	id_Typ_Płatności	int	NO		
	Typ	nchar(20)	NO		

Tabela słownikowa „**Typ_Płatności**” przechowuje typy płatności, jakie mogą zostać wybrane przez klienta. Przykładowo, może to być płatność z góry lub za pobraniem u kuriera. Podobnie jak przy poprzedniej tabeli wariantów zazwyczaj jest wiele.

Kluczem głównym jest numer identyfikacyjny przesyłki: „id_Typ_Płatności”.

Brak kluczy obcych.

Brak zdefiniowanych masek wprowadzania.

Status

Key	Column Name	Data Type	Allow Nulls	Default Value	Constraint
PK	id_Status	int	NO		
	Nazwa	nchar(20)	NO		

Tabela słownikowa „**Status**” przechowuje statusy jakie paczka może otrzymać w swojej drodze do punktu docelowego. Przykładowymi statusami mogą być: „oczekuje na realizację”, „dostarczona”. Podobnie jak w przypadku wcześniejszych tabel, możliwa jest bardzo duża mnogość różnych statusów.

Kluczem głównym jest numer identyfikacyjny przesyłki: „id_Status”.

Brak kluczy obcych.

Brak zdefiniowanych masek wprowadzania.

Historia

Key	Column Name	Data Type	Allow Nulls	Default Value	Constraint
PK	id_Historia	int	NO		
FK	id_Przesyłka	int	NO		
FK	id_Status	int	NO		
	Data_zmiany_statusu	smalldate time	NO	GETDATE()	
	Opis	nchar(50)	YES		

Tabela „**Historia**” przechowuje wszystkie statusy paczek jakie zostały odnotowane podczas dokonywania transportu. W tabeli znajdziemy takie informacje jak: numer identyfikujący przesyłki („*id_Przesyłka*”), numer statusu („*id_Status*”) jaki otrzymała oraz konkretną datę wraz z uwzględnieniem czasu, kiedy zmiana statusu nastąpiła („*Data_zmiany_statusu*”). W razie potrzeby można również zamieścić dodatkowy opis („*Opis*”). Tabela ma za zadanie zbierać w jednym miejscu wszystkie statusy, które odnoszą się do konkretnych przesyłek.

Kluczem głównym jest numer identyfikacyjny przesyłki: „*id_Historia*”.

W tabeli występują następujące klucze obce:

- a) „*id_Przesyłka*” tworzy wiązanie z tabelą „**Przesyłka**”.
- b) „*id_Status*” tworzy wiązanie z tabelą „**Status**”.

Brak zdefiniowanych masek wprowadzania.

Kurs

Key	Column Name	Data Type	Allow Nulls	Default Value	Constraint
PK	id_Kurs	int	NO		
	id_Pracownik	int	NO		
	id_Pojazd	int	NO		
	Data	smalldate time	NO	GETDATE()	
	Liczba_kilometrów	float	NO		
	Uwagi	nchar(50)	YES		

Tabela „**Kurs**” przechowuje informacje dotyczące pojedynczych transportów, które są wykonywane przez konkretnych pracowników przy wykorzystaniu danych pojazdów. W tabeli znajdują się ponadto informacje dotyczące daty rozpoczęcia kursu („*Data*”) oraz liczby kilometrów niezbędnych do ukończenia kursu („*Liczba_kilometrów*”).

Kluczem głównym jest numer identyfikacyjny przesyłki: „id_Kurs”.

W tabeli występują następujące klucze obce:

- a) „id_Pracownik” tworzy wiązanie z tabelą „**Pracownik**”.
- b) „id_Pojazd” tworzy wiązanie z tabelą „**Pojazd**”.

Brak zdefiniowanych masek wprowadzania.

Pojazd

Key	Column Name	Data Type	Allow Nulls	Default Value	Constraint
PK	id_Pojazd	int	NO		
	Marka	nchar(20)	NO		
	Model	nchar(20)	NO		
	Nr_rejestracyjny	nchar(10)	NO		
FK	id_Typ_Pojazdu	int	NO		
	Max_ładowność	float	NO		
	Jednostka_ładowności	nchar(5)	NO	„KG”	
	Rocznik	date	NO	„tys.KM”	
	Stan_licznika	float	NO		
	Jednostka_licznika	nchar(8)	NO		
	Stan_licznika_na_dzień	date	NO	GETDATE()	
	Rodzaj_paliwa	nchar(10)	NO	„DIESEL”	

Tabela „**Pojazd**” przechowuje informacje dotyczące pojazdów wykorzystywanych w firmie. Znajdują się tutaj pola identyfikujące pojazd oraz kluczowe informacje takie jak maksymalna ładowność, która jest parametrem kluczowym przy wyborze odpowiedniego środka transportu do spełnienia zapotrzebowania konkretnego kursu przewozowego. Dodatkowymi danymi są tutaj pola określające stan licznika oraz data, w której dany stan został zanotowany.

Kluczem głównym jest numer identyfikacyjny przesyłki: „id_Pojazd”.

W tabeli występują następujące klucze obce:

a) „id_Typ_Pojazdu” tworzy wiązanie z tabelą „Typ_Pojazdu”.

Brak zdefiniowanych masek wprowadzania.

Typ_Pojazdu

Key	Column Name	Data Type	Allow Nulls	Default Value	Constraint
PK	id_Typ_Pojazdu	int	NO		
	Typ	nchar(30)	NO		

Tabela słownikowa „**Typ_Pojazdu**” przechowuje typy pojazdów jakie znajdują się w tabeli „**Pojazd**” oraz jest wykorzystywana do identyfikacji uprawnień posiadanych przez pracowników na konkretne kategorie pojazdów.

Kluczem głównym jest numer identyfikacyjny przesyłki:
„id_Typ_Pojazdu”.

Brak kluczy obcych.

Brak zdefiniowanych masek wprowadzania.

Uprawnienie

Key	Column Name	Data Type	Allow Nulls	Default Value	Constraint
PK	id_Uprawnienie	int	NO		
FK	id_Pracownik	int	NO		
FK	id_Typ_Pojazdu	int	NO		
	Nr_dokumentu	nchar(30)	NO		

Tabela „**Uprawnienie**” przechowuje informacje dotyczące uprawnień posiadanych przez pracowników. Pracownik posiada uprawnienie na dany typ pojazdu, co potwierdza znajdujący się w bazie danych numer dokumentu („*Nr_dokumentu*”).

Kluczem głównym jest numer identyfikacyjny przesyłki:
„id_Uprawnienie”.

W tabeli występują następujące klucze obce:

- a) „*id_Pracownik*” tworzy wiązanie z tabelą „**Pracownik**”.
- b) „*id_Pojazd*” tworzy wiązanie z tabelą „**Pojazd**”.

Brak zdefiniowanych masek wprowadzania.

Ubezpieczenie

Key	Column Name	Data Type	Allow Nulls	Default Value	Constraint
PK	id_Ubezpieczenie	int	NO		
FK	id_Pojazd	int	NO		
FK	id_Typ_Ubezpieczenia	int	NO		
FK	id_Firma_Ubezpieczeniowa	int	NO		
	Nr_polisy	nchar(20)	NO		
	Obowiązuje_od	date	NO	GETDATE()	
	Obowiązuje_do	date	NO		>[Obowiązuje_od]
	Cena	small money	NO		>=0
	Waluta	nchar(5)	NO	„PLN”	

Tabela „**Ubezpieczenie**” przechowuje informacje dotyczące ubezpieczeń pojazdów. Znajdują się w niej pola niezbędne do identyfikacji konkretnego pojazdu, typu ubezpieczenia z jakim mamy do czynienia oraz numerem identyfikacyjnym firmy, która świadczy nam tę usługę. Nieodzownym elementem jest pole zawierające numer identyfikujący polisę („*Nr_polisy*”). Bardzo ważna z punktu biznesowego jest kontrola nad ciągłością ubezpieczenia, dlatego też w tabeli znajdują się pola daty obowiązywania polisy. Dodatkową informacją zawartą w strukturze jest również cena usługi.

Kluczem głównym jest numer identyfikacyjny przesyłki:
„*id_Ubezpieczenie*”.

W tabeli występują następujące klucze obce:

- a) „*id_Pojazd*” tworzy wiązanie z tabelą „**Pojazd**”.
- b) „*id_Typ_Ubezpieczenia*” tworzy wiązanie z tabelą „**Typ_Ubezpieczenia**”.
- c) „*id_Firma_Ubezpieczeniowa*” tworzy wiązanie z tabelą „**Firma_Ubezpieczeniowa**”.

Na tabeli zdefiniowane są następujące maski wprowadzania:

- a) Maską wprowadzania dla pola „*Cena*”:

▼ (General)	
Expression	[[Cena]>=0]
▼ Identity	
(Name)	Cena
Description	Wartość nieujemna

- b) Maską wprowadzania dla pola „*Obowiązuje_do*”:

▼ (General)	
Expression	[[Obowiązuje_do]>[Obowiązuje_od]]
▼ Identity	
(Name)	Data1
Description	Okres ubezpieczenia

Typ_Ubezpieczenia

Key	Column Name	Data Type	Allow Nulls	Default Value	Constraint
PK	id_Typ_Ubezpieczenia	int	NO		
	Typ	nchar(20)	NO		

Tabela słownikowa „**Typ_Ubezpieczenia**” przechowuje typy ubezpieczeń jakie mogą zostać zakupione u ubezpieczyciela. Przykładowo, może to być ubezpieczenie OC, AC lub także ubezpieczenie przewożonego ładunku.

Kluczem głównym jest numer identyfikacyjny przesyłki: „id_Typ_Ubezpieczenia”.

Brak kluczy obcych.

Brak zdefiniowanych masek wprowadzania.

Firma_Ubezpieczeniowa

Key	Column Name	Data Type	Allow Nulls	Default Value	Constraint
PK	id_Pracownik	int	NO		
	Nazwa	nchar(30)	NO		
	NIP	nchar(13)	NO		000-000-00-00
	Państwo	nchar(20)	NO		
	Miasto	nchar(30)	NO		
	Kod_pocztowy	nchar(6)	NO		00-000
	Ulica	nchar(20)	NO		
	[Nr_budynku]	nchar(10)	NO		
	Nr_telefonu	nchar(15)	NO		+00 000000000
	[Adres_e-mail]	nchar(50)	YES		

Tabela „**Firma_Ubezpieczeniowa**” przechowuje informacje dotyczące firm ubezpieczeniowych, z których usług korzysta nasza firma przy ubezpieczaniu naszych pojazdów. Znajdują się tutaj pola, które opisują dane personalne oraz dane dotyczące położenia geograficznego kontrahenta. Z punktu widzenia przedsiębiorcy, ważnymi informacjami w tabeli są numer identyfikacyjny NIP oraz telefon kontaktowy do firmy („Nr_telefonu”).

Kluczem głównym jest numer identyfikacyjny przesyłki: „id_Firma_Ubezpieczeniowa”.

Brak kluczy obcych.

Na tabeli zdefiniowane są następujące maski wprowadzania:

a) Maska wprowadzania dla pola „*NIP*”:

▼ (General)	
Expression	(([NIP] like '[0-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]'))
▼ Identity	
(Name)	NIP
Description	Format numeru NIP 999-999-99-99

```
(([NIP] like '[0-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]'))
```

b) Maska wprowadzania dla pola „*Nr_telefonu*”:

▼ (General)	
Expression	(([Nr_telefonu] like '+[0-9][0-9] [0-9][0-9][0-9][0-9][0-9][0-9]'))
▼ Identity	
(Name)	Numer_telefonu1
Description	Format numeru telefonu +99 999999999

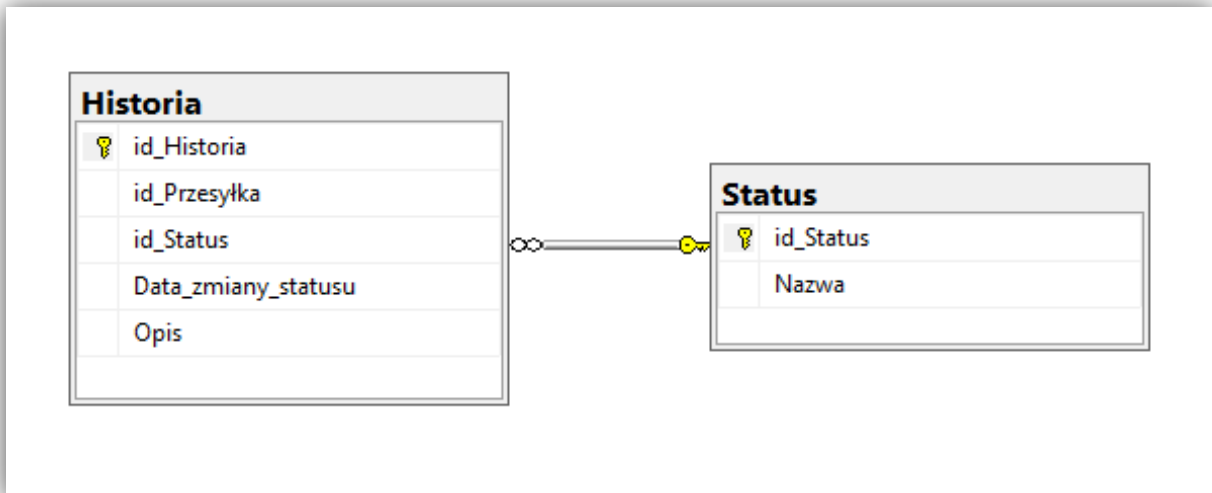
```
(([Nr_telefonu] like '+[0-9][0-9] [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))
```

c) Maska wprowadzania dla pola „*Kod_pocztowy*”:

▼ (General)	
Expression	(([Kod_pocztowy] like '[0-9][0-9]-[0-9][0-9][0-9]'))
▼ Identity	
(Name)	Kod_pocztowy1
Description	Format kodu pocztowego 99-999

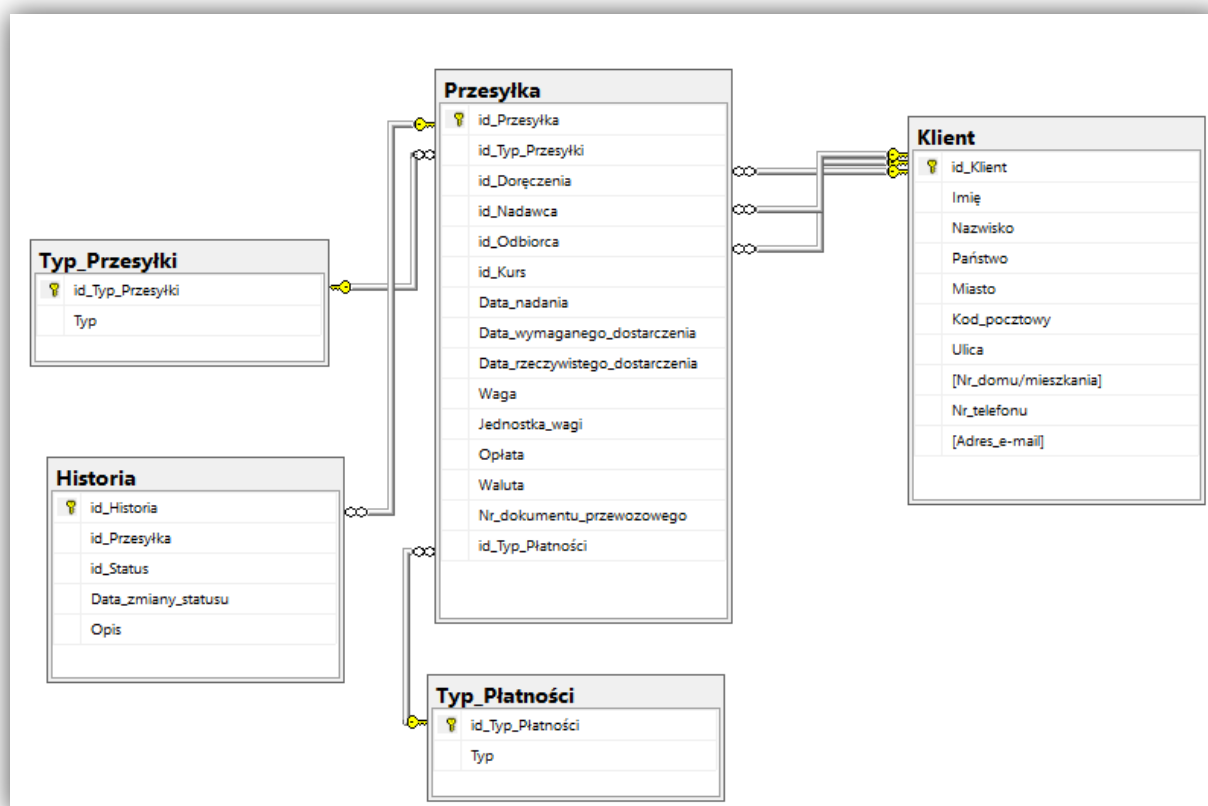
5) Opis poszczególnych wiązań pomiędzy tabelami:

Historia ↔ Status



Pomiędzy tabelami „**Historia**” i „**Status**” istnieje wiązanie jeden do wielu, ponieważ paczka w swoim procesie podróży może przyjmować wiele statusów. Co więcej, w tabeli „**Historia**” dany status będzie identyczny dla wielu rekordów, to znaczy będzie X przesyłek z danym statusem.

Przesyłka ↔ Klient ↔ Historia ↔
 Typ_Przesyłki ↔ Typ_Płatności



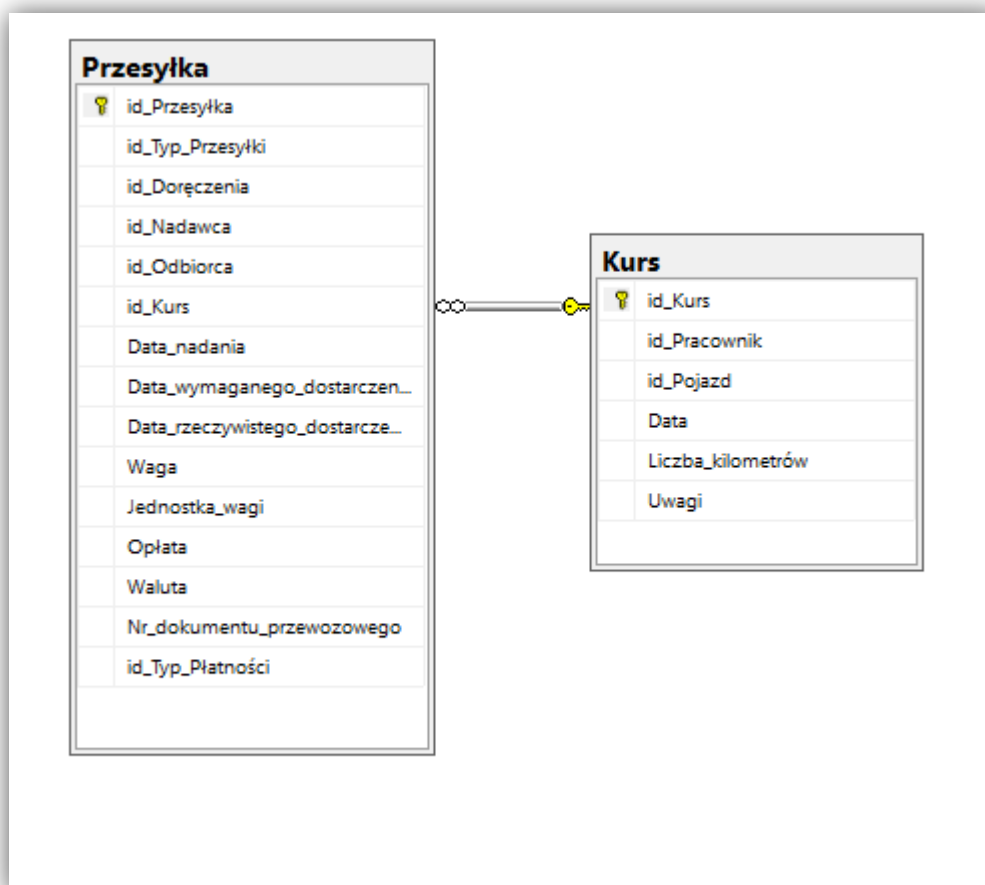
Pomiędzy tabelami „Przesyłka” i „Historia” istnieje wiązanie jeden do wielu. Jest to spowodowane tym, że przesyłka w trakcie swojej podróży będzie generowała wiele zmian statusów, które to właśnie są zapisywane w historii. W konsekwencji, w tabeli „Historia” będzie się znajdowało wiele rekordów poświęconych danemu konkretnemu pakunkowi.

Pomiędzy tabelami „Typ_Przesyłki” i „Przesyłka” oraz „Typ_Płatności” i „Przesyłka” występują wiązania jeden do wielu. Podyktowane jest to faktem, że w bazie danych znajdzie się wiele przesyłek, które będą miały identyczne typy przesyłek bądź takie same typy płatności za nie.

Pomiędzy tabelą „Klient” oraz „Przesyłka” znajdują się trzy wiązania jeden do wielu. Klucz główny z tabeli „Klient”, to jest „id_Klient”, jest połączony z trzema kluczami obcymi: „id_Doręczenia”, „id_Nadawca”, a także „id_Odbiorca”. Sytuacja wygląda tak, że dana pojedyncza osoba, która w zależności od przesyłki może być odbiorcą lub adresatem (czasami także odbiorcą i adresatem naraz) może w czasie działania firmy być wielokrotnie

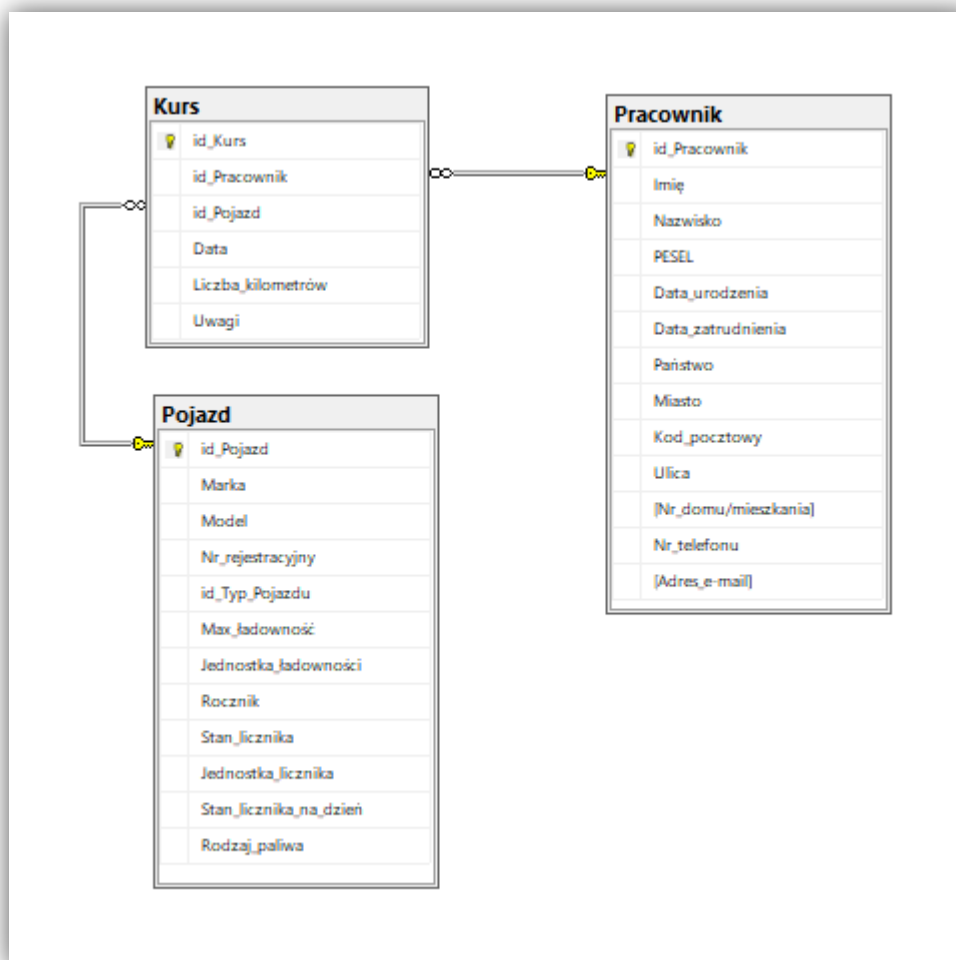
klientem korzystającym z usług transportowych występując w dowolnej roli podczas tego procesu. Pole „*id_Doręczenia*” jest dodatkowym zabezpieczeniem w przypadku, gdy adres odbioru jest niezwiązany z odbierającym paczkę. W każdym innym przypadku będzie widniał tam numer identyczny do „*id_Odbiorca*”. Taka sytuacja może zdarzyć się wtedy, gdy odbierającym jest osoba fizyczna przebywająca pod adresem innej osoby fizycznej. W takim przypadku, do naszej bazy należy wprowadzić adres oraz dane osoby, u której odbierający odbierze swoją przesyłkę.

Przesyłka ↔ Kurs



Pomiędzy tabelami „**Kurs**” oraz „**Przesyłka**” istnieje wiązanie jeden do wielu, ponieważ w ramach jednego kursu zazwyczaj transportowana jest dużo większa liczba paczek niż wyłącznie jedna sztuka.

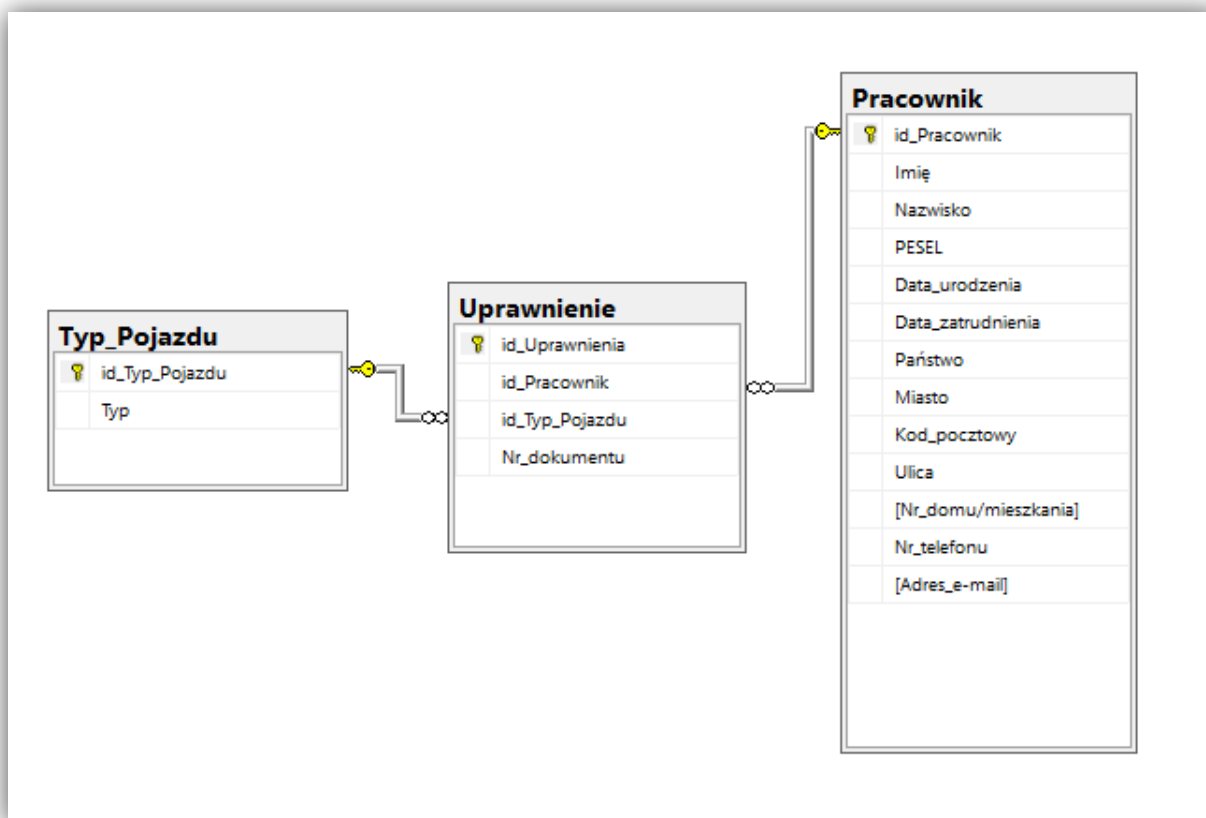
Kurs ↔ Pracownik ↔ Pojazd



Pomiędzy tabelami „**Pracownik**” i „**Kurs**” istnieje wiązanie jeden do wielu. Dany pracownik może wykonać w swojej karierze w firmie wiele różnych kursów, które co więcej, mogą odbyć się tego samego lub zupełnie innego dnia/tygodnia/miesiąca/roku.

Pomiędzy tabelami „**Pojazd**” i „**Kurs**” istnieje wiązanie jeden do wielu. Podobnie jak poprzednio wygląda sprawa z pojazdem, który w czasie okresu jego użytkowania w firmie może zostać wykorzystany do przejechania wielu kursów i podczas jego trwania być użytkowanym przez różnych pracowników.

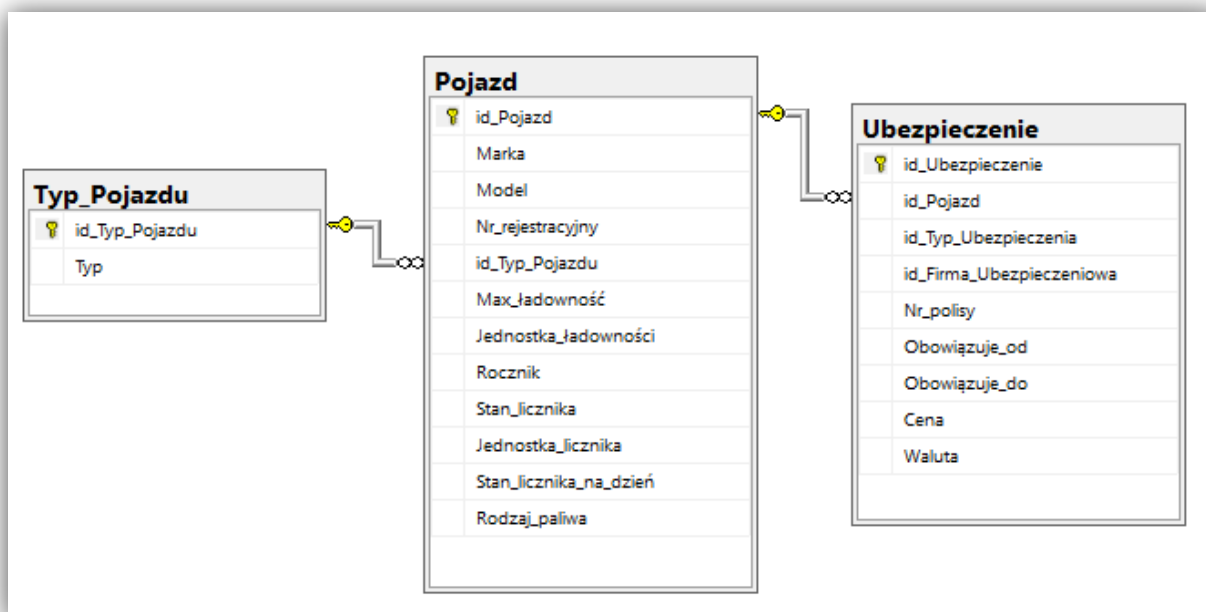
Pracownik \leftrightarrow Uprawnienie \leftrightarrow Typ_Pojazdu



Pomiędzy tabelami „**Typ_Pojazdu**” i „**Uprawnienie**” istnieje wiązanie jeden do wielu. Pojedynczy pracownik może mieć wiele uprawnień do kierowania różnymi typami pojazdów.

Pomiędzy tabelami „**Pracownik**” i „**Uprawnienie**” istnieje wiązanie jeden do wielu. W przypadku większej liczby pracowników zdarzeniem pewnym jest, że pewna ich grupa będzie miała uprawnienia na kilka różnych typów pojazdów.

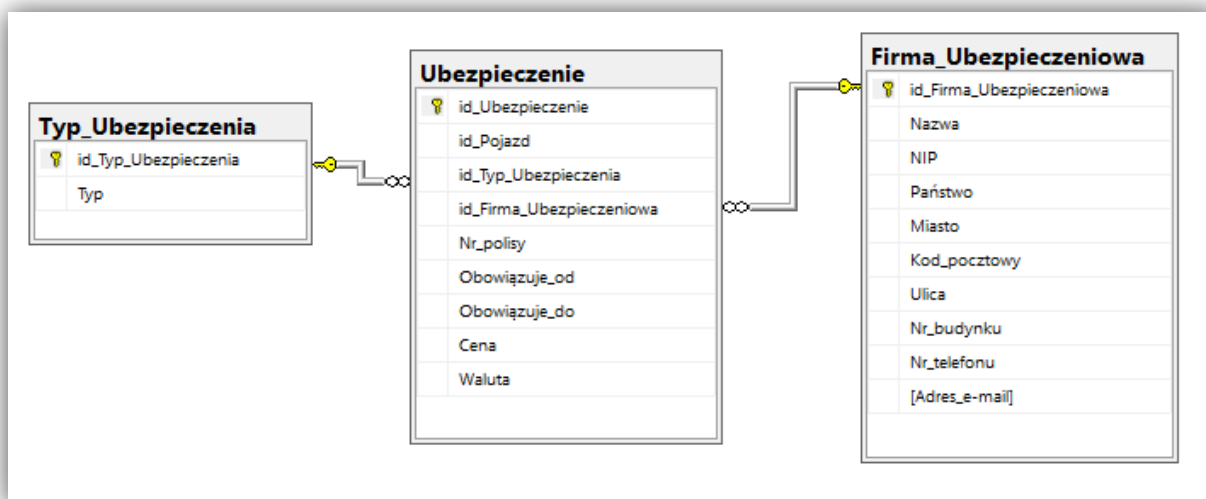
Pojazd ↔ Ubezpieczenie ↔ Typ_Pojazdu



Pomiędzy tabelami „**Typ_Pojazdu**” i „**Pojazd**” istnieje wiązanie jeden do wielu. Dzieje się tak, ponieważ w bazie możemy mieć wiele pojazdów tego samego typu, np. do 3.5 lub powyżej 3.5 tony dopuszczalnej masy całkowitej.

Pomiędzy tabelami „**Pojazd**” i „**Ubezpieczenie**” istnieje wiązanie jeden do wielu. Jeden pojazd może mieć wiele ubezpieczeń. Mogą być to zarówno ubezpieczenia, które jeszcze obowiązują oraz te już wygasłe.

Ubezpieczenie ↔ Firma_Ubezpieczeniowa ↔ Typ_Ubezpieczenia



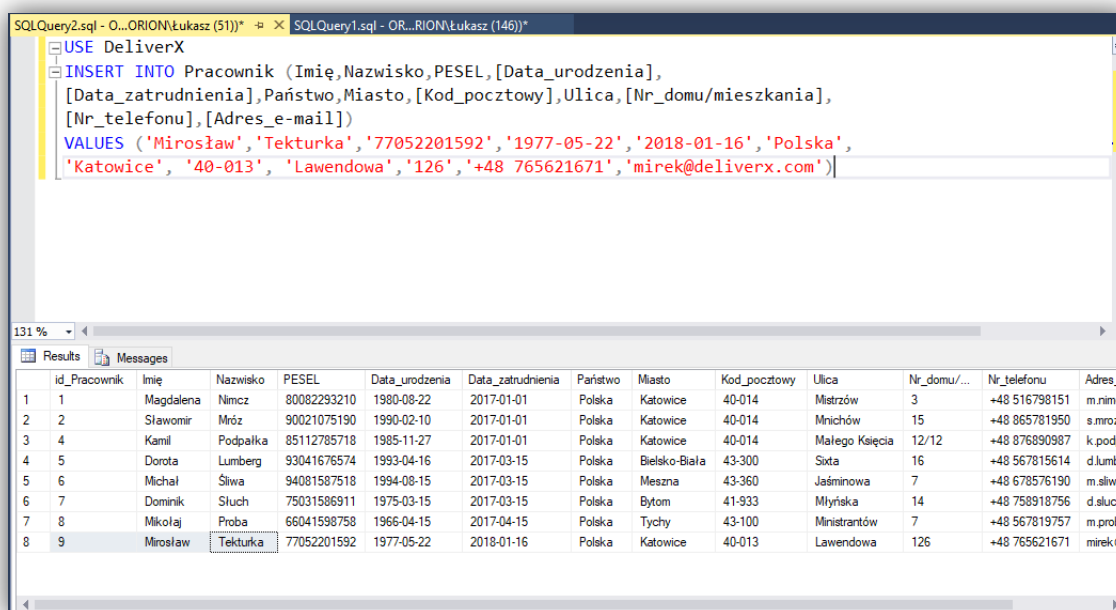
Pomiędzy tabelami „**Firma_Ubezpieczeniowa**” i „**Ubezpieczenie**” istnieje wiązanie jeden do wielu. W naszej bazie możemy posiadać wiele ubezpieczeń od tej samej firmy ubezpieczeniowej, co w praktyce bardzo często ma miejsce z uwagi na korzystniejsze ceny.

Pomiędzy tabelami „**Typ_Ubezpieczenia**” i „**Ubezpieczenie**” istnieje wiązanie jeden do wielu. W tabeli „**Ubezpieczenie**” może znajdować się wiele rekordów odnoszących się do tego samego typu ubezpieczenia.

6) Wybrane zapytania:

Zapytania „INSERT”

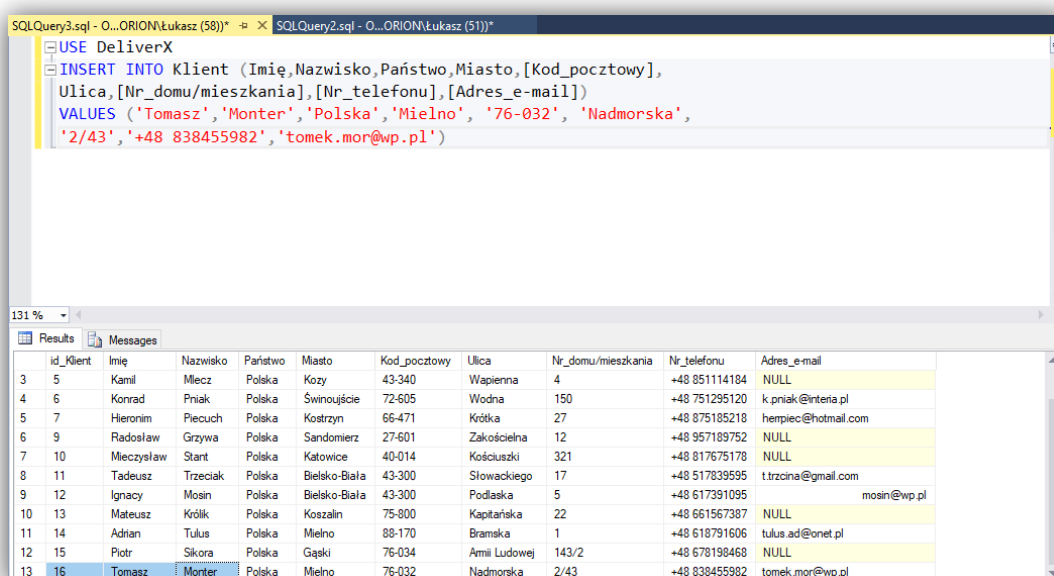
a) Zapytanie dodające nowego pracownika do tabeli „Pracownik”.



```
USE DeliverX
INSERT INTO Pracownik (Imię,Nazwisko,PESEL,[Data_urodzenia],
[Data_zatrudnienia],Państwo,Miasto,[Kod_pocztowy],Ulica,[Nr_domu/mieszkania],
[Nr_telefonu],[Adres_e-mail])
VALUES ('Mirosław','Tekturka','77052201592','1977-05-22','2018-01-16','Polska',
'Katowice','40-013','Lawendowa','126','+48 765621671','mirek@deliverx.com')
```

	id_Pracownik	Imię	Nazwisko	PESEL	Data_urodzenia	Data_zatrudnienia	Państwo	Miasto	Kod_pocztowy	Ulica	Nr_domu/...	Nr_telefonu	Adres_e
1	1	Magdalena	Nimcz	80082293210	1980-08-22	2017-01-01	Polska	Katowice	40-014	Mistrzów	3	+48 516798151	m.nimcz
2	2	Sławomir	Mról	90021075190	1990-02-10	2017-01-01	Polska	Katowice	40-014	Mnichów	15	+48 365781950	s.mroz@
3	4	Kamil	Podpalika	85112785718	1985-11-27	2017-01-01	Polska	Katowice	40-014	Małego Księcia	12/12	+48 876890987	k.podpa
4	5	Dorota	Lumberg	93041676574	1993-04-16	2017-03-15	Polska	Bielsko-Biała	43-300	Szta	16	+48 567815614	d.lumbe
5	6	Michał	Słiwa	94081587518	1994-08-15	2017-03-15	Polska	Meszna	43-360	Jaśminowa	7	+48 678576190	m.sliwa
6	7	Dominik	Słuch	75031586911	1975-03-15	2017-03-15	Polska	Bytom	41-933	Młyńska	14	+48 758918756	d.słuch
7	8	Mikołaj	Proba	66041598758	1966-04-15	2017-04-15	Polska	Tychy	43-100	Ministrantów	7	+48 567819757	m.proba
8	9	Mirosław	Tekturka	77052201592	1977-05-22	2018-01-16	Polska	Katowice	40-013	Lawendowa	126	+48 765621671	mirek@

b) Zapytanie dodające nowego klienta do tabeli „Klient”.



```
USE DeliverX
INSERT INTO Klient (Imię,Nazwisko,Państwo,Miasto,[Kod_pocztowy],
Ulica,[Nr_domu/mieszkania],[Nr_telefonu],[Adres_e-mail])
VALUES ('Tomasz','Monter','Polska','Mielno','76-032','Nadmorska',
'2/43','+48 838455982','tomek.mor@wp.pl')
```

	id_Klient	Imię	Nazwisko	Państwo	Miasto	Kod_pocztowy	Ulica	Nr_domu/mieszkania	Nr_telefonu	Adres_e-mail
3	5	Kamil	Miecz	Polska	Kozy	43-340	Wapienna	4	+48 851114184	NULL
4	6	Konrad	Pniak	Polska	Swinoujście	72-605	Wodna	150	+48 751295120	k.pniak@interia.pl
5	7	Hieronim	Piecuch	Polska	Kostrzyn	66-471	Krótką	27	+48 875185218	herpiec@hotmail.com
6	9	Radosław	Grzywa	Polska	Sandomierz	27-601	Zakościelna	12	+48 957189752	NULL
7	10	Mieczysław	Stant	Polska	Katowice	40-014	Kościuszki	321	+48 817675178	NULL
8	11	Tadeusz	Trzeciak	Polska	Bielsko-Biała	43-300	Słowackiego	17	+48 517839595	t.trzcina@gmail.com
9	12	Ignacy	Mosin	Polska	Bielsko-Biała	43-300	Podlaska	5	+48 617391095	mosin@wp.pl
10	13	Mateusz	Królik	Polska	Koszalin	75-800	Kapitańska	22	+48 661567387	NULL
11	14	Adrian	Tulus	Polska	Mielno	88-170	Bramska	1	+48 618791606	tulus.ad@onet.pl
12	15	Piotr	Sikora	Polska	Gąski	76-034	Amii Ludowej	143/2	+48 678198468	NULL
13	16	Tomasz	Monter	Polska	Mielno	76-032	Nadmorska	2/43	+48 838455982	tomek.mor@wp.pl

c) Zapytanie dodające nową przesyłkę do tabeli „Przesyłka”.

SQLQuery14.sql - O...ORION\Łukasz (58)) * SQLQuery12.sql - O...RION\Łukasz (144)) *

```

USE DeliverX
INSERT INTO Przesyłka(id_Typ_Przesyłki,id_Doręczenia,id_Nadawca,id_Odbiorca,
id_Kurs,[Data_nadania],[Data_wymaganego_dostarczenia],Waga,Jednostka_wagi,
Opłata,Waluta,[Nr_dokumentu_przewozowego],id_Typ_Płatności)
VALUES ('1','16','5','16','11','2018-01-15 14:00','2018-01-24 00:00',
'5','KG','260','PLN','7458250','3')

```

131 %

Results Messages

	id_Przesyłka	id_Typ_Przesyłki	id_Doręczenia	id_Nadawca	id_Odbiorca	id_Kurs	Data_nadania	Data_wymaganego_dostarczenia	Data_rzeczywistego_dostarczenia	Waga	Jednos
1	6	1	5	6	5	2	2017-12-28 12:00:00	2018-01-05 00:00:00	2018-01-05 13:00:00	2	KG
2	8	3	5	9	5	2	2018-01-04 13:30:00	2018-01-10 00:00:00	2018-01-05 19:00:00	370	KG
3	10	2	10	13	10	4	2018-01-02 10:00:00	2018-01-06 00:00:00	2018-01-07 14:33:00	15	KG
4	12	4	9	12	9	4	2017-12-29 15:00:00	2018-01-06 00:00:00	2018-01-07 17:37:00	26	KG
5	13	3	6	3	6	9	2018-01-02 09:00:00	2018-01-15 00:00:00	2018-01-10 09:00:00	2000	KG
6	15	2	3	5	3	5	2018-01-07 18:00:00	2018-01-09 00:00:00	2018-01-08 08:00:00	35	KG
7	16	3	7	10	7	8	2018-01-07 13:00:00	2018-01-11 00:00:00	2018-01-09 14:00:00	800	KG
8	17	3	1	11	1	8	2018-01-07 16:00:00	2018-01-13 00:00:00	2018-01-09 20:00:00	600	KG
9	20	1	16	5	16	11	2018-01-15 14:00:00	2018-01-24 00:00:00	NULL	5	KG

d) Zapytanie dodające nowy kurs do tabeli „Kurs”.

SQLQuery12.sql - O...RION\Łukasz (144)) * SQLQuery6.sql - O...ORION\Łukasz (58)) *

```

USE DeliverX
INSERT INTO Kurs (id_Pracownik,id_Pojazd,Data,Liczba_kilometrów)
VALUES ('1','2','2018-01-20','500')

```

131 %

Results Messages

	id_Kurs	id_Pracownik	id_Pojazd	Data	Liczba_kilometrów	Uwagi
1	2	4	2	2018-01-05 13:00:00	450	NULL
2	4	2	3	2018-01-07 12:00:00	350	NULL
3	5	7	5	2018-01-08 07:00:00	100	NULL
4	8	8	6	2018-01-09 06:00:00	350	NULL
5	9	8	10	2018-01-10 17:00:00	700	NULL
6	11	1	2	2018-01-20 00:00:00	500	NULL

e) Zapytanie dodające nowy pojazd do tabeli „Pojazd”.

SQLQuery2.sql - ORION\lukasz (144)* ORION.DeliverX - dbo.Pojazd SQLQuery1.sql - O...ORION\lukasz (57)*

```
USE DeliverX
INSERT INTO Pojazd (Marka,Model,[Nr_rejestracyjny],id_Typ_Pojazdu,[Max_ładowność],
[Jednostka_ładowności],Rocznik,[Stan_licznika],
[Stan_licznika_na_dzień])
VALUES ('Mercedes','Vito','SK1366','1','1150', 'KG', '2018-01-01',
'0','2018-01-15')
```

131 %

Results Messages

	id_Pojazd	Marka	Model	Nr_rejestracyjny	id_Typ_Pojazdu	Max_ładowność	Jednostka_ładowności	Rocznik	Stan_licznika	Jednostka_licznika	Stan_licznika_na_dzień
2	3	Volkswagen	Caddy	SK1325B	1	600	KG	2017-01-01	177	tys. KM	2018-01-01
3	5	Volkswagen	Caddy	SK1326B	1	600	KG	2017-01-01	120	tys. KM	2018-01-01
4	6	Ford	Transit	SK1578A	1	1630	KG	2016-06-06	240	tys. KM	2018-01-01
5	7	Ford	Transit	SK1578A	1	1630	KG	2016-06-09	260	tys. KM	2018-01-01
6	8	Ford	Transit	SK1579A	1	1630	KG	2016-06-10	240	tys. KM	2018-01-01
7	9	Iveco	Eurocargo	SK31256	2	3300	KG	2015-01-01	450	tys. KM	2018-01-01
8	10	Iveco	Eurocargo	SK71485	2	3500	KG	2015-01-01	400	tys. KM	2018-01-01
9	11	Iveco	Eurocargo	SK78491	2	3900	KG	2015-05-15	376	tys. KM	2018-01-01
10	12	Mercedes	Vito	SK1366	1	1150	KG	2018-01-01	0	tys. KM	2018-01-15

f) Zapytanie dodające nową zmianę statusu paczki do tabeli „Historia”.

SQLQuery15.sql - ...ORION\lukasz (57)* SQLQuery14.sql - ...ORION\lukasz (58)*

```
USE DeliverX
INSERT INTO Historia(id_Przesyłka,id_Status,[Data_zmiany_statusu])
VALUES ('20','1','2018-01-15 14:10')
```

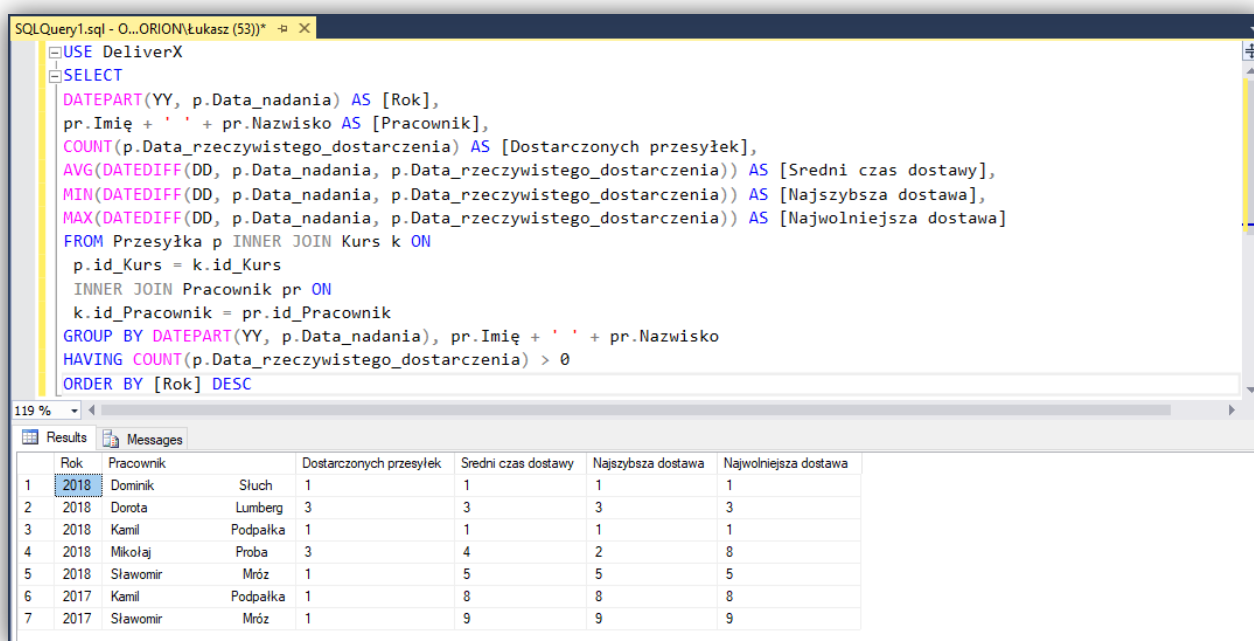
131 %

Results Messages

	id_Historia	id_Przesyłka	id_Status	Data_zmiany_statusu	Opis
16	16	17	2	2018-01-08 13:00:00	NULL
17	17	6	3	2018-01-05 13:00:00	NULL
18	18	12	4	2018-01-07 17:37:00	zły ...
19	19	13	3	2018-01-10 09:00:00	NULL
20	20	10	3	2018-01-07 14:33:00	NULL
21	21	8	4	2018-01-05 19:00:00	nie ...
22	22	15	3	2018-01-08 08:00:00	NULL
23	23	16	3	2018-01-09 14:00:00	usz...
24	24	17	5	2018-01-09 20:00:00	prze...
25	26	20	1	2018-01-15 14:10:00	NULL

Zapytania „SELECT”

- a) Zapytanie przedstawia zestawienie liczby przesyłek dostarczonych w danym roku przez konkretnego pracownika. Ponadto zapytanie dostarcza pracodawcy informacji na temat czasów dostaw odnotowywanych przez poszczególnych pracowników.



The screenshot shows a SQL query window with the following code:

```
USE DeliverX
SELECT
    DATEPART(YY, p.Data_nadania) AS [Rok],
    pr.Imię + ' ' + pr.Nazwisko AS [Pracownik],
    COUNT(p.Data_rzeczywistego_dostarczenia) AS [Dostarczonych przesyłek],
    AVG(DATEDIFF(DD, p.Data_nadania, p.Data_rzeczywistego_dostarczenia)) AS [Sredni czas dostawy],
    MIN(DATEDIFF(DD, p.Data_nadania, p.Data_rzeczywistego_dostarczenia)) AS [Najszybsza dostawa],
    MAX(DATEDIFF(DD, p.Data_nadania, p.Data_rzeczywistego_dostarczenia)) AS [Najwolniejsza dostawa]
FROM Przesyłka p INNER JOIN Kurs k ON
    p.id_Kurs = k.id_Kurs
    INNER JOIN Pracownik pr ON
    k.id_Pracownik = pr.id_Pracownik
GROUP BY DATEPART(YY, p.Data_nadania), pr.Imię + ' ' + pr.Nazwisko
HAVING COUNT(p.Data_rzeczywistego_dostarczenia) > 0
ORDER BY [Rok] DESC
```

The results are displayed in a table with the following data:

	Rok	Pracownik	Dostarczonych przesyłek	Sredni czas dostawy	Najszybsza dostawa	Najwolniejsza dostawa
1	2018	Dominik Słuch	1	1	1	1
2	2018	Dorota Lumberg	3	3	3	3
3	2018	Kamil Podpałka	1	1	1	1
4	2018	Mikołaj Proba	3	4	2	8
5	2018	Sławomir Mróz	1	5	5	5
6	2017	Kamil Podpałka	1	8	8	8
7	2017	Sławomir Mróz	1	9	9	9

- b) Zapytanie zwracające klasyfikację ilości klientów w stosunku do miejsca ich zamieszkania.

```

USE DeliverX
SELECT
    k.Miasto,
    COUNT(k.id_Klient) AS [Liczba klientów],
    CASE
        WHEN COUNT(k.id_Klient) < 2
        THEN 'do poprawki'
        WHEN COUNT(k.id_Klient) BETWEEN 2 AND 3
        THEN 'mogłoby być lepiej'
        WHEN COUNT(k.id_Klient) BETWEEN 4 AND 10
        THEN 'w porządku'
        WHEN COUNT(k.id_Klient) > 10
        THEN 'fenomenalnie'
    END AS [Klasyfikacja]
FROM Klient k
GROUP BY k.Miasto
ORDER BY Klasyfikacja DESC

```

	Miasto	Liczba klientów	Klasyfikacja
1	Sandomierz	4	w porządku
2	Mielno	2	mogłoby być lepiej
3	Bielsko-Biała	2	mogłoby być lepiej
4	Cieszyn	1	do poprawki
5	Gąski	1	do poprawki
6	Katowice	1	do poprawki

- c) Zapytanie prezentuje klasyfikację liczby paczek odebranych przez poszczególnych klientów w konkretnym roku kalendarzowym.

```

USE DeliverX
SELECT
    k.Imię + ' ' + k.Nazwisko AS [Klient],
    YEAR(p.Data_rzeczywistego_dostarczenia) AS [Rok],
    COUNT(p.Data_rzeczywistego_dostarczenia) [Ilość odebranych paczek]
FROM Klient k INNER JOIN Przesyłka p
ON k.id_Klient = p.id_Odbiorca
GROUP BY k.Imię + ' ' + k.Nazwisko, YEAR(p.Data_rzeczywistego_dostarczenia)
HAVING YEAR(p.Data_rzeczywistego_dostarczenia) > 0
ORDER BY YEAR(p.Data_rzeczywistego_dostarczenia)

```

	Klient	Rok	Ilość odebranych paczek
1	Adrian Tulus	2018	1
2	Borys Maciej	2018	1
3	Hieronim Piecuch	2018	1
4	Jarosław Kowala	2018	1
5	Kamil Miecz	2018	2
6	Konrad Pniak	2018	1
7	Mieczysław Stant	2018	1
8	Piotr Sikora	2018	1
9	Radosław Grzywa	2018	1
10	Tomasz Monter	2018	1

- d) Zapytanie ukazuje strukturę liczebności wysłanych przez klientów paczek na przestrzeni lat.

SQLQuery1.sql - O...ORION\Łukasz (53))

```

USE DeliverX
SELECT
    k.Imię + ' ' + k.Nazwisko AS [Klient],
    YEAR(p.Data_rzeczywistego_dostarczenia) AS [Rok],
    COUNT(p.Data_rzeczywistego_dostarczenia) [Ilość wysłanych paczek]
FROM Klient k INNER JOIN Przesyłka p
ON k.id_Klient = p.id_Nadawca
GROUP BY k.Imię + ' ' + k.Nazwisko, YEAR(p.Data_rzeczywistego_dostarczenia)
HAVING YEAR(p.Data_rzeczywistego_dostarczenia) > 0
ORDER BY YEAR(p.Data_rzeczywistego_dostarczenia)

```

119 %

	Klient	Rok	Ilość wysłanych paczek
1	Borys Maciej	2018	1
2	Ignacy Mosin	2018	1
3	Kamil Miecz	2018	1
4	Konrad Priak	2018	1
5	Mateusz Królik	2018	1
6	Mieczysław Stant	2018	1
7	Radosław Grzywa	2018	1
8	Tadeusz Trzeciak	2018	4

- e) Zapytanie przedstawia średnie oraz skrajne ceny ubezpieczeń floty pojazdów jaką dysponuje firma. Instrukcja grupuje informacje względem typu ubezpieczenia („**Typ_Ubezpieczenia.Typ**”) oraz typu pojazdu („**Typ_Pojazdu.Typ**”). Są to niezmiennie ważne informacje w przypadku negocjowania nowych umów ubezpieczeniowych.

SQLQuery4.sql - O...ORION\Łukasz (58))

```

SELECT
    tu.Type AS [Typ ubezpieczenia],
    tp.Type AS [Typ pojazdu],
    COUNT(u.id_Ubezpieczenie) AS [Ilość ubezpieczeń],
    AVG(u.Cena) AS [Średnia cena],
    MAX(u.Cena) AS [Najdroższa cena],
    MIN(u.Cena) AS [Najtańsza cena]
FROM Ubezpieczenie u INNER JOIN Typ_Ubezpieczenia tu ON
u.id_Typ_Ubezpieczenia = tu.id_Typ_Ubezpieczenia INNER JOIN Pojazd p ON
u.id_Pojazd = p.id_Pojazd INNER JOIN Typ_Pojazdu tp ON
p.id_Typ_Pojazdu = tp.id_Typ_Pojazdu
GROUP BY tp.Type, tu.Type

```

131 %

	Typ ubezpieczenia	Typ pojazdu	Ilość ubezpieczeń	Średnia cena	Najdroższa cena	Najtańsza cena
1	AC	do 3.5 tony	1	6000,00	6000,00	6000,00
2	AC	powyżej 3.5 tony	1	5500,00	5500,00	5500,00
3	assistance	do 3.5 tony	1	3300,00	3300,00	3300,00
4	NNW	do 3.5 tony	1	4400,00	4400,00	4400,00
5	OC	do 3.5 tony	6	1908,3333	2500,00	1300,00
6	OC	powyżej 3.5 tony	3	3250,00	3700,00	2750,00
7	ub. ładunku	do 3.5 tony	2	5000,00	7700,00	2300,00

- f) Zapytanie klasyfikuje liczbę nadanych statusów przesyłek w przestrzeni czasu.

SQLQuery1.sql - O...ORION\Łukasz (53)*

```

USE DeliverX
SELECT
    st.Nazwa AS [Status],
    DATEPART(YYYY,(h.Data_zmiany_statusu)) AS [Rok],
    DATEPART(MM,(h.Data_zmiany_statusu)) AS [Miesiąc],
    COUNT(h.id_Przesyłka) AS [Liczba przesyłek, które otrzymały status]
FROM Historia h INNER JOIN Status st ON
    h.id_Status = st.id_Status
GROUP BY st.Nazwa,DATEPART(YYYY,(h.Data_zmiany_statusu)), DATEPART(MM,(h.Data_zmiany_statusu))
ORDER BY DATEPART(YYYY,(h.Data_zmiany_statusu)) DESC

```

119 %

Results Messages

	Status	Rok	Miesiąc	Liczba przesyłek, które otrzymały status
1	nadana	2018	1	10
2	odebrana	2018	1	8
3	uszkodzenie	2018	1	1
4	w trakcie realizacji	2018	1	9
5	zwrot	2018	1	2
6	nadana	2017	12	2
7	w trakcie realizacji	2017	1	1
8	w trakcie realizacji	2017	12	1

- g) Zapytanie przedstawia liczbę pojazdów danego typu oraz ich średni: przebieg, wiek oraz ładowność.

SQLQuery1.sql - O...ORION\Łukasz (58)*

```

USE DeliverX
SELECT
    tp.Type,
    COUNT(p.id_Pojazd) AS [Liczba pojazdów],
    AVG(p.Stan_licznika) AS [Średni przebieg],
    AVG(DATEDIFF(YYYY,p.rocznik,GETDATE())) AS [Średni wiek pojazdu (lat)],
    AVG(Max_ładowność) AS [Średnia ładowność]
FROM Pojazd p INNER JOIN Typ_Pojazdu tp ON
    p.id_Typ_Pojazdu = tp.id_Typ_Pojazdu
GROUP BY tp.Type

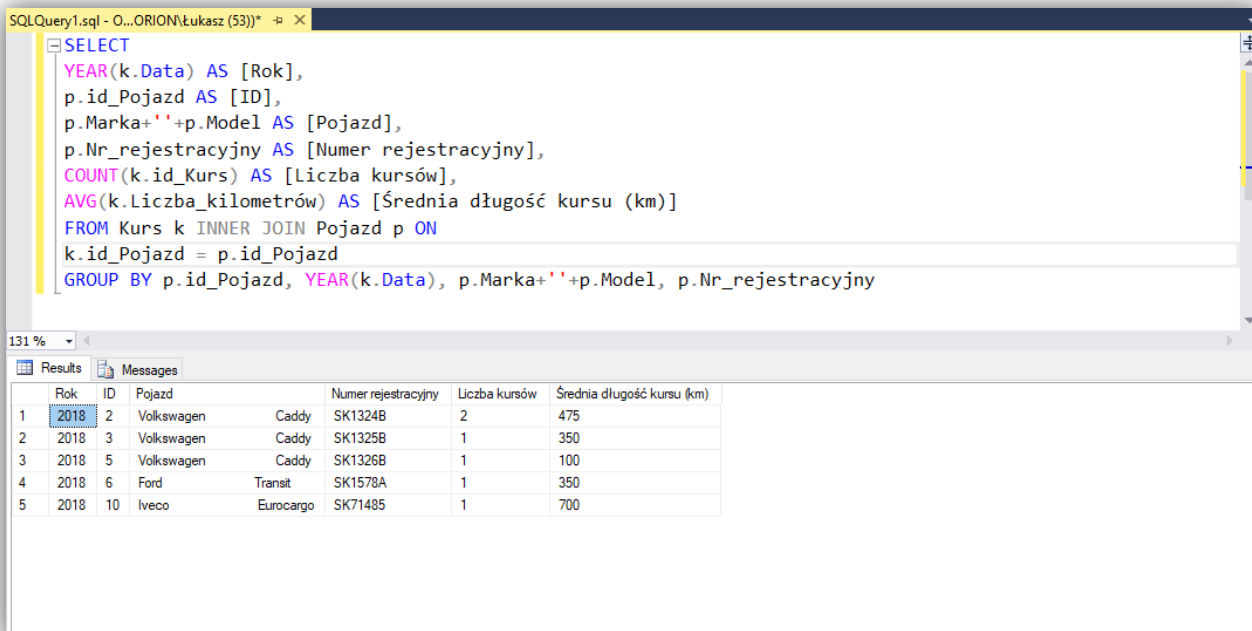
```

131 %

Results Messages

	Typ	Liczba pojazdów	Średni przebieg	Średni wiek pojazdu (lat)	Średnia ładowność
1	do 3.5 tony	7	169.571428571429	1	1120
2	powyżej 3.5 tony	3	408.666666666667	3	3566.66666666667

- h) Zapytanie prezentuje ilość wykonanych kursów przy użyciu konkretnych pojazdów z uwzględnieniem roku dokonywania transportu.



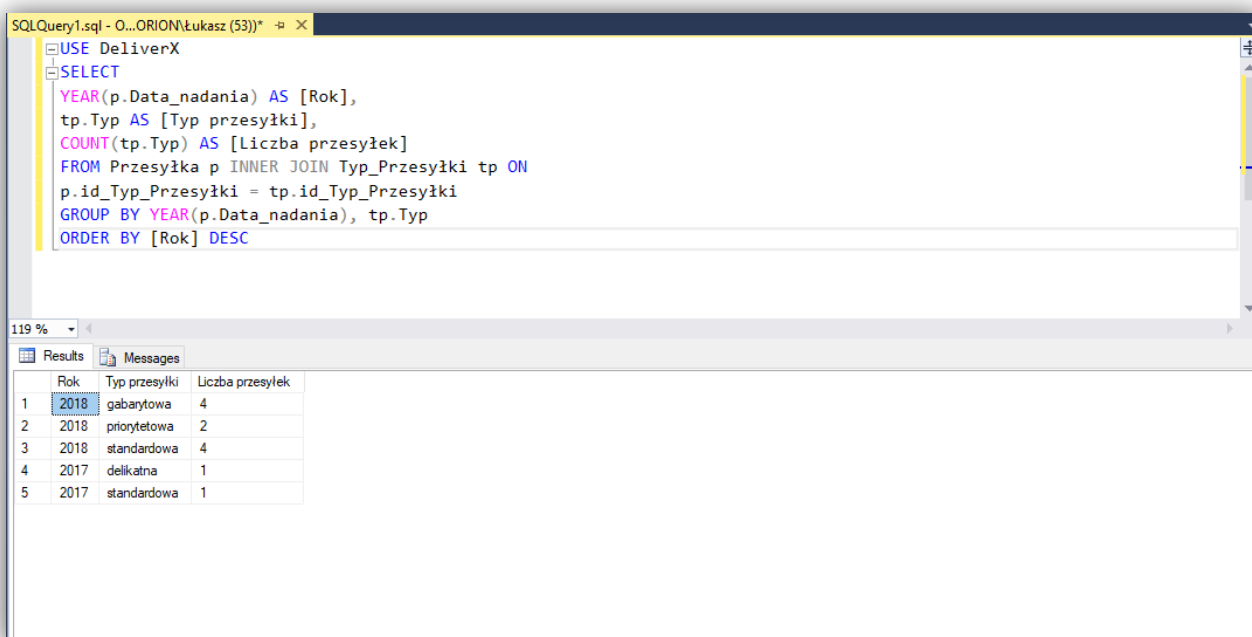
The screenshot shows a SQL query window with the following query:

```
SELECT
    YEAR(k.Data) AS [Rok],
    p.id_Pojazd AS [ID],
    p.Marka+' '+p.Model AS [Pojazd],
    p.Nr_rejestracyjny AS [Numer rejestracyjny],
    COUNT(k.id_Kurs) AS [Liczba kursów],
    AVG(k.Liczba_kilometrów) AS [Średnia długość kursu (km)]
FROM Kurs k INNER JOIN Pojazd p ON
    k.id_Pojazd = p.id_Pojazd
GROUP BY p.id_Pojazd, YEAR(k.Data), p.Marka+' '+p.Model, p.Nr_rejestracyjny
```

The results are displayed in a table with the following data:

Rok	ID	Pojazd	Numer rejestracyjny	Liczba kursów	Średnia długość kursu (km)
2018	2	Volkswagen Caddy	SK1324B	2	475
2018	3	Volkswagen Caddy	SK1325B	1	350
2018	5	Volkswagen Caddy	SK1326B	1	100
2018	6	Ford Transit	SK1578A	1	350
2018	10	Iveco Eurocargo	SK71485	1	700

- i) Zapytanie zwraca informację dotyczącą liczby paczek konkretnego typu nadanych w danym roku.



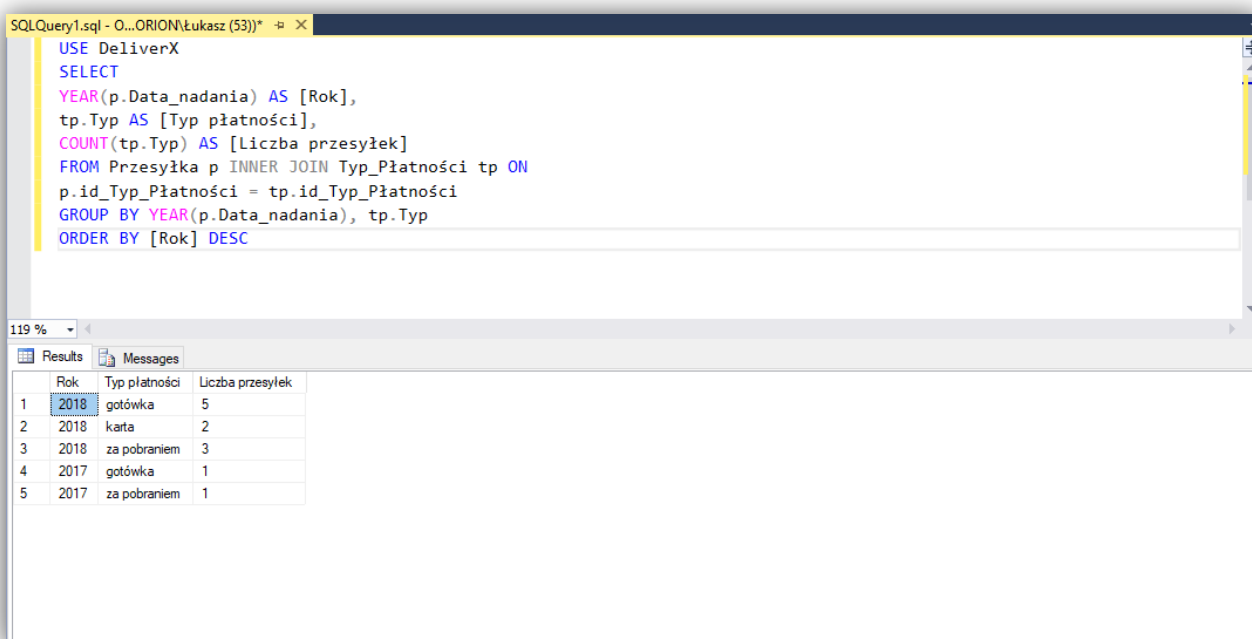
The screenshot shows a SQL query window with the following query:

```
USE DeliverX
SELECT
    YEAR(p.Data_nadania) AS [Rok],
    tp.Type AS [Typ przesyłki],
    COUNT(tp.Type) AS [Liczba przesyłek]
FROM Przesyłka p INNER JOIN Typ_Przesyłki tp ON
    p.id_Typ_Przesyłki = tp.id_Typ_Przesyłki
GROUP BY YEAR(p.Data_nadania), tp.Type
ORDER BY [Rok] DESC
```

The results are displayed in a table with the following data:

Rok	Typ przesyłki	Liczba przesyłek
2018	gabarytowa	4
2018	priorytetowa	2
2018	standardowa	4
2017	delikatna	1
2017	standardowa	1

- j) Zapytanie zwraca informację dotyczącą liczby paczek, za które zapłacono przy pomocy konkretnej metody płatności w danym roku.



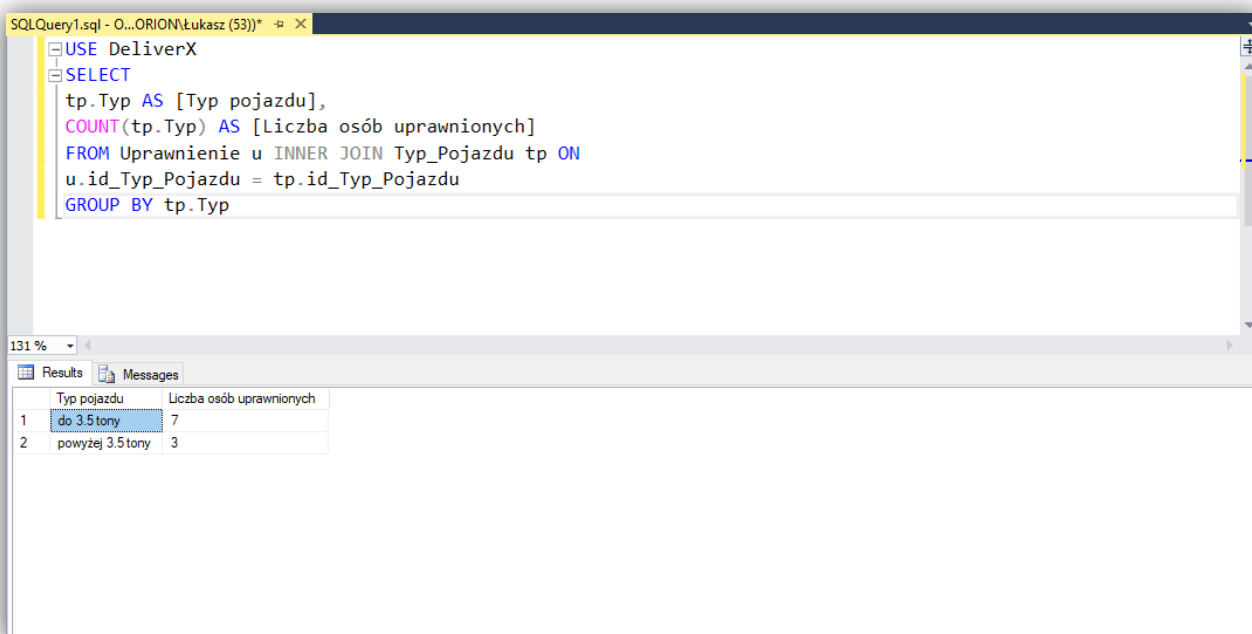
The screenshot shows a SQL query window with the following text:

```
USE DeliverX
SELECT
YEAR(p.Data_nadania) AS [Rok],
tp.Type AS [Typ płatności],
COUNT(tp.Type) AS [Liczba przesyłek]
FROM Przesyłka p INNER JOIN Typ_Płatności tp ON
p.id_Typ_Płatności = tp.id_Typ_Płatności
GROUP BY YEAR(p.Data_nadania), tp.Type
ORDER BY [Rok] DESC
```

Below the query, the 'Results' tab displays the following data:

	Rok	Typ płatności	Liczba przesyłek
1	2018	gotówka	5
2	2018	karta	2
3	2018	za pobraniem	3
4	2017	gotówka	1
5	2017	za pobraniem	1

- k) Zapytanie przedstawia liczbę pracowników posiadających uprawnienie na dany typ pojazdu.



The screenshot shows a SQL query window with the following text:

```
USE DeliverX
SELECT
tp.Type AS [Typ pojazdu],
COUNT(tp.Type) AS [Liczba osób uprawnionych]
FROM Uprawnienie u INNER JOIN Typ_Pojazdu tp ON
u.id_Typ_Pojazdu = tp.id_Typ_Pojazdu
GROUP BY tp.Type
```

Below the query, the 'Results' tab displays the following data:

	Typ pojazdu	Liczba osób uprawnionych
1	do 3.5 tony	7
2	powyżej 3.5 tony	3

Widoki

- a) Widok tabeli „**Historia**” z zastąpionym polem „**id_Status**” przez nazwę statusu z tabeli „**Status**”:

SQLQuery18.sql -...ORION\Łukasz (60))

CREATE VIEW Historia_widok

AS

SELECT

Historia.id_Historia, Historia.id_Przesyłka, Status.Nazwa,

Historia.Data_zmiany_statusu, Historia.Opis

FROM Historia INNER JOIN Status

ON Historia.id_Status = Status.id_Status

WITH CHECK OPTION

131 %

Results

Messages

	id_Historia	id_Przesyłka	Nazwa	Data_zmiany_statusu	Opis
16	16	17	w trakcie realizacji	2018-01-08 13:00:00	NULL
17	17	6	odebrana	2018-01-05 13:00:00	NULL
18	18	12	zwrot	2018-01-07 17:37:00	zły adres
19	19	13	odebrana	2018-01-10 09:00:00	NULL
20	20	10	odebrana	2018-01-07 14:33:00	NULL
21	21	8	zwrot	2018-01-05 19:00:00	nie przyjęcie paczki
22	22	15	odebrana	2018-01-08 08:00:00	NULL
23	23	16	odebrana	2018-01-09 14:00:00	uszkodzone opak...
24	24	17	uszkodzenie	2018-01-09 20:00:00	przesyłka uszkod...
25	26	20	nadana	2018-01-15 14:10:00	NULL

- b) Widok tabeli „**Pojazd**”. Pole „**Typ_pojazdu**” zostało pobrane z tabeli „**Typ_Pojazdu**”.

SQLQuery5.sql - O...ORION\tukasz (59))

CREATE VIEW Pojazd_widok

AS

SELECT

Pojazd.id_Pojazd, Pojazd.Marka, Pojazd.Model, Pojazd.Nr_rejestracyjny,

Typ_Pojazdu.Typ AS Typ_pojazdu, Pojazd.Max_ładowność, Pojazd.Jednostka_ładowności, Pojazd.Rocznik,

Pojazd.Stan_licznika, Pojazd.Jednostka_licznika, Pojazd.Stan_licznika_na_dzień,

dbo.Pojazd.Rodzaj_paliwa

FROM Pojazd INNER JOIN Typ_Pojazdu ON

Pojazd.id_Typ_Pojazdu = Typ_Pojazdu.id_Typ_Pojazdu

WITH CHECK OPTION

131 %

Results

Messages

	id_Pojazd	Marka	Model	Nr_rejestracyjny	Typ_pojazdu	Max_ładowność	Jednostka_ładowności	Rocznik	Stan_licznika	Jednostka_licznika	Stan_licznika_na_dzień
1	2	Volkswagen	Caddy	SK1324B	do 3.5 tony	600	KG	2017-01-01	150	tys. KM	2018-01-01
2	3	Volkswagen	Caddy	SK1325B	do 3.5 tony	600	KG	2017-01-01	177	tys. KM	2018-01-01
3	5	Volkswagen	Caddy	SK1326B	do 3.5 tony	600	KG	2017-01-01	120	tys. KM	2018-01-01
4	6	Ford	Transit	SK1578A	do 3.5 tony	1630	KG	2016-06-06	240	tys. KM	2018-01-01
5	7	Ford	Transit	SK1578A	do 3.5 tony	1630	KG	2016-06-09	260	tys. KM	2018-01-01
6	8	Ford	Transit	SK1579A	do 3.5 tony	1630	KG	2016-06-10	240	tys. KM	2018-01-01
7	9	Iveco	Eurocargo	SK31256	powyżej 3.5 tony	3300	KG	2015-01-01	450	tys. KM	2018-01-01
8	10	Iveco	Eurocargo	SK71485	powyżej 3.5 tony	3500	KG	2015-01-01	400	tys. KM	2018-01-01
9	11	Iveco	Eurocargo	SK78491	powyżej 3.5 tony	3900	KG	2015-05-15	376	tys. KM	2018-01-01
10	12	Mercedes	Vito	SK1366	do 3.5 tony	1150	KG	2018-01-01	0	tys. KM	2018-01-15

- c) Widok tabeli „**Ubezpieczenie**”. Do tabeli zostały dołączone informacje odnośnie Marki, Modelu oraz numeru rejestracyjnego danego pojazdu z tabeli „**Pojazd**”. Z relacji „**Typ_Ubezpieczenia**” zostało wprowadzone pole - typu świadczenia, natomiast z relacji „**Firma_Ubezpieczeniowa**” zapożyczono pole z nazwą firmy udzielającej ubezpieczenia.

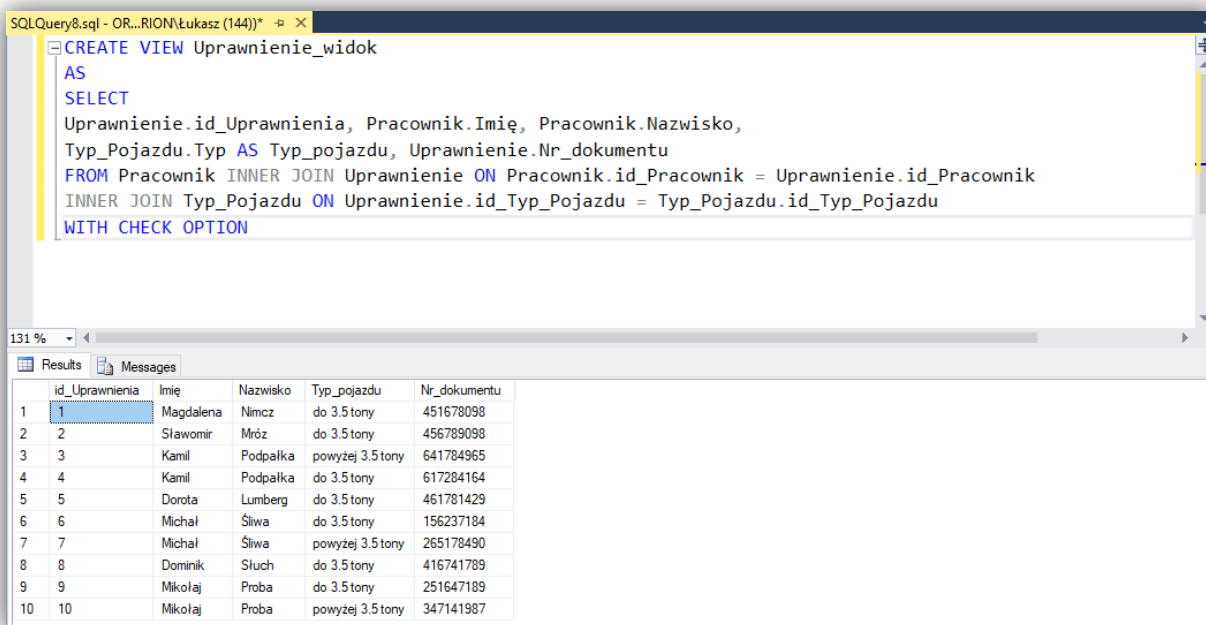
```

SQLQuery3.sql - O...ORION\Łukasz (59)) *
CREATE VIEW Ubezpieczenie_widok
AS
SELECT
    Ubezpieczenie.id_Ubezpieczenie, Pojazd.Marka, Pojazd.Model, Typ_Pojazdu.Type AS [Typ_pojazdu],
    Pojazd.Nr_rejestracyjny, Typ_Ubezpieczenia.Type AS [Typ_Ubezpieczenia], Firma_Ubezpieczeniowa.Nazwa AS [Firma_Ubezpieczeniowa],
    Ubezpieczenie.Nr_polisy, Ubezpieczenie.Obowiazuje_od, Ubezpieczenie.Obowiazuje_do, Ubezpieczenie.Cena,
    Ubezpieczenie.Waluta
FROM Ubezpieczenie
INNER JOIN Typ_Ubezpieczenia ON
    Ubezpieczenie.id_Typ_Ubezpieczenia = Typ_Ubezpieczenia.id_Typ_Ubezpieczenia
INNER JOIN Firma_Ubezpieczeniowa ON
    Ubezpieczenie.id_Firma_Ubezpieczeniowa = Firma_Ubezpieczeniowa.id_Firma_Ubezpieczeniowa
INNER JOIN Pojazd ON
    Ubezpieczenie.id_Pojazd = Pojazd.id_Pojazd
INNER JOIN Typ_Pojazdu ON
    Pojazd.id_Typ_Pojazdu = Typ_Pojazdu.id_Typ_Pojazdu
WITH CHECK OPTION

```

	id_Ubezpieczenie	Marka	Model	Typ_pojazdu	Nr_rejestracyjny	Typ_Ubezpieczenia	Firma	Nr_polisy	Obowiazuje_od	Obowiazuje_do	Cena	Waluta
1	2	Volkswagen	Caddy	do 3.5 tony	SK1324B	OC	PZU	56174890	2018-01-13	2019-01-13	1800.00	PLN
2	4	Volkswagen	Caddy	do 3.5 tony	SK1325B	OC	AXA	56718192	2018-01-13	2019-01-13	1650.00	PLN
3	5	Volkswagen	Caddy	do 3.5 tony	SK1326B	OC	PZU	51764829	2018-01-13	2019-01-13	1700.00	PLN
4	6	Ford	Transit	do 3.5 tony	SK1578A	OC	LINK4	17428194	2018-01-13	2019-01-13	1300.00	PLN
5	7	Ford	Transit	do 3.5 tony	SK1578A	OC	PZU	14714818	2018-01-13	2019-01-13	2500.00	PLN
6	8	Ford	Transit	do 3.5 tony	SK1579A	OC	PZU	78141981	2018-01-13	2019-01-13	2500.00	PLN
7	9	Iveco	Eurocargo	powyżej 3.5 tony	SK31256	OC	AXA	45162781	2018-01-13	2019-01-13	3300.00	PLN
8	11	Iveco	Eurocargo	powyżej 3.5 tony	SK71485	OC	AXA	71464111	2017-11-22	2018-11-22	3700.00	PLN
9	12	Iveco	Eurocargo	powyżej 3.5 tony	SK78491	OC	LINK4	56174281	2017-10-10	2018-10-10	2750.00	PLN
10	14	Ford	Transit	do 3.5 tony	SK1578A	AC	AXA	34762748	2018-01-13	2019-01-13	6000.00	PLN
11	15	Iveco	Eurocargo	powyżej 3.5 tony	SK31256	AC	LINK4	67148216	2018-01-13	2019-01-13	5500.00	PLN
12	17	Ford	Transit	do 3.5 tony	SK1579A	assistance	PZU	64718628	2018-01-13	2019-01-13	3300.00	PLN
13	18	Ford	Transit	do 3.5 tony	SK1578A	NNW	AXA	67741818	2018-01-13	2019-01-13	4400.00	PLN
14	19	Volkswagen	Caddy	do 3.5 tony	SK1326B	ub. ładunku	AXA	64217124	2018-01-13	2019-01-13	2300.00	PLN
15	21	Ford	Transit	do 3.5 tony	SK1578A	ub. ładunku	LINK4	16253678	2018-01-13	2019-01-13	7700.00	PLN

- d) Widok tabeli „**Uprawnienie**”. Pole „*id_Pracownik*” zostało zastąpione przez pola – imię oraz nazwisko pochodzące z tabeli „**Pracownik**”. Pole „*Typ_pojazdu*” zostało zastąpione nazwą typu z tablicy słownikowej „**Typ_Pojazdu**”.



```
CREATE VIEW Uprawnienie_widok
AS
SELECT
Uprawnienie.id_Uprawnienia, Pracownik.Imię, Pracownik.Nazwisko,
Typ_Pojazdu.Typ AS Typ_pojazdu, Uprawnienie.Nr_dokumentu
FROM Pracownik INNER JOIN Uprawnienie ON Pracownik.id_Pracownik = Uprawnienie.id_Pracownik
INNER JOIN Typ_Pojazdu ON Uprawnienie.id_Typ_Pojazdu = Typ_Pojazdu.id_Typ_Pojazdu
WITH CHECK OPTION
```

id_Uprawnienia	Imię	Nazwisko	Typ_pojazdu	Nr_dokumentu
1	Magdalena	Nimcz	do 3.5 tony	451678098
2	Sławomir	Mról	do 3.5 tony	456789098
3	Kamil	Podpalka	powyżej 3.5 tony	641784965
4	Kamil	Podpalka	do 3.5 tony	617284164
5	Dorota	Lumberg	do 3.5 tony	461781429
6	Michał	Śliwa	do 3.5 tony	156237184
7	Michał	Śliwa	powyżej 3.5 tony	265178490
8	Dominik	Śluch	do 3.5 tony	416741789
9	Mikołaj	Proba	do 3.5 tony	251647189
10	Mikołaj	Proba	powyżej 3.5 tony	347141987

- e) Widok tabeli „**Przesyłka**”. Zostały zamienione pola identyfikujące odbiorcę („*id_Odbiorca*”), nadawcę („*id_Nadawca*”) przesyłki na odpowiadające im wartości z pól „Imię” oraz „Nazwisko” z tabeli „**Klient**”. Ponadto, przy pomocy klucza obcego „*id_Doręczenia*” w widoku znajduje się dokładny adres miejsca dostarczenia paczki (miasto, ulica, numer domu/mieszkania). Pole „*Typ_przesyłki*” zostało zastąpione odpowiadającymi wartościami z pola „Typ” z relacji „**Typ_Przesyłki**”. Analogiczny proces został zastosowany do pola „*Typ_płatności*”, które teraz odpowiada wartościom pola „Typ” z relacji „**Typ_Płatności**”. Dodatkowymi informacjami zawartymi w widoku są dane personalne pracownika odpowiedzialnego za kurs, w którym bierze lub będzie brała udział przesyłka („*Imię_Pracownika*” oraz „*Nazwisko_Pracownika*”). Dodanie tych informacji do widoku było możliwe dzięki skojarzeniu klucza obcego „*id_Kurs*” z tabelą „**Kurs**”, gdzie natomiast znajdował się klucz obcy „*id_Pracownik*” powiązany z tabelą „**Pracownik**”, w której to zawarte są dane dotyczące pracowników.

```

SQLQuery10.sql - O...RION\lukasz (147)) *
CREATE VIEW Przesyłka_widok
AS
SELECT
Przesyłka.id_Przesyłka, Typ_Przesyłki.Type AS Typ_Przesyłki, Odbiorca.Imię AS Odbiorca_Imię,
Odbiorca.Nazwisko AS Odbiorca_Nazwisko, Nadawca.Imię AS Nadawca_Imię,
Nadawca.Nazwisko AS Nadawca_Nazwisko, Doręczenie.Miasto AS Do_miasta, Doręczenie.Ulica AS Na_ulicę,
Doręczenie.[Nr_domu/mieszkania] AS Do_mieszkania,
Pracownik.Imię AS Pracownik_Imię, Pracownik.Nazwisko AS Pracownik_Nazwisko, Przesyłka.Data_nadania,
Przesyłka.Data_wymaganego_dostarczenia, Przesyłka.Data_rzeczywistego_dostarczenia,
Przesyłka.Waga, Przesyłka.Jednostka_wagi, Przesyłka.Opłata, Przesyłka.Waluta,
Przesyłka.Nr_dokumentu_przewozowego, Typ_Płatności.Type AS Typ_płatności
FROM Przesyłka INNER JOIN Klient Odbiorca ON
Odbiorca.id_Klient = Przesyłka.id_Doręczenia INNER JOIN Klient Nadawca ON
Nadawca.id_Klient = Przesyłka.id_Nadawca INNER JOIN Klient Doręczenie ON
Doręczenie.id_Klient = Przesyłka.id_Odbiorca INNER JOIN
Typ_Płatności ON
Przesyłka.id_Typ_Płatności = Typ_Płatności.id_Typ_Płatności INNER JOIN Typ_Przesyłki ON
Przesyłka.id_Typ_Przesyłki = Typ_Przesyłki.id_Typ_Przesyłki INNER JOIN Kurs ON
Przesyłka.id_Kurs = Kurs.id_Kurs INNER JOIN Pracownik ON
Kurs.id_Pracownik = Pracownik.id_Pracownik
WITH CHECK OPTION

```

	id_Przesyłka	Typ_Przesyłki	Odbiorca_Imię	Odbiorca_Nazwisko	Nadawca_Imię	Nadawca_Nazwisko	Do_miasta	Na_ulicę	Do_mieszkania	Pracownik_Imię	Pracownik_Nazwisko
1	6	standardowa	Kamil	Miecz	Konrad	Pniak	Kozy	Wapienna	4	Kamil	Podpałka
2	8	gabarytowa	Kamil	Miecz	Radosław	Grzywa	Kozy	Wapienna	4	Kamil	Podpałka
3	10	priorytetowa	Mieczysław	Stant	Mateusz	Królik	Katowice	Kościuszki	321	Sławomir	Mróz
4	12	delikatna	Radosław	Grzywa	Ignacy	Mosin	Sandomierz	Zakościelna	12	Sławomir	Mróz
5	13	gabarytowa	Konrad	Pniak	Borys	Maciej	Świnoujście	Wodna	150	Mikołaj	Proba
6	15	priorytetowa	Borys	Maciej	Kamil	Miecz	Cieszyn	Zwierzyniecka	1/12	Dominik	Słuch
7	16	gabarytowa	Hieronim	Piecuch	Mieczysław	Stant	Kostrzyn	Krótką	27	Mikołaj	Proba
8	17	gabarytowa	Jarosław	Kowala	Tadeusz	Trzeciak	Łódź	Marszałka	10	Mikołaj	Proba
9	20	standardowa	Tomasz	Monter	Kamil	Miecz	Mielno	Nadmorska	2/43	Magdalena	Nimcz

	Pracownik_Nazwisko	Data_nadania	Data_wymaganego_...	Data_rzeczywistego_dostarczenia	Waga	Jednostka_wagi	Opłata	Waluta	Nr_dokumentu_przewozowego	Typ_płatności
1	Podpałka	2017-12-28 12:0...	2018-01-05 00:00:00	2018-01-05 13:00:00	2	KG	0,00	PLN	8515289	gotówka
2	Podpałka	2018-01-04 13:3...	2018-01-10 00:00:00	2018-01-05 19:00:00	370	KG	2000,00	PLN	5617244	za pobraniem
3	Mróz	2018-01-02 10:0...	2018-01-06 00:00:00	2018-01-07 14:33:00	15	KG	0,00	PLN	5416757	karta
4	Mróz	2017-12-29 15:0...	2018-01-06 00:00:00	2018-01-07 17:37:00	26	KG	250,00	PLN	7651621	za pobraniem
5	Proba	2018-01-02 09:0...	2018-01-15 00:00:00	2018-01-10 09:00:00	2000	KG	0,00	PLN	1561885	gotówka
6	Słuch	2018-01-07 18:0...	2018-01-09 00:00:00	2018-01-08 08:00:00	35	KG	450,00	PLN	4575871	za pobraniem
7	Proba	2018-01-07 13:0...	2018-01-11 00:00:00	2018-01-09 14:00:00	800	KG	0,00	PLN	6581891	gotówka
8	Proba	2018-01-07 16:0...	2018-01-13 00:00:00	2018-01-09 20:00:00	600	KG	0,00	PLN	7561857	karta
9	Nimcz	2018-01-15 14:0...	2018-01-24 00:00:00	NULL	5	KG	260,00	PLN	7458250	za pobraniem

FIRMA KURIERSKA

FAST CHEAP PRECISELY
DELIVER!



autor: Łukasz Czepielik
WBMil Informatyka sem.III