

Projekt BADA cz. I

Port tranzytowy

Paweł Popiołek 303716

Mateusz Borkowski 303686

Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych

24 listopada 2020

Spis treści

1. Zakres i cel projektu	2
1.1. Opis działalności biznesowej	2
2. Definicja systemu	2
2.1. Perspektywy użytkowników	2
3. Model koncepcyjny	2
3.1. Definicja zbiorów encji określonych w projekcie	2
3.2. Ustalenie związków między encjami i ich typów	12
3.3. Określenie atrybutów i ich dziedzin	14
3.4. Dodatkowe reguły integralnościowe	29
3.5. Klucze kandydujące i klucze główne	29
3.6. Schemat ER na poziomie koncepcyjnym	32
3.7. Problem pułapek szczelinowych i wachlarzowych	32
3.7.1. Pułapka szczelinowa	32
3.7.2. Pułapka wachlarzowa	33
4. Model logiczny	34
4.1. Charakterystyka modelu relacyjnego	34
4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym	34
4.3. Proces normalizacji	34
4.4. Schemat ER na poziomie modelu logicznego	35
4.5. Więzy integralności	37
4.6. Proces denormalizacji	37
5. Faza fizyczna	38
5.1. Projekt transakcji i weryfikacja ich wykonalności	38
5.2. Strojenie bazy danych - dobór indeksów	38
5.3. Skrypt SQL zakładający bazę danych	39
5.4. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych	54
5.4.1. Zapełnienie danymi	54
5.4.2. Przykładowe zapytania	58

1. Zakres i cel projektu

Celem pierwszej części projektu jest zaprojektowanie oraz zaimplementowanie relacyjnej bazy danych. Obejmuje ona realizację: fazy koncepcyjnej, fazy projektowania logicznego, oraz fazy fizycznej.

1.1. Opis działalności biznesowej

Realizowany projekt dotyczy przedsiębiorstwa zajmującego się portem tranzytowym. Działalność portu tranzytowego polega na obsługiwaniu zleceń przekazanych przez klientów. W ramach zlecenia, port ma za zadanie obsłużyć dostawę ładunków (zorganizowaną przez klienta), następnie przechować ładunki w magazynie i w odpowiednim czasie wydać je dla drugiego transportu, również zorganizowanego przez tego samego klienta. Każdy ładunek ma określoną masę oraz wielkość. W celu przechowania ładunku, zostaje mu przydzielone miejsce (magazyn, regał, półka) oraz czas składowania, który nie musi być tożsamy z datą dostawy oraz odbioru zawartymi w zleceniu, ponieważ dopuszcza się zaistnienie opóźnień w transportach. Port zatrudnia pracowników biurowych oraz magazynierów, którzy muszą dodatkowo złożyć badania lekarskie oraz uprawnienia na wózek widłowy. Magazyn jest obsługiwany przez określonych magazynierów, którzy mogą mieć do dyspozycji wózki widłowe przypisane do danego magazynu.

2. Definicja systemu

2.1. Perspektywy użytkowników

- **Właściciel** - posiada dostęp do wszystkich danych,
- **Pracownik** - posiada wgląd w swoje dane osobowe, wysokość i datę wypłaty najbliższego wynagrodzenia, oraz opis swojego stanowiska (pracownik na stanowisku *pracownik biurowy* widzi w szczególności dane klientów portu, oraz dane dotyczące wszystkich zleceń),
- **Magazynier** - magazynier jest pracownikiem który w szczególności posiada również wgląd do daty ważności swojego badania lekarskiego, oraz daty wygaśnięcia swoich uprawnień na wózek widłowy, ma on również dostęp do informacji o magazynach do których jest aktualnie przypisany (szczegóły dotyczące pojemności magazynu, miejsc składowania w magazynie, oraz ładunków składowanych w magazynie),
- **Klient** - widzi swoje dane, oraz szczegóły dotyczące swoich zleceń, włącznie ze szczegółami dotyczącymi ładunku (bez wiedzy o miejscu składowania ładunku).

3. Model konceptualny

3.1. Definicja zbiorów encji określonych w projekcie

Entity Name	Primary Unique Identifier	# Attributes
Klient	Nr_klienta	5
Ładunek	Nr_ładunku	5
Magazyn	Nr_magazynu	3
Magazynier		2
Miejsce_skladowania	Nr_miejsca	4
Port	Nr_portu	5
Pracownik	Nr_pracownika	12
Wózek_widlowy	Nr_wozka	7
Zlecenie	Nr_zlecenia	6

Zbiór encji

Entity 'Port'

Caption	Port
Primary Unique Identifier	PK_Port

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Nr_portu		SmallInt	YES	NO	NO	Unikatowy numer portu
NO	Nazwa		VarChar(30)	YES	NO	NO	
NO	Adres		VarChar(400)	YES	NO	NO	Adres portu, pole segmentowe. Zawiera: miasto, ulica, numer lokalu, kod pocztowy, kraj.
NO	Dane_kontaktowe		VarChar(100)	YES	NO	NO	Dane kontaktowe do portu, pole segmentowe. Zawiera: nr telefonu, adres email.
NO	Wlasciciel		VarChar(100)	YES	NO	NO	Właściciel portu, pole wielowartościowe, segmentowe. Imię i nazwisko właściciela.

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
Posiada	Non-Identifying	Port - Magazyn	1..1 - 1..m
ma_klienta	Non-Identifying	Port - Klient	1..1 - 0..m
Zatrudnia	Non-Identifying	Port - Pracownik	1..1 - 0..m
posiada	Non-Identifying	Port - Wozek_widlowy	1..1 - 0..m

Unique Identifiers

PUI	Unique Identifier Name	Attributes
YES	PK_Port	Nr_portu

Description

Encja port

Encja port

Entity 'Pracownik'

Caption	Pracownik
Primary Unique Identifier	PK_Pracownik

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Nr_pracownika		SmallInt	YES	NO	NO	Unikalny numer pracownika.
NO	Imie		VarChar(30)	YES	NO	NO	Imię pracownika.
NO	Nazwisko		VarChar(30)	YES	NO	NO	Nazwisko pracownika.
NO	Plec	PlecD	Character(1)	YES	NO	NO	Płeć pracownika.
NO	Data_urodzenia		Date	YES	NO	NO	Data urodzenia pracownika.
NO	Data_zatrudnienia		Date	YES	NO	NO	Data zatrudnienia pracownika.
NO	Dane_kontaktowe		VarChar(100)	YES	NO	NO	Dane kontaktowe do pracownika, pole segmentowe. Zawiera: nr telefonu, adres email.
NO	Stanowisko		VarChar(30)	YES	NO	NO	Stanowisko pracownika.
NO	Adres		VarChar(400)	YES	NO	NO	Adres pracownika, pole segmentowe. Zawiera: miasto, ulica, numer lokalu, kod pocztowy, kraj.
NO	Wynagrodzenie		SmallInt	YES	NO	NO	Wynagrodzenie pracownika w PLN.
NO	PESEL		Character(11)	NO	NO	NO	Nr PESEL pracownika.
NO	Nr_konta		VarChar(50)	NO	NO	NO	Numer konta bankowego pracownika

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
Zatrudnia	Non-Identifying	Port - Pracownik	1..1 - 0..m

Unique Identifiers

PUI	Unique Identifier Name	Attributes
YES	PK_Pracownik	Nr_pracownika

Description

Encja Pracownik

Encja pracownik

Entity 'Magazynier' (child entity of Pracownik)

Caption	Magazynier
---------	------------

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
NO	Data_badania_lekarskiego		Date	YES	NO	NO	Data wydania badania lekarskiego
NO	Data_uprawnien_na_wozek		Date	YES	NO	NO	Data wygaśnięcia uprawnień na wózek widłowy

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
Obsluguje	Non-Identifying	Magazynier - Magazyn	0..n - 0..m

Description

Encja Magazynier

Encja magazynier

Entity 'Klient'

Caption	Klient
Primary Unique Identifier	PK_Klient

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Nr_klienta		Integer	YES	NO	NO	Unikalny numer klienta.
NO	Nazwa_firmy		VarChar(100)	YES	NO	NO	Opisuje nazwę firmy.
NO	Adres		VarChar(400)	YES	NO	NO	Adres klienta, pole segmentowe. Zawiera: miasto, ulica, numer lokalu, kod pocztowy.
NO	Dane_kontaktowe		VarChar(100)	YES	NO	NO	Dane kontaktowe do klienta, pole segmentowe. Zawiera: nr telefonu, adres email.
NO	Nr_konta		VarChar(50)	NO	NO	NO	Numer konta bankowego klienta

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
ma_klienta	Non-Identifying	Port - Klient	1..1 - 0..m
ma_zlecenia	Non-Identifying	Zlecenie - Klient	0..n - 1..1

Unique Identifiers

PUI	Unique Identifier Name	Attributes
YES	PK_Klient	Nr_klienta
NO	AK_Klient	Nazwa_firmy

Description

Encja Klient

Encja klient

Entity 'Zlecenie'

Caption	Zlecenie
Primary Unique Identifier	PK_Zlecenie

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Nr_zlecenia		Integer	YES	NO	NO	Unikalny numer zlecenia.
NO	Cena		Integer	YES	NO	NO	Cena zlecenia.
NO	Data_przyjecia		Date	YES	NO	NO	Data przyjęcia zlecenia.
NO	Data_dostawy		Date	YES	NO	NO	Data dostawy ładunków
NO	Data_odbioru		Date	YES	NO	NO	Data odbioru ładunków
NO	Status	StatusD	Character(1)	YES	NO	NO	Pole opisujące status zlecenia

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
zawiera_ladunek	Non-Identifying	Zlecenie - Ladunek	1..1 - 1..m
ma_zlecenia	Non-Identifying	Zlecenie - Klient	0..n - 1..1

Unique Identifiers

PUI	Unique Identifier Name	Attributes
YES	PK_Zlecenie	Nr_zlecenia

Description

Encja zlecenie.

Encja zlecenie

Entity 'Magazyn'

Caption	Magazyn
Primary Unique Identifier	PK_Magazyn

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Nr_magazynu		Integer	YES	NO	NO	Unikatowy numer magazynu.
NO	Pojemnosc		Integer	YES	NO	NO	Całkowita pojemność magazynu podana w ilości europalet ładunku które możemy w nim zmieścić.
NO	Nr_telefonu		VarChar(20)	NO	NO	NO	Numer telefonu do danego magazynu.

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
Posiada	Non-Identifying	Port - Magazyn	1..1 - 1..m
Obsluguje	Non-Identifying	Magazynier - Magazyn	0..n - 0..m
przynalezy	Non-Identifying	Magazyn - Wozek_widlowy	0..1 - 0..m
ma_miejsca_skladowania	Non-Identifying	Magazyn - Miejsce_skladowania	1..1 - 1..m

Unique Identifiers

PUI	Unique Identifier Name	Attributes
YES	PK_Magazyn	Nr_magazynu

Description

Encja Magazyn

Encja magazyn

Entity 'Wozek_widlowy'

Caption	Wozek_widlowy
Primary Unique Identifier	PK_Wozek_widlowy

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Nr_wozka		Integer	YES	NO	NO	Unikatowy numer wózka widłowego
NO	Nr_fabryczny		VarChar(30)	YES	NO	NO	Numer fabryczny wózka widłowego
NO	Marka		VarChar(50)	YES	NO	NO	Marka wózka widłowego
NO	Model		VarChar(50)	YES	NO	NO	Model wózka widłowego
NO	Rok_produkcji		SmallInt	YES	NO	NO	Rok produkcji wózka widłowego
NO	Udzwig		SmallInt	YES	NO	NO	Udzwig wózka widłowego w kilogramach
NO	Wysokosc_podnoszenia		SmallInt	YES	NO	NO	Wysokość podnoszenia wózka widłowego wyrażona w metrach

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
przynalezy	Non-Identifying	Magazyn - Wozek_widlowy	0..1 - 0..m
posiada	Non-Identifying	Port - Wozek_widlowy	1..1 - 0..m

Unique Identifiers

PUI	Unique Identifier Name	Attributes
YES	PK_Wozek_widlowy	Nr_wozka
NO	AK_Wozek_widlowy	Nr_fabryczny

Description

Encja wózka widłowego

Encja wózek widłowy

Entity 'Ladunek'

Caption	Ladunek
Primary Unique Identifier	PK_Ladunek

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Nr_ladunku		Integer	YES	NO	NO	Unikalny numer ładunku.
NO	Objetosc		SmallInt	YES	NO	NO	Objętość danego ładunku wyrażona w ilości europalet.
NO	Masa		SmallInt	YES	NO	NO	Masa ładunku wyrażona w kilogramach (z dokładnością do kilograma).
NO	Data_poczatku		Date	YES	NO	NO	
NO	Data_konca		Date	YES	NO	NO	

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
zawiera_ladunek	Non-Identifying	Zlecenie - Ladunek	1..1 - 1..m
skladowany_w	Non-Identifying	Ladunek - Miejsce_skladowania	0..n - 1..m

Unique Identifiers

PUI	Unique Identifier Name	Attributes
YES	PK_Ladunek	Nr_ladunku

Description

Encja Ładunek

Encja Ładunek

Entity 'Skladowanie'

Caption	Skladowanie
Primary Unique Identifier	PK_Skladowanie

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Nr_skladowania		Integer	YES	NO	NO	Unikatowy numer składowania ładunku.
NO	Data_poczatku		Date	YES	NO	NO	Data początku składowania ładunku.
NO	Data_konca		Date	YES	NO	NO	Data końca składowania ładunku.

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
jest_skladowany	Non-Identifying	Ladunek - Skladowanie	1..1 - 1..1
w_miejscu	Non-Identifying	Skladowanie - Miejsce_skladowania	0..n - 1..m

Unique Identifiers

PUI	Unique Identifier Name	Attributes
YES	PK_Skladowanie	Nr_skladowania

Description

Encja Składowanie.

Encja składowanie

Entity 'Miejsce_skladowania'

Caption	Miejsce_skladowania
Primary Unique Identifier	PK_Miejsce_Skladowania

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Nr_miejsca		Integer	YES	NO	NO	Unikatowy numer miejsca składowania ładunku
NO	Nr_regalu		Integer	YES	NO	NO	Numer regału
NO	Nr_polki		Integer	YES	NO	NO	Numer półki
NO	Wysokosc		SmallInt	YES	NO	NO	Wysokość nad ziemią miejsca składowania (w metrach)

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
ma_miejsca_skladowania	Non-Identifying	Magazyn - Miejsce_skladowania	1..1 - 1..m
skladowany_w	Non-Identifying	Ladunek - Miejsce_skladowania	0..n - 1..m

Unique Identifiers

PUI	Unique Identifier Name	Attributes
YES	PK_Miejsce_Skladowania	Nr_miejsca

Description

Encja Miejsce składowania.

Encja miejsce składowania

3.2. Ustalenie związków między encjami i ich typów

Relationship Name	Relationship Type	Relationship Between	Cardinality
ma_klienta	Non-Identifying	Port - Klient	1..1 - 0..m
ma_miejsca_skladowania	Non-Identifying	Magazyn - Miejsce_skladowania	1..1 - 1..m
ma_zlecenia	Non-Identifying	Zlecenie - Klient	0..n - 1..1
Obsluguje	Non-Identifying	Magazynier - Magazyn	0..n - 0..m
Posiada	Non-Identifying	Port - Magazyn	1..1 - 1..m
posiada	Non-Identifying	Port - Wozek_widlowy	1..1 - 0..m
przynalezy	Non-Identifying	Magazyn - Wozek_widlowy	0..1 - 0..m
skladowany_w	Non-Identifying	Ladunek - Miejsce_skladowania	0..n - 1..m
Zatrudnia	Non-Identifying	Port - Pracownik	1..1 - 0..m
zawiera_ladunek	Non-Identifying	Zlecenie - Ladunek	1..1 - 1..m

Związki między encjami i ich typy

Inheritance Name	Exclu sive	Comp lete	Parent Entity
Specjalizacja	NO	NO	Pracownik

Dziedziczenie

3.3. Określenie atrybutów i ich dziedzin

Attribute Name	Entity Name	Domain	Data Type	M	PUI
Adres	Klient		VarChar(400)	YES	NO
Dane_kontaktowe	Klient		VarChar(100)	YES	NO
Nazwa_firmy	Klient		VarChar(100)	YES	NO
Nr_klienta	Klient		Integer	YES	YES
Nr_konta	Klient		VarChar(50)	NO	NO
Data_konca	Ladunek		Date	YES	NO
Data_poczatku	Ladunek		Date	YES	NO
Masa	Ladunek		SmallInt	YES	NO
Nr_ladunku	Ladunek		Integer	YES	YES
Objetosc	Ladunek		SmallInt	YES	NO
Nr_magazynu	Magazyn		Integer	YES	YES
Nr_telefonu	Magazyn		VarChar(20)	NO	NO
Pojemnosc	Magazyn		Integer	YES	NO
Data_badiana_lekarskiego	Magazynier		Date	YES	NO
Data_uprawnien_na_wozek	Magazynier		Date	YES	NO
Nr_miejsca	Miejsce_skladowania		Integer	YES	YES
Nr_polki	Miejsce_skladowania		Integer	YES	NO
Nr_regalu	Miejsce_skladowania		Integer	YES	NO
Wysokosc	Miejsce_skladowania		SmallInt	YES	NO
Adres	Port		VarChar(400)	YES	NO
Dane_kontaktowe	Port		VarChar(100)	YES	NO
Nazwa	Port		VarChar(30)	YES	NO
Nr_portu	Port		SmallInt	YES	YES
Wlasciciel	Port		VarChar(100)	YES	NO
Adres	Pracownik		VarChar(400)	YES	NO
Dane_kontaktowe	Pracownik		VarChar(100)	YES	NO
Data_urodzenia	Pracownik		Date	YES	NO
Data_zatrudnienia	Pracownik		Date	YES	NO
Imie	Pracownik		VarChar(30)	YES	NO
Nazwisko	Pracownik		VarChar(30)	YES	NO
Nr_konta	Pracownik		VarChar(50)	NO	NO
Nr_pracownika	Pracownik		SmallInt	YES	YES
PESEL	Pracownik		Character(11)	NO	NO
Plec	Pracownik	PlecD	Character(1)	YES	NO
Stanowisko	Pracownik		VarChar(30)	YES	NO
Wynagordzenie	Pracownik		SmallInt	YES	NO
Marka	Wozek_widlowy		VarChar(50)	YES	NO

Attribute Name	Entity Name	Domain	Data Type	M	PUI
Model	Wozek_widlowy		VarChar(50)	YES	NO
Nr_fabryczny	Wozek_widlowy		VarChar(30)	YES	NO
Nr_wozka	Wozek_widlowy		Integer	YES	YES
Rok_produkcji	Wozek_widlowy		SmallInt	YES	NO
Udzieg	Wozek_widlowy		SmallInt	YES	NO
Wysokosc_podnoszenia	Wozek_widlowy		SmallInt	YES	NO
Cena	Zlecenie		Integer	YES	NO
Data_dostawy	Zlecenie		Date	YES	NO
Data_odbioru	Zlecenie		Date	YES	NO
Data_przyjecia	Zlecenie		Date	YES	NO
Nr_zlecenia	Zlecenie		Integer	YES	YES
Status	Zlecenie	StatusD	Character(1)	YES	NO

Zbiór atrybutów

Attribute 'Adres'

Caption	Adres
Entity Name	Klient
Data Type	VarChar(400)
Mandatory	YES

Description

Adres klienta, pole segmentowe. Zawiera: miasto, ulica, numer lokalu, kod pocztowy.

Attribute 'Dane_kontaktowe'

Caption	Dane_kontaktowe
Entity Name	Klient
Data Type	VarChar(100)
Mandatory	YES

Description

Dane kontaktowe do klienta, pole segmentowe. Zawiera: nr telefonu, adres email.

Attribute 'Nazwa_firmy'

Caption	Nazwa_firmy
Entity Name	Klient
Data Type	VarChar(100)
Mandatory	YES

Valid Values

Valid Values Type	Valid Values
Enumeration	{ "injakulm" }

Description

Opisuje nazwę firmy.

Attribute 'Nr_klienta'

Caption	Nr_klienta
Entity Name	Klient
Primary Unique Identifier	YES

Data Type	Integer
Mandatory	YES

Description

Unikalny numer klienta.

Attribute 'Nr_konta'

Caption	Nr konta
Entity Name	Klient
Data Type	VarChar(50)
Mandatory	NO

Description

Numer konta bankowego klienta

Attribute 'Lokalizacja_skladowania'

Caption	Lokalizacja_skladowania
Entity Name	Ladunek
Data Type	VarChar(400)
Mandatory	YES

Description

Pole segmentowe opisujące miejsce składowania w magazynie (regał, półka)

Attribute 'Masa'

Caption	Masa
Entity Name	Ladunek
Data Type	SmallInt
Mandatory	YES

Description

Masa ładunku wyrażona w kilogramach (z dokładnością do kilograma).

Attribute 'Nr_ladunku'

Caption	Nr_ladunku
Entity Name	Ladunek
Primary Unique Identifier	YES
Data Type	Integer
Mandatory	YES

Data Type	Integer
Mandatory	YES

Description

Unikalny numer klienta.

Attribute 'Nr_konta'

Caption	Nr_konta
Entity Name	Klient
Data Type	VarChar(50)
Mandatory	NO

Description

Numer konta bankowego klienta

Attribute 'Data_konca'

Caption	Data_konca
Entity Name	Ladunek
Data Type	Date
Mandatory	YES

Attribute 'Data_poczatku'

Caption	Data_poczatku
Entity Name	Ladunek
Data Type	Date
Mandatory	YES

Attribute 'Masa'

Caption	Masa
Entity Name	Ladunek
Data Type	SmallInt
Mandatory	YES

Description

Masa ładunku wyrażona w kilogramach (z dokładnością do kilograma).

Attribute 'Nr_ladunku'

Caption	Nr_ladunku
Entity Name	Ladunek
Primary Unique Identifier	YES

Data Type	Integer
Mandatory	YES

Description

Unikalny numer ładunku.

Attribute 'Objetosc'

Caption	Objetosc
Entity Name	Ladunek
Data Type	SmallInt
Mandatory	YES

Description

Objętość danego ładunku wyrażona w ilości europalet.

Attribute 'Nr_magazynu'

Caption	Nr_magazynu
Entity Name	Magazyn
Primary Unique Identifier	YES
Data Type	Integer
Mandatory	YES

Description

Unikatowy numer magazynu.

Attribute 'Nr_telefonu'

Caption	Nr_telefonu
Entity Name	Magazyn
Data Type	VarChar(20)
Mandatory	NO

Description

Numer telefonu do danego magazynu.

Attribute 'Pojemnosc'

Caption	Pojemnosc
Entity Name	Magazyn
Data Type	Integer
Mandatory	YES

Description

Całkowita pojemność magazynu podana w ilości europalet ładunku które możemy w nim zmieścić.

Attribute 'Data_badania_lekarskiego'

Caption	Data_badania_lekarskiego
Entity Name	Magazynier
Data Type	Date
Mandatory	YES

Description

Data wydania badania lekarskiego

Attribute 'Data_uprawnien_na_wozek'

Caption	Data_uprawnien_na_wozek
Entity Name	Magazynier
Data Type	Date
Mandatory	YES

Description

Data wygaśnięcia uprawnień na wózek widłowy

Attribute 'Nr_miejsca'

Caption	Nr_miejsca
Entity Name	Miejsce_skladowania
Primary Unique Identifier	YES
Data Type	Integer
Mandatory	YES

Description

Unikatowy numer miejsca składowania ładunku

Attribute 'Nr_polki'

Caption	Nr_polki
Entity Name	Miejsce_skladowania
Data Type	Integer
Mandatory	YES

Description

Numer półki

Attribute 'Nr_regalu'

Caption	Nr_regalu
Entity Name	Miejsce_skladowania
Data Type	Integer
Mandatory	YES

Description

Numer regału

Attribute 'Wysokosc'

Caption	Wysokosc
Entity Name	Miejsce_skladowania
Data Type	SmallInt
Mandatory	YES

Description

Wysokość nad ziemią miejsca składowania (w metrach)

Attribute 'Adres'

Caption	Adres
Entity Name	Port
Data Type	VarChar(400)
Mandatory	YES

Description

Adres portu, pole segmentowe. Zawiera: miasto, ulica, numer lokalu, kod pocztowy, kraj.

Attribute 'Dane_kontaktowe'

Caption	Dane_kontaktowe
Entity Name	Port
Data Type	VarChar(100)
Mandatory	YES

Description

Dane kontaktowe do portu, pole segmentowe. Zawiera: nr telefonu, adres email.

Attribute 'Nazwa'

Caption	Nazwa
Entity Name	Port
Data Type	VarChar(30)
Mandatory	YES

Attribute 'Nr_portu'

Caption	Nr_portu
Entity Name	Port
Primary Unique Identifier	YES
Data Type	SmallInt
Mandatory	YES

Description

Unikatowy numer portu

Attribute 'Wlasciciel'

Caption	Wlasciciel
Entity Name	Port
Data Type	VarChar(100)
Mandatory	YES

Description

Właściciel portu, pole wielowartościowe, segmentowe. Imię i nazwisko właściciela.

Attribute 'Adres'

Caption	Adres
Entity Name	Pracownik
Data Type	VarChar(400)
Mandatory	YES

Description

Adres pracownika, pole segmentowe. Zawiera: miasto, ulica, numer lokalu, kod pocztowy, kraj.

Attribute 'Dane_kontaktowe'

Caption	Dane_kontaktowe
---------	-----------------

Entity Name	Pracownik
Data Type	VarChar(100)
Mandatory	YES

Description

Dane kontaktowe do pracownika, pole segmentowe. Zawiera: nr telefonu, adres email.

Attribute 'Data_urodzenia'

Caption	Data_urodzenia
Entity Name	Pracownik
Data Type	Date
Mandatory	YES

Description

Data urodzenia pracownika.

Attribute 'Data_zatrudnienia'

Caption	Data_zatrudnienia
Entity Name	Pracownik
Data Type	Date
Mandatory	YES

Description

Data zatrudnienia pracownika.

Attribute 'Imie'

Caption	Imie
Entity Name	Pracownik
Data Type	VarChar(30)
Mandatory	YES

Description

Imię pracownika.

Attribute 'Nazwisko'

Caption	Nazwisko
Entity Name	Pracownik
Data Type	VarChar(30)
Mandatory	YES

Description

Nazwisko pracownika.

Attribute 'Nr_konta'

Caption	Nr konta
Entity Name	Pracownik
Data Type	VarChar(50)
Mandatory	NO

Description

Numer konta bankowego pracownika

Attribute 'Nr_pracownika'

Caption	Nr_pracownika
Entity Name	Pracownik
Primary Unique Identifier	YES
Data Type	SmallInt
Mandatory	YES

Description

Unikalny numer pracownika.

Attribute 'PESEL'

Caption	PESEL
Entity Name	Pracownik
Data Type	Character(11)
Mandatory	NO

Description

Nr PESEL pracownika.

Attribute 'Plec'

Caption	Plec
Entity Name	Pracownik
Domain	PlecD
Data Type	Character(1)
Mandatory	YES

Description

Płeć pracownika.

Attribute 'Stanowisko'

Caption	Stanowisko
Entity Name	Pracownik
Data Type	VarChar(30)
Mandatory	YES

Description

Stanowisko pracownika.

Attribute 'Wynagordzenie'

Caption	Wynagordzenie
Entity Name	Pracownik
Data Type	SmallInt
Mandatory	YES

Description

Wynagordzenie pracownika w PLN.

Attribute 'Marka'

Caption	Marka
Entity Name	Wozek_widlowy
Data Type	VarChar(50)
Mandatory	YES

Description

Marka wózka widłowego

Attribute 'Model'

Caption	Model
Entity Name	Wozek_widlowy
Data Type	VarChar(50)
Mandatory	YES

Description

Model wózka widłowego

Attribute 'Nr_fabryczny'

Caption	Nr_fabryczny
Entity Name	Wozek_widlowy
Data Type	VarChar(30)
Mandatory	YES

Description

Numer fabryczny wózka widłowego

Attribute 'Nr_wozka'

Caption	Nr_wozka
Entity Name	Wozek_widlowy
Primary Unique Identifier	YES
Data Type	Integer
Mandatory	YES

Description

Unikatowy numer wózka widłowego

Attribute 'Rok_pordukcji'

Caption	Rok_pordukcji
Entity Name	Wozek_widlowy
Data Type	SmallInt
Mandatory	YES

Description

Rok produkcji wózka widłowego

Attribute 'Udziwg'

Caption	Udziwg
Entity Name	Wozek_widlowy
Data Type	SmallInt
Mandatory	YES

Description

Udziwg wózka widłowego w kilogramach

Attribute 'Wysokosc_podnoszenia'

Caption	Wysokosc podnoszenia
Entity Name	Wozek widlowy
Data Type	SmallInt
Mandatory	YES

Description

Wysokość podnoszenia wózka widłowego wyrażona w metrach

Attribute 'Cena'

Caption	Cena
Entity Name	Zlecenie
Data Type	Integer
Mandatory	YES

Description

Cena zlecenia.

Attribute 'Data_dostawy'

Caption	Data_dostawy
Entity Name	Zlecenie
Data Type	Date
Mandatory	YES

Description

Data dostawy ładunków

Attribute 'Data_odbioru'

Caption	Data odbioru
Entity Name	Zlecenie
Data Type	Date
Mandatory	YES

Description

Data odbioru ładunków

Attribute 'Data_przyjecia'

Caption	Data_przyjecia
---------	----------------

Entity Name	Zlecenie
Data Type	Date
Mandatory	YES

Description

Data przyjęcia zlecenia.

Attribute 'Nr_zlecenia'

Caption	Nr_zlecenia
Entity Name	Zlecenie
Primary Unique Identifier	YES
Data Type	Integer
Mandatory	YES

Description

Unikalny numer zlecenia.

Attribute 'Status'

Caption	Status
Entity Name	Zlecenie
Domain	StatusD
Data Type	Character(1)
Mandatory	YES

Description

Pole opisujące status zlecenia

3.4. Dodatkowe reguły integralnościowe

Rule 'PlecR'

Caption	PlecR
Rule	Plec IN ('K', 'M', 'I')

Used in Domains

Domain Name
PlecD

Rule 'StatusR'

Caption	StatusR
Rule	Status IN ('Aktywne', 'Zrealizowane')

Used in Domains

Domain Name
StatusD

Własne reguły integralnościowe

3.5. Klucze kandydujące i klucze główne

UI Type	Unique Identifier Name	Entity Name
UI	AK Klient	Klient
PUI	PK Klient	Klient
PUI	PK Ladunek	Ladunek
PUI	PK Magazyn	Magazyn
PUI	PK Miejsce Składowania	Miejsce składowania
PUI	PK Port	Port
PUI	PK Pracownik	Pracownik
UI	AK Wozek widłowy	Wozek widłowy
PUI	PK Wozek widłowy	Wozek widłowy
PUI	PK Zlecenie	Zlecenie

Zbiór kluczy głównych i alternatywnych

Unique Identifier 'AK_Klient'

Caption	AK_Klient
Entity Name	Klient
Unique Identifier Type	Unique Identifier
Attribute Names	Nazwa_firmy

Unique Identifier 'PK_Klient'

Caption	PK_Klient
Entity Name	Klient
Unique Identifier Type	Primary Unique Identifier
Attribute Names	Nr_klienta

Unique Identifier 'PK_Ladunek'

Caption	PK_Ladunek
Entity Name	Ladunek
Unique Identifier Type	Primary Unique Identifier
Attribute Names	Nr_ladunku

Unique Identifier 'PK_Magazyn'

Caption	PK_Magazyn
Entity Name	Magazyn
Unique Identifier Type	Primary Unique Identifier
Attribute Names	Nr_magazynu

Unique Identifier 'PK_Miejsce_Skladowania'

Caption	PK_Miejsce_Skladowania
Entity Name	Miejsce_skladowania
Unique Identifier Type	Primary Unique Identifier
Attribute Names	Nr_miejsca

Unique Identifier 'PK_Port'

Caption	PK_Port
Entity Name	Port
Unique Identifier Type	Primary Unique Identifier
Attribute Names	Nr_portu

Unique Identifier 'PK_Pracownik'

Caption	PK_Pracownik
Entity Name	Pracownik
Unique Identifier Type	Primary Unique Identifier
Attribute Names	Nr_pracownika

Unique Identifier 'AK_Wozek_widlowy'

Caption	AK_Wozek_widlowy
Entity Name	Wozek_widlowy
Unique Identifier Type	Unique Identifier
Attribute Names	Nr_fabryczny

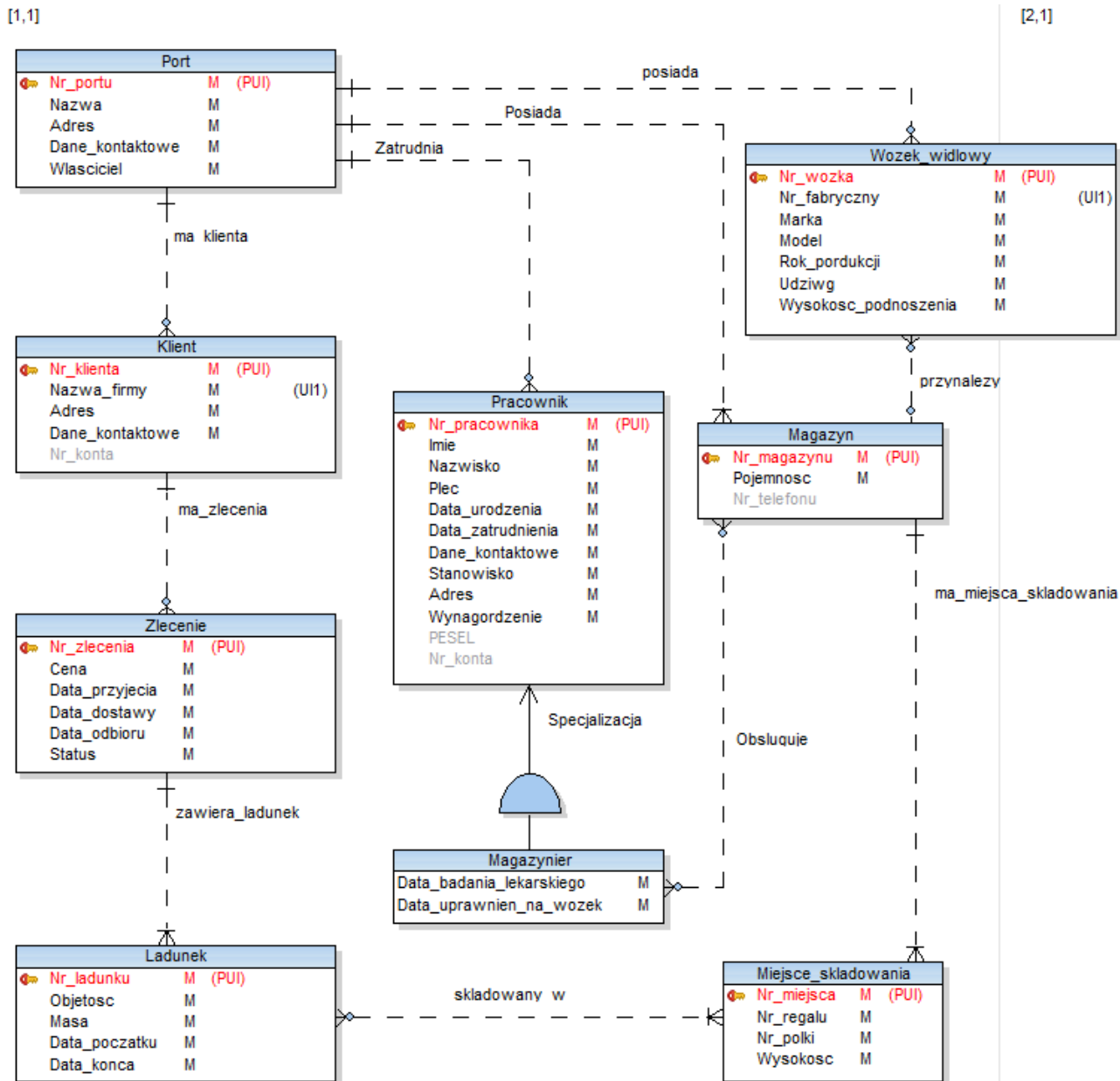
Unique Identifier 'PK_Wozek_widlowy'

Caption	PK_Wozek_widlowy
Entity Name	Wozek_widlowy
Unique Identifier Type	Primary Unique Identifier
Attribute Names	Nr_wozka

Unique Identifier 'PK_Zlecenie'

Caption	PK_Zlecenie
Entity Name	Zlecenie
Unique Identifier Type	Primary Unique Identifier
Attribute Names	Nr_zlecenia

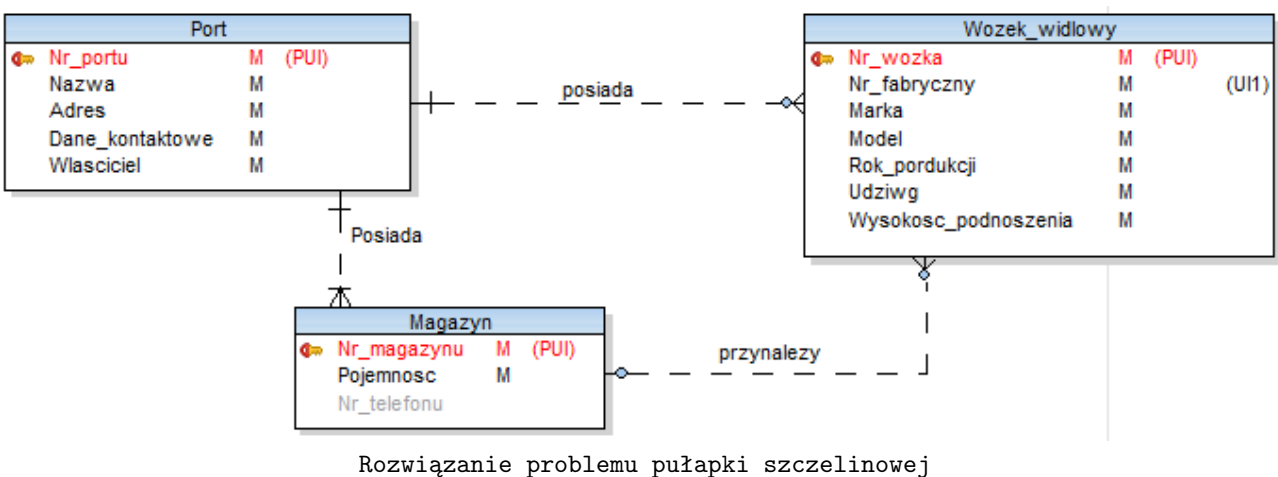
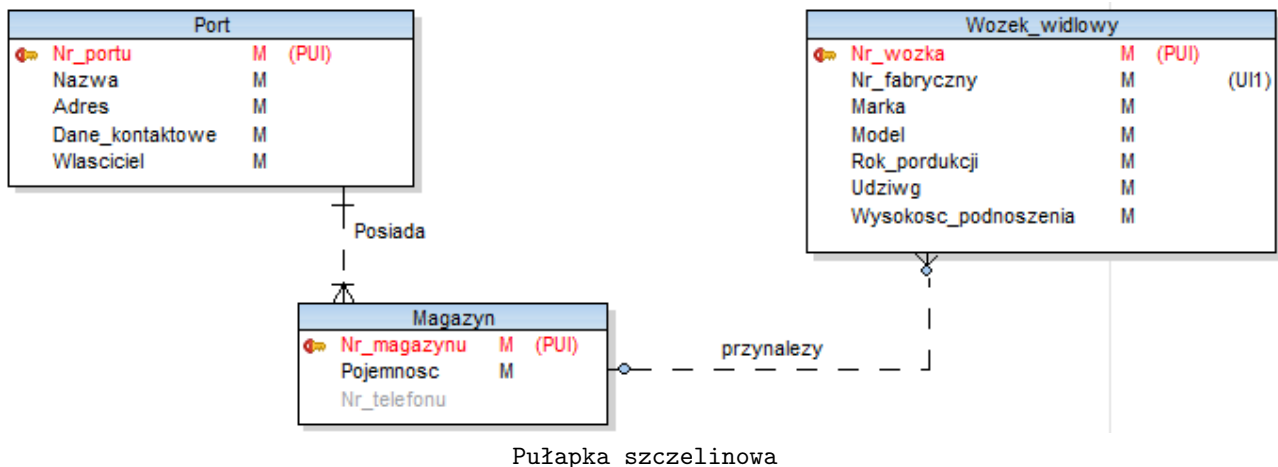
3.6. Schemat ER na poziomie konceptualnym



3.7. Problem pułapek szczelinowych i wachlarzowych

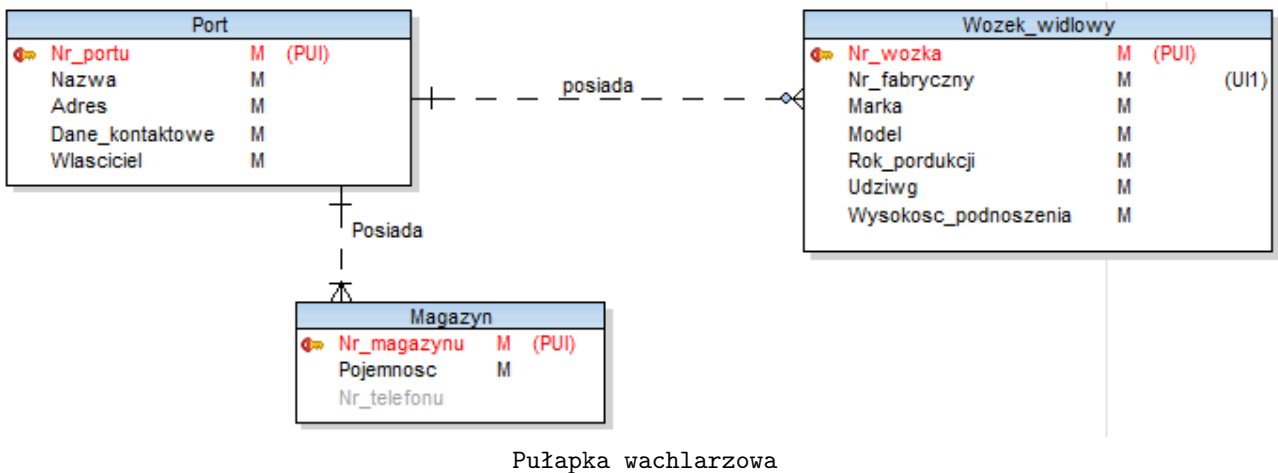
3.7.1. Pułapka szczelinowa

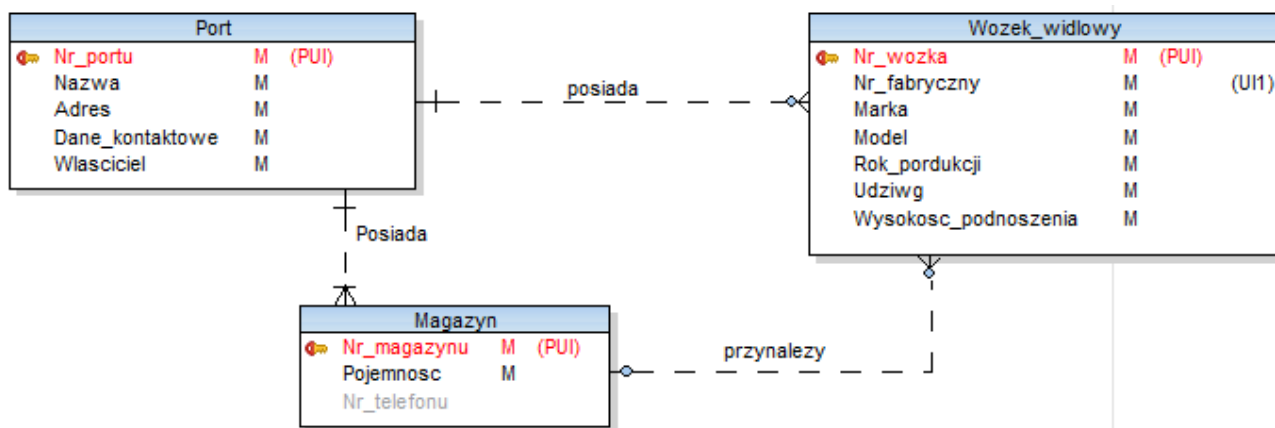
Pułapka szczelinowa pojawiła się u nas w relacji Port-Magazyn-Wozek_widlowy. Mogła wtedy zaistnieć sytuacja, gdzie na przykład nowo zakupiony wózek widłowy, który nie został jeszcze przypisany do konkretnego magazynu, nie istniałby na wyposażeniu portu. Rozwiązaniem tej pułapki było dodatkowe połączenie wózka widłowego z portem tak jak na poniższym rysunku. Dzięki tej modyfikacji, dalej można uzyskać informacje o wszystkich wózkach należących do portu, a nie tylko o tych, które są przypisane do magazynu.



3.7.2. Pułapka wachlarzowa

Żadna pułapka wachlarzowa nie wystąpiła w naszym modelu konceptualnym. Przykładem miejsca w którym mogłaby ona wystąpić jest relacja: Port-Magazyn-Wozek_widlowy. W przypadku gdyby port mógł posiadać wiele magazynów, oraz wiele wózków widłowych, a same magazyny nie były połączone z wózkami widłowymi, nie moglibyśmy określić który wózek widłowy jest w którym magazynie (mielibyśmy wachlarz możliwości) - co nie byłoby zgodne z naszym założeniem, że wózek widłowy przynależy do konkretnego, co najwyżej jednego magazynu. Oczwistym rozwiązaniem opisanego problemu jest połączenie magazynów z wózkami widłowymi odpowiednią relacją.





Rozwiązanie problemu pułapki wachlarzowej

4. Model logiczny

4.1. Charakterystyka modelu relacyjnego

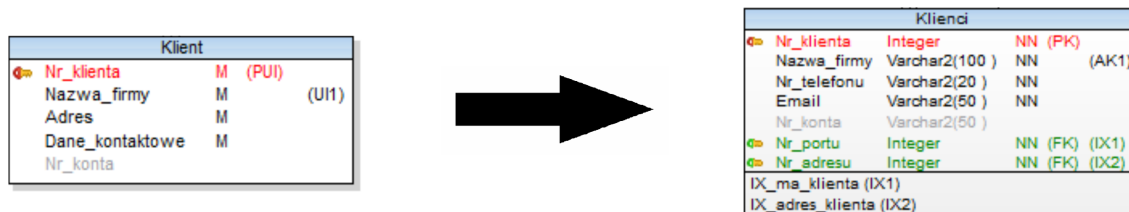
Model relacyjny składa się relacji, które są przedstawione za pomocą tabel, w których kolumny reprezentują atrybuty, a wiersze nazywa się krotkami. Cechą charakterystyczną, w porównaniu do modelu konceptualnego, jest zastąpienie każdej encji odpowiadającą jej tabelą - tutaj nazwy powinny być w liczbie mnogiej. Zauważalny jest również brak relacji wielu do wielu - muszą one zostać zastąpione dodatkową tabelą łączącą. Następną cechą, która nie pojawiła się w modelu koncepcyjnym, są klucze obce w tabelach, które umożliwiają tworzenie relacji między tabelami. Model relacyjny cechuje się daleko posuniętą niezależnością danych. Posiada również solidne podstawy pozwalające rozwiązywać problemy semantyki, spójności i redundancji danych.

4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym

- dla każdej encji z modelu konceptualnego zostaje utworzona odpowiadająca jej tabela, w której powinniśmy zmienić nazwę na liczbę mnogą w celu odróżnienia relacji od encji,
- należy usunąć związków wielu do wielu (poprzez zastąpienie ich dodatkową tabelą łączącą),
- identyfikujący atrybut encji staje się kluczem głównym tabeli, a wszystkie inne atrybuty encji stają się niegłównymi atrybutami tabeli,
- dla każdego związku *jeden do wiele* - dla strony jednej linii związku wstawiamy klucz główny tabeli, a dla drugiej strony (wiele linii) - w zależności od tego czy strona jest wymagana, nie dopuszczamy klucza obcego o wartości null. Jeśli strona wiele jest opcjonalna, dopuszczamy wartość null klucza obcego.

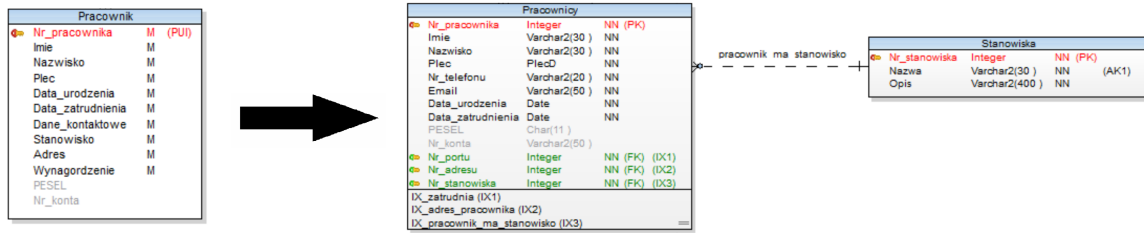
4.3. Proces normalizacji

Naszym założeniem na tym etapie było doprowadzenie bazy do 3 postaci normalnej. Jednak aby to było osiągalne należało po kolei normalizować naszą bazę. Na pierwszym etapie doprowadziliśmy wszystkie atrybuty do postaci elementarnej. Przykładowo, w tym celu w wielu miejscach, zamieniliśmy atrybut *Dane_kontaktowe* na *Email* oraz *Telefon*.



Doprowadzenie do atrybutów w postaci atomowej

Następnym etapem było usunięcie powtarzających się grup. Przykładem tego działania, jest wyprowadzenie stanowiska z pracownika do oddzielnej tabeli Stanowiska.



Wyprowadzenie powtarzających się grup do oddzielnej tabeli

Po tych i kilku innych działaniach, doprowadziliśmy bazę danych do pierwszej postaci normalnej. Ponieważ wszystkie klucze potencjalne w naszych relacjach są kluczami prostymi, to nasza baza danych jest również w drugiej postaci normalnej.

Teraz, aby doprowadzić tę bazę danych do 3 postaci normalnej należało upewnić się, że w każdej relacji, wszystkie niekluczowe elementy są określone całym kluczem i niczym innym. W naszej bazie, nie zaobserwowaliśmy relacji, które nie spełniałyby tego warunku.

4.4. Schemat ER na poziomie modelu logicznego

4.5. Więzy integralności

Więzy integralności zapewniają spójność danych. Są one pewnym zbiorem zasad nałożonym na tabele w bazie danych. W celu ich zapewnienia należało ustawić wszelkie klucze jako wartości *UNIQUE*. Co więcej dla odpowiednich atrybutów należało ustawić wartości *NOT NULL* - aby nie otrzymać pola *null* w nieporządanym miejscu, co mogłoby skutkować np. utratą pewnej części danych. Dla odpowiednich pól sprawdzamy za pomocą *CHECK* czy wpisywana wartość mieści się w dziedzinie oczekiwanych dla danego pola wartości. Przykładem naruszenia więzów integralności w naszej bazie danych mogłoby być zezwolenie na ustawienie pola *Nr.klienta* stanowiącego klucz obcy w tabeli *Zlecenia* jako *null*. Skutkowałoby to powstaniem pułapki szczelinowej dla relacji Klienci-Zlecenia i możliwym zaburzeniem integralności danych, poprzez utratę części informacji.

4.6. Proces denormalizacji

Zastanawialiśmy się, czy nie włączyć zawartości tabeli *Stanowiska* z powrotem do tabeli *Pracownicy*. Mogłoby to korzystnie wpłynąć na szybkość działania bazy danych w przypadku gdy tabela *Stanowiska* zawierałaby wiele rekordów. Stwierdziliśmy jednak, że korzyści nie przewyższą ewentualnych wad tego rozwiązania, takich jak wysokie ryzyko popełnienia błędu (na przykład literówki), który prowadziłby do otrzymywania niepoprawnych wyników. Dlatego też, podjęliśmy decyzję projektową aby w całej bazie nie przeprowadzać procesu denormalizacji, która mogła by negatywnie wpłynąć na integralność danych oraz utrudniałaby rozwój i przebudowę bazy w przyszłości. Mieliśmy na uwadze również rodzaj przedsiębiorstwa z jakim mamy do czynienia i uznaliśmy, że nie mamy w tym wypadku do czynienia z tysiącami transakcji na godzinę i dlatego lepiej będzie przełożyć spójność nad szybkość działania.

5. Faza fizyczna

5.1. Projekt transakcji i weryfikacja ich wykonalności

Transakcja	Potrzebne zasoby	Czy realizowalne
Dodawanie, usuwanie i modyfikowanie danych dotyczących portu	Porty, Adresy	TAK
Podgląd danych dotyczących portu	Porty, Adresy	TAK
Dodawanie, usuwanie i modyfikowanie danych dotyczących klientów	Klienci, Adresy	TAK
Podgląd danych dotyczących klientów	Klienci, Adresy	TAK
Dodawanie, usuwanie i modyfikowanie danych dotyczących pracowników	Pracownicy, Adresy, Stanowiska	TAK
Podgląd danych dotyczących pracowników	Pracownicy, Adresy, Stanowiska	TAK
Dodawanie, usuwanie i modyfikowanie danych dotyczących zleceń	Zlecenia, Klienci, Ladunki	TAK
Podgląd danych dotyczących zleceń	Zlecenia, Klienci, Ladunki	TAK
Dodawanie, usuwanie i modyfikowanie danych dotyczących magazynów	Magazyny, Wozki_widlowe, Magazynierzy	TAK
Podgląd danych dotyczących magazynów	Magazyny, Wozki_widlowe, Magazynierzy	TAK
Dodawanie, usuwanie i modyfikowanie danych dotyczących wózków widłowych	Wozki_widlowe, Modele, Marki	TAK
Podgląd danych dotyczących wózków widłowych	Wozki_widlowe, Modele, Marki	TAK
Dodawanie, usuwanie i modyfikowanie danych dotyczących ładunków	Ladunki, Skladowania, Miejsca_sklad, Zlecenia	TAK
Podgląd danych dotyczących ładunków	Ladunki, Skladowania, Miejsca_sklad, Zlecenia	TAK
Pogląd zawartości magazynów	Magazyny, Miejsca_sklad, Skladowania, Ladunki	TAK

Projekt transakcji

5.2. Strojenie bazy danych - dobór indeksów

Biorąc pod uwagę indeksy w automatycznie wygenerowanych skrypcie DDL, dla naszego modelu logicznego, zdecydowaliśmy się na usunięcie niektórych indeksów do klucza obcego, oraz pozostawienie tych dotyczących najczęstszych odwołań do klucza obcego (staraliśmy się zminimalizować liczbę rzadko używanych indeksów ze względu na dodatkowy koszt, który wiąże się z ich utrzymaniem i używaniem).

```

--create indexes
CREATE INDEX IX_adres_portu ON Porty (Nr_adresu)
CREATE INDEX IX_zawiera_ladunek ON Ladunki (Nr_zlecenia)
CREATE INDEX IX_ma_miejsce_skladowania ON Miejsca_sklad (Nr_magazynu)
CREATE INDEX IX_zatrudnia ON Pracownicy (Nr_portu)
CREATE INDEX IX_pracownik_ma_stanowisko ON Pracownicy (Nr_stanowiska)
CREATE INDEX IX_ma_klienta ON Klienci (Nr_portu)
CREATE INDEX IX_adres_klienta ON Klienci (Nr_adresu)

```

Indeksy dotyczące najczęstszych odwołań do klucza obcego w naszej bazie danych.

5.3. Skrypt SQL zakładający bazę danych

```

1  /*
2  Created: 17.11.2020
3  Modified: 26.11.2020
4  Model: Port_tranzytowy
5  Database: Oracle 12c Release 2
6  */
7
8
9  -- Create sequences section -----
10
11 CREATE SEQUENCE PortSeq1
12 INCREMENT BY 1
13 START WITH 1
14 NOMAXVALUE
15 NOMINVALUE
16 CACHE 20
17 /
18
19 CREATE SEQUENCE MagazynSeq1
20 INCREMENT BY 1
21 START WITH 1
22 NOMAXVALUE
23 NOMINVALUE
24 CACHE 20
25 /
26
27 CREATE SEQUENCE LadunekSeq1
28 INCREMENT BY 1
29 START WITH 1
30 NOMAXVALUE
31 NOMINVALUE
32 CACHE 20
33 /
34
35 CREATE SEQUENCE ZlecenieSeq1
36 INCREMENT BY 1
37 START WITH 1
38 NOMAXVALUE
39 NOMINVALUE
40 CACHE 20

```

```

41 /
42
43 CREATE SEQUENCE AdresSeq1
44     INCREMENT BY 1
45     START WITH 1
46     NOMAXVALUE
47     NOMINVALUE
48     CACHE 20
49 /
50
51 CREATE SEQUENCE WlascicielSeq1
52     INCREMENT BY 1
53     START WITH 1
54     NOMAXVALUE
55     NOMINVALUE
56     CACHE 20
57 /
58
59 CREATE SEQUENCE PracownikSeq1
60     INCREMENT BY 1
61     START WITH 1
62     NOMAXVALUE
63     NOMINVALUE
64     CACHE 20
65 /
66
67 CREATE SEQUENCE SkladSeq1
68     INCREMENT BY 1
69     START WITH 1
70     NOMAXVALUE
71     NOMINVALUE
72     CACHE 20
73 /
74
75 CREATE SEQUENCE Msc_sklSeq1
76     INCREMENT BY 1
77     START WITH 1
78     NOMAXVALUE
79     NOMINVALUE
80     CACHE 20
81 /
82
83 CREATE SEQUENCE WozekSeq1
84     INCREMENT BY 1
85     START WITH 1
86     NOMAXVALUE
87     NOMINVALUE
88     CACHE 20
89 /
90
91 CREATE SEQUENCE MarkaSeq1
92     INCREMENT BY 1
93     START WITH 1
94     NOMAXVALUE
95     NOMINVALUE
96     CACHE 20

```



```

97  /
98
99  CREATE SEQUENCE ModelSeq1
100    INCREMENT BY 1
101    START WITH 1
102    NOMAXVALUE
103    NOMINVALUE
104    CACHE 20
105  /
106
107  CREATE SEQUENCE WynagSeq1
108    INCREMENT BY 1
109    START WITH 1
110    NOMAXVALUE
111    NOMINVALUE
112    CACHE 20
113  /
114
115  CREATE SEQUENCE StanowiskoSeq1
116    INCREMENT BY 1
117    START WITH 1
118    NOMAXVALUE
119    NOMINVALUE
120    CACHE 20
121  /
122
123  CREATE SEQUENCE KlientSeq1
124    INCREMENT BY 1
125    START WITH 1
126    NOMAXVALUE
127    NOMINVALUE
128    CACHE 20
129  /
130
131  -- Create tables section -----
132
133  -- Table Porty
134
135  CREATE TABLE Porty(
136    Nr_portu Integer NOT NULL,
137    Nazwa Varchar2(30 ) NOT NULL,
138    Email Varchar2(50 ) NOT NULL,
139    Nr_telefonu Varchar2(20 ) NOT NULL,
140    Nr_adresu Integer NOT NULL
141  )
142  /
143
144
145
146  -- Add keys for table Porty
147
148  ALTER TABLE Porty ADD CONSTRAINT PK_Port PRIMARY KEY (Nr_portu)
149  /
150
151  -- Table Magazyny
152

```

```

153 CREATE TABLE Magazyny(
154     Nr_magazynu Integer NOT NULL,
155     Pojemnosc Integer NOT NULL,
156     Nr_telefonu Varchar2(20 ),
157     Nr_portu Integer NOT NULL
158 )
159 /
160
161
162 -- Add keys for table Magazyny
163
164 ALTER TABLE Magazyny ADD CONSTRAINT PK_Magazyn PRIMARY KEY (Nr_magazynu)
165 /
166
167 -- Table Ladunki
168
169 CREATE TABLE Ladunki(
170     Nr_ladunku Integer NOT NULL,
171     Objetosc Integer NOT NULL,
172     Masa Integer NOT NULL,
173     Nr_zlecenia Integer NOT NULL
174 )
175 /
176
177
178 -- Add keys for table Ladunki
179
180
181 ALTER TABLE Ladunki ADD CONSTRAINT PK_Ladunek PRIMARY KEY (Nr_ladunku)
182 /
183
184 -- Table Klienci
185
186 CREATE TABLE Klienci(
187     Nr_klienta Integer NOT NULL,
188     Nazwa_firmy Varchar2(100 ) NOT NULL,
189     Nr_telefonu Varchar2(20 ) NOT NULL,
190     Email Varchar2(50 ) NOT NULL,
191     Nr_konta Varchar2(50 ),
192     Nr_portu Integer NOT NULL,
193     Nr_adresu Integer NOT NULL
194 )
195 /
196
197
198 -- Add keys for table Klienci
199
200 ALTER TABLE Klienci ADD CONSTRAINT PK_Klient PRIMARY KEY (Nr_klienta)
201 /
202
203 ALTER TABLE Klienci ADD CONSTRAINT AK_Klient UNIQUE (Nazwa_firmy)
204 /
205
206 -- Table Pracownicy
207
208 CREATE TABLE Pracownicy(

```

```

209     Nr_pracownika Integer NOT NULL,
210     Imie Varchar2(30 ) NOT NULL,
211     Nazwisko Varchar2(30 ) NOT NULL,
212     Plec Char(1 ) NOT NULL
213         CHECK (Plec IN ('K', 'M', 'I')),
214     Nr_telefonu Varchar2(20 ) NOT NULL,
215     Email Varchar2(50 ) NOT NULL,
216     Data_urodzenia Date NOT NULL,
217     Data_zatrudnienia Date NOT NULL,
218     PESEL Char(11 ),
219     Nr_konta Varchar2(50 ),
220     Nr_portu Integer NOT NULL,
221     Nr_adresu Integer NOT NULL,
222     Nr_stanowiska Integer NOT NULL
223 )
224 /
225
226 -- Add keys for table Pracownicy
227
228 ALTER TABLE Pracownicy ADD CONSTRAINT PK_Pracownik PRIMARY KEY (Nr_pracownika)
229 /
230
231 -- Table Magazynierzy
232
233 CREATE TABLE Magazynierzy(
234     Nr_pracownika Integer NOT NULL,
235     Data_badania Date NOT NULL,
236     Data_uprawnien Date NOT NULL
237 )
238 /
239
240 -- Add keys for table Magazynierzy
241
242 ALTER TABLE Magazynierzy ADD CONSTRAINT Unique_Identifier8
243 PRIMARY KEY (Nr_pracownika)
244 /
245
246 -- Table Magazynierzy_Magazyny
247
248 CREATE TABLE Magazynierzy_Magazyny(
249     Nr_magazynu Integer NOT NULL,
250     Nr_pracownika Integer NOT NULL
251 )
252 /
253
254 -- Table Adresy
255
256 CREATE TABLE Adresy(
257     Nr_adresu Integer NOT NULL,
258     Miasto Varchar2(30 ) NOT NULL,
259     Ulica Varchar2(30 ) NOT NULL,
260     Nr_lokalu Varchar2(5 ) NOT NULL,
261     Kod_pocztowy Char(6 ) NOT NULL,
262     Kraj Varchar2(30 ) NOT NULL
263 )
264 /

```

```

265
266 -- Add keys for table Adresy
267
268 ALTER TABLE Adresy ADD CONSTRAINT PK_Adresy PRIMARY KEY (Nr_adresu)
269 /
270
271 -- Table and Columns comments section
272
273 COMMENT ON COLUMN Adresy.Nr_adresu IS 'Unikatowy_numer_adresu'
274 /
275 COMMENT ON COLUMN Adresy.Miasto IS 'Miasto'
276 /
277 COMMENT ON COLUMN Adresy.Ulica IS 'Ulica'
278 /
279 COMMENT ON COLUMN Adresy.Nr_lokalu IS 'Numer_lokalu'
280 /
281 COMMENT ON COLUMN Adresy.Kod_pocztowy IS 'Kod_pocztowy'
282 /
283 COMMENT ON COLUMN Adresy.Kraj IS 'Kraj'
284 /
285
286 -- Table Wlasciciele
287
288 CREATE TABLE Wlasciciele(
289     Nr_wlasciciela Integer NOT NULL,
290     Imie Varchar2(30 ) NOT NULL,
291     Nazwisko Varchar2(30 ) NOT NULL,
292     Nr_portu Integer NOT NULL
293 )
294 /
295
296 -- Add keys for table Wlasciciele
297
298 ALTER TABLE Wlasciciele ADD CONSTRAINT PK_Wlasciciele PRIMARY KEY (Nr_wlasciciela)
299 /
300
301 -- Table and Columns comments section
302
303 COMMENT ON COLUMN Wlasciciele.Nr_wlasciciela IS 'Unikatowy_numer_wlasciciela'
304 /
305 COMMENT ON COLUMN Wlasciciele.Imie IS 'Imie_wlasciciela'
306 /
307 COMMENT ON COLUMN Wlasciciele.Nazwisko IS 'Nazwisko_wlasciciela'
308 /
309
310 -- Table Stanowiska
311
312 CREATE TABLE Stanowiska(
313     Nr_stanowiska Integer NOT NULL,
314     Nazwa Varchar2(30 ) NOT NULL,
315     Opis Varchar2(400 ) NOT NULL
316 )
317 /
318
319 -- Add keys for table Stanowiska
320

```

```

321 ALTER TABLE Stanowiska ADD CONSTRAINT PK_Stalowiska PRIMARY KEY (Nr_stalowiska)
322 /
323
324 ALTER TABLE Stanowiska ADD CONSTRAINT Nazwa UNIQUE (Nazwa)
325 /
326
327 -- Table and Columns comments section
328
329 COMMENT ON COLUMN Stanowiska.Nr_stalowiska IS 'Unikatowy_numer_stalowiska'
330 /
331 COMMENT ON COLUMN Stanowiska.Nazwa IS 'Nazwa_stalowiska'
332 /
333 COMMENT ON COLUMN Stanowiska.Opis IS 'Opis_stalowiska'
334 /
335
336 -- Table Wynagrodzenia
337
338 CREATE TABLE Wynagrodzenia(
339     Nr_wynagrodzenia Integer NOT NULL,
340     Data Date NOT NULL,
341     Kwota_podstawowa Number(8,2) NOT NULL,
342     Kwota_dodatkowa Number(7,2),
343     Nr_pracownika Integer NOT NULL
344 )
345 /
346
347
348 -- Add keys for table Wynagrodzenia
349
350 ALTER TABLE Wynagrodzenia ADD CONSTRAINT PK_Wynagrodzenia
351 PRIMARY KEY (Nr_wynagrodzenia)
352 /
353
354 -- Table and Columns comments section
355
356 COMMENT ON COLUMN Wynagrodzenia.Nr_wynagrodzenia IS 'Unikatowy_numer_wynagrodzenia'
357 /
358 COMMENT ON COLUMN Wynagrodzenia.Data IS 'Data_wyplaty_wynagrodzenia'
359 /
360 COMMENT ON COLUMN Wynagrodzenia.Kwota_podstawowa IS 'Kwota_podstawowa_wynagordzenia'
361 /
362 COMMENT ON COLUMN Wynagrodzenia.Kwota_dodatkowa IS 'Kwota_dodatkowa'
363 /
364
365 -- Table Zlecenia
366
367 CREATE TABLE Zlecenia(
368     Nr_zlecenia Integer NOT NULL,
369     Cena Integer NOT NULL,
370     Data_przyjecia_zlecenia Date NOT NULL,
371     Status Varchar2(12 ) NOT NULL
372         CHECK (Status IN ('Aktywne', 'Zrealizowane')),
373     Data_dostawy Date NOT NULL,
374     Data_odbioru Date NOT NULL,
375     Opis Varchar2(1200 ),
376     Nr_klienta Integer NOT NULL

```

```

377 )
378 /
379
380 -- Add keys for table Zlecenia
381
382 ALTER TABLE Zlecenia ADD CONSTRAINT PK_Zlecenie PRIMARY KEY (Nr_zlecenia)
383 /
384
385 -- Table and Columns comments section
386
387 COMMENT ON COLUMN Zlecenia.Opis IS 'Opis_zlecenia'
388 /
389
390 -- Table Wozki_widlowe
391
392 CREATE TABLE Wozki_widlowe(
393     Nr_wozka Integer NOT NULL,
394     Nr_fabryczny Varchar2(30 ) NOT NULL,
395     Rok_produkcji Integer NOT NULL,
396     Udzwig Integer NOT NULL,
397     Wysokosc_podnoszenia Integer NOT NULL,
398     Nr_modelu Integer NOT NULL,
399     Nr_portu Integer NOT NULL,
400     Nr_magazynu Integer
401 )
402 /
403
404 -- Add keys for table Wozki_widlowe
405
406 ALTER TABLE Wozki_widlowe ADD CONSTRAINT PK_Wozki_widlowe PRIMARY KEY (Nr_wozka)
407 /
408
409 ALTER TABLE Wozki_widlowe ADD CONSTRAINT AK_Wozki_widlowe UNIQUE (Nr_fabryczny)
410 /
411
412 -- Table and Columns comments section
413
414 COMMENT ON COLUMN Wozki_widlowe.Nr_wozka IS 'Unikatowy_numer_wozka_widlowego'
415 /
416 COMMENT ON COLUMN Wozki_widlowe.Nr_fabryczny IS 'Numer_fabryczny_wozka_widlowego'
417 /
418 COMMENT ON COLUMN Wozki_widlowe.Udzwig IS 'Udzwig_wozka_widlowego_w_kilogramach'
419 /
420 COMMENT ON COLUMN Wozki_widlowe.Wysokosc_podnoszenia IS
421 'Wysokosc_podnoszenia_wozka_widlowego'
422 /
423
424 -- Table Marki
425
426 CREATE TABLE Marki(
427     Nr_marki Integer NOT NULL,
428     Nazwa_marki Varchar2(35 ) NOT NULL,
429     Opis Varchar2(400 )
430 )
431 /
432

```

```

433 -- Add keys for table Marki
434
435 ALTER TABLE Marki ADD CONSTRAINT PK_Marki PRIMARY KEY (Nr_marki)
436 /
437
438 ALTER TABLE Marki ADD CONSTRAINT AK_Marki UNIQUE (Nazwa_marki)
439 /
440
441 -- Table and Columns comments section
442
443 COMMENT ON COLUMN Marki.Nr_marki IS 'Unikatowy_numer_marki_wozka_widlowego'
444 /
445 COMMENT ON COLUMN Marki.Opis IS 'Opis_marki_wozka_widlowego'
446 /
447
448 -- Table Modele
449
450 CREATE TABLE Modele(
451     Nr_modelu Integer NOT NULL,
452     Nazwa_modelu Varchar2(50 ) NOT NULL,
453     Opis Varchar2(400 ),
454     Nr_marki Integer NOT NULL
455 )
456 /
457
458 -- Add keys for table Modele
459
460 ALTER TABLE Modele ADD CONSTRAINT PK_Modele PRIMARY KEY (Nr_modelu)
461 /
462
463 -- Table and Columns comments section
464
465 COMMENT ON COLUMN Modele.Nr_modelu IS 'Unikatowy_numer_modelu_wozka_widlowego'
466 /
467 COMMENT ON COLUMN Modele.Opis IS 'Opis_modelu_wozka_widlowego'
468 /
469
470 -- Table Skladowania
471
472 CREATE TABLE Skladowania(
473     Data_poczatku Date NOT NULL,
474     Nr_ladunku Integer NOT NULL,
475     Nr_miejsca Integer NOT NULL,
476     Data_konca Date NOT NULL
477 )
478 /
479
480 -- Add keys for table Skladowania
481
482 ALTER TABLE Skladowania ADD CONSTRAINT PK_Skladowania PRIMARY KEY
483 (Nr_ladunku,Nr_miejsca,Data_poczatku)
484 /
485
486 -- Table and Columns comments section
487
488 COMMENT ON COLUMN Skladowania.Data_poczatku IS 'Data_poczatku_skladowania_ladunku.'

```

```

489 /
490 COMMENT ON COLUMN Skladowania.Data_konca IS 'Data_konca_skladowania_ladunku.'
491 /
492
493 -- Table Miejsca_sklad
494
495 CREATE TABLE Miejsca_sklad(
496     Nr_miejsca Integer NOT NULL,
497     Nr_regalu Integer NOT NULL,
498     Nr_polki Integer NOT NULL,
499     Wysokosc Integer NOT NULL,
500     Nr_magazynu Integer NOT NULL
501 )
502 /
503
504 -- Add keys for table Miejsca_sklad
505
506 ALTER TABLE Miejsca_sklad ADD CONSTRAINT PK_Miejsca_sklad PRIMARY KEY (Nr_miejsca)
507 /
508
509 -- Table and Columns comments section
510
511 COMMENT ON COLUMN Miejsca_sklad.Nr_miejsca IS
512 'Unikatowy_numer_miejsca_przechowywania_ladunku.'
513 /
514 COMMENT ON COLUMN Miejsca_sklad.Nr_regalu IS 'Numer_regalu.'
515 /
516 COMMENT ON COLUMN Miejsca_sklad.Nr_polki IS 'Numer_polki.'
517 /
518 COMMENT ON COLUMN Miejsca_sklad.Wysokosc IS
519 'Wysokosc_nad_ziemia_miejsca_skladowania(w_metrach).'
520 /
521
522 -- Trigger for sequence PortSeq1 for column Nr_portu in table Porty -----
523 CREATE OR REPLACE TRIGGER ts_Porty_PortSeq1 BEFORE INSERT
524 ON Porty FOR EACH ROW
525 BEGIN
526     :new.Nr_portu := PortSeq1.nextval;
527 END;
528 /
529 CREATE OR REPLACE TRIGGER tsu_Porty_PortSeq1 AFTER UPDATE OF Nr_portu
530 ON Porty FOR EACH ROW
531 BEGIN
532     RAISE_APPLICATION_ERROR(-20010,'Cannot_update_column_Nr_portu_in_table
533 Porty_as_it_uses_sequence.');
```



```

545 ON Magazyny FOR EACH ROW
546 BEGIN
547     RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_magazynu in table
548     Magazyny as it uses sequence. ');
549 END;
550 /
551
552 -- Trigger for sequence LadunekSeq1 for column Nr_ladunku in table Ladunki -----
553 CREATE OR REPLACE TRIGGER ts_Ladunki_LadunekSeq1 BEFORE INSERT
554 ON Ladunki FOR EACH ROW
555 BEGIN
556     :new.Nr_ladunku := LadunekSeq1.nextval;
557 END;
558 /
559 CREATE OR REPLACE TRIGGER tsu_Ladunki_LadunekSeq1 AFTER UPDATE OF Nr_ladunku
560 ON Ladunki FOR EACH ROW
561 BEGIN
562     RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_ladunku in table
563     Ladunki as it uses sequence. ');
564 END;
565 /
566
567 -- Trigger for sequence KlientSeq1 for column Nr_klienta in table Klienci -----
568 CREATE OR REPLACE TRIGGER ts_Klienci_KlientSeq1 BEFORE INSERT
569 ON Klienci FOR EACH ROW
570 BEGIN
571     :new.Nr_klienta := KlientSeq1.nextval;
572 END;
573 /
574 CREATE OR REPLACE TRIGGER tsu_Klienci_KlientSeq1 AFTER UPDATE OF Nr_klienta
575 ON Klienci FOR EACH ROW
576 BEGIN
577     RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_klienta in table
578     Klienci as it uses sequence. ');
579 END;
580 /
581
582 -- Trigger for sequence PracownikSeq1 for column Nr_pracownika in table Pracownicy ---
583 CREATE OR REPLACE TRIGGER ts_Pracownicy_PracownikSeq1 BEFORE INSERT
584 ON Pracownicy FOR EACH ROW
585 BEGIN
586     :new.Nr_pracownika := PracownikSeq1.nextval;
587 END;
588 /
589 CREATE OR REPLACE TRIGGER tsu_Pracownicy_PracownikSeq1 AFTER UPDATE OF Nr_pracownika
590 ON Pracownicy FOR EACH ROW
591 BEGIN
592     RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_pracownika in table
593     Pracownicy as it uses sequence. ');
594 END;
595 /
596
597 -- Trigger for sequence AdresSeq1 for column Nr_adresu in table Adresy -----
598 CREATE OR REPLACE TRIGGER ts_Adresy_AdresSeq1 BEFORE INSERT
599 ON Adresy FOR EACH ROW
600 BEGIN

```

```

601      :new.Nr_adresu := AdresSeq1.nextval;
602 END;
603 /
604 CREATE OR REPLACE TRIGGER tsu_Adresy_AdresSeq1 AFTER UPDATE OF Nr_adresu
605 ON Adresy FOR EACH ROW
606 BEGIN
607     RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_adresu in table
608 Adresy as it uses sequence. ');
609 END;
610 /
611
612 -- Trigger for sequence WlascicielSeq1 for column Nr_wlasciciela in table Wlasciciele
613 CREATE OR REPLACE TRIGGER ts_Wlasciciele_WlascicielSeq1 BEFORE INSERT
614 ON Wlasciciele FOR EACH ROW
615 BEGIN
616     :new.Nr_wlasciciela := WlascicielSeq1.nextval;
617 END;
618 /
619 CREATE OR REPLACE TRIGGER tsu_Wlasciciele_WlascicielSeq1 AFTER UPDATE OF
620 Nr_wlasciciela
621 ON Wlasciciele FOR EACH ROW
622 BEGIN
623     RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_wlasciciela in table
624 Wlasciciele as it uses sequence. ');
625 END;
626 /
627
628 -- Trigger for sequence StanowiskoSeq1 for column Nr_stanowiska in table Stanowiska -
629 CREATE OR REPLACE TRIGGER ts_Stalowiska_StalowiskoSeq1 BEFORE INSERT
630 ON Stanowiska FOR EACH ROW
631 BEGIN
632     :new.Nr_stalowiska := StanowiskoSeq1.nextval;
633 END;
634 /
635 CREATE OR REPLACE TRIGGER tsu_Stalowiska_StalowiskoSeq1 AFTER UPDATE OF Nr_stalowiska
636 ON Stanowiska FOR EACH ROW
637 BEGIN
638     RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_stalowiska in table
639 Stanowiska as it uses sequence. ');
640 END;
641 /
642
643 -- Trigger for sequence WynagSeq1 for column Nr_wynagrodzenia in table Wynagrodzenia -
644 CREATE OR REPLACE TRIGGER ts_Wynagrodzenia_WynagSeq1 BEFORE INSERT
645 ON Wynagrodzenia FOR EACH ROW
646 BEGIN
647     :new.Nr_wynagrodzenia := WynagSeq1.nextval;
648 END;
649 /
650 CREATE OR REPLACE TRIGGER tsu_Wynagrodzenia_WynagSeq1 AFTER UPDATE OF
651 Nr_wynagrodzenia
652 ON Wynagrodzenia FOR EACH ROW
653 BEGIN
654     RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_wynagrodzenia in table
655 Wynagrodzenia as it uses sequence. ');
656 END;

```

```

657 /
658
659 -- Trigger for sequence ZlecenieSeq1 for column Nr_zlecenia in table Zlecenia -----
660 CREATE OR REPLACE TRIGGER ts_Zlecenia_ZlecenieSeq1 BEFORE INSERT
661 ON Zlecenia FOR EACH ROW
662 BEGIN
663     :new.Nr_zlecenia := ZlecenieSeq1.nextval;
664 END;
665 /
666 CREATE OR REPLACE TRIGGER tsu_Zlecenia_ZlecenieSeq1 AFTER UPDATE OF Nr_zlecenia
667 ON Zlecenia FOR EACH ROW
668 BEGIN
669     RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_zlecenia in table
670 Zlecenia as it uses sequence. ');
671 END;
672 /
673
674 -- Trigger for sequence WozekSeq1 for column Nr_wozka in table
675 Wozki_widlowe -----
676 CREATE OR REPLACE TRIGGER ts_Wozki_widlowe_WozekSeq1 BEFORE INSERT
677 ON Wozki_widlowe FOR EACH ROW
678 BEGIN
679     :new.Nr_wozka := WozekSeq1.nextval;
680 END;
681 /
682 CREATE OR REPLACE TRIGGER tsu_Wozki_widlowe_WozekSeq1 AFTER UPDATE OF Nr_wozka
683 ON Wozki_widlowe FOR EACH ROW
684 BEGIN
685     RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_wozka in table
686 Wozki_widlowe as it uses sequence. ');
687 END;
688 /
689
690 -- Trigger for sequence MarkaSeq1 for column Nr_marki in table Marki -----
691 CREATE OR REPLACE TRIGGER ts_Marki_MarkaSeq1 BEFORE INSERT
692 ON Marki FOR EACH ROW
693 BEGIN
694     :new.Nr_marki := MarkaSeq1.nextval;
695 END;
696 /
697 CREATE OR REPLACE TRIGGER tsu_Marki_MarkaSeq1 AFTER UPDATE OF Nr_marki
698 ON Marki FOR EACH ROW
699 BEGIN
700     RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_marki in table
701 Marki as it uses sequence. ');
702 END;
703 /
704
705 -- Trigger for sequence ModelSeq1 for column Nr_modelu in table Modele -----
706 CREATE OR REPLACE TRIGGER ts_Modele_ModelSeq1 BEFORE INSERT
707 ON Modele FOR EACH ROW
708 BEGIN
709     :new.Nr_modelu := ModelSeq1.nextval;
710 END;
711 /
712 CREATE OR REPLACE TRIGGER tsu_Modele_ModelSeq1 AFTER UPDATE OF Nr_modelu

```

```

713 ON Modele FOR EACH ROW
714 BEGIN
715     RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_modelu in table
716 Modele as it uses sequence. ');
717 END;
718 /
719
720 -- Trigger for sequence Msc_sklSeq1 for column Nr_miejsca in table Miejsca_sklad -----
721 CREATE OR REPLACE TRIGGER ts_Miejsca_sklad_Msc_sklSeq1 BEFORE INSERT
722 ON Miejsca_sklad FOR EACH ROW
723 BEGIN
724     :new.Nr_miejsca := Msc_sklSeq1.nextval;
725 END;
726 /
727 CREATE OR REPLACE TRIGGER tsu_Miejsca_sklad_Msc_sklSeq1 AFTER UPDATE OF Nr_miejsca
728 ON Miejsca_sklad FOR EACH ROW
729 BEGIN
730     RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_miejsca in table
731 Miejsca_sklad as it uses sequence. ');
732 END;
733 /
734
735
736 -- Create foreign keys (relationships) section -----
737
738 ALTER TABLE Magazyny ADD CONSTRAINT posiada_magazyn FOREIGN KEY (Nr_portu)
739 REFERENCES Porty (Nr_portu)
740 /
741
742
743
744 ALTER TABLE Klienci ADD CONSTRAINT ma_klienta FOREIGN KEY (Nr_portu)
745 REFERENCES Porty (Nr_portu)
746 /
747
748
749
750 ALTER TABLE Pracownicy ADD CONSTRAINT zatrudnia FOREIGN KEY (Nr_portu)
751 REFERENCES Porty (Nr_portu)
752 /
753
754
755
756 ALTER TABLE Klienci ADD CONSTRAINT adres_klienta FOREIGN KEY (Nr_adresu)
757 REFERENCES Adresy (Nr_adresu)
758 /
759
760
761
762 ALTER TABLE Porty ADD CONSTRAINT adres_portu FOREIGN KEY (Nr_adresu)
763 REFERENCES Adresy (Nr_adresu)
764 /
765
766
767
768 ALTER TABLE Pracownicy ADD CONSTRAINT adres_pracownika FOREIGN KEY (Nr_adresu)

```

```

769 REFERENCES Adresy (Nr_adresu)
770 /
771
772
773
774 ALTER TABLE Wlasciciele ADD CONSTRAINT port_ma_wlasciciela FOREIGN KEY (Nr_portu)
775 REFERENCES Porty (Nr_portu)
776 /
777
778
779
780 ALTER TABLE Pracownicy ADD CONSTRAINT pracownik_ma_stanowisko FOREIGN KEY
781 (Nr_stanowiska) REFERENCES Stanowiska (Nr_stanowiska)
782 /
783
784
785
786 ALTER TABLE Wynagrodzenia ADD CONSTRAINT otrzymuje_wynagrodzenie FOREIGN KEY
787 (Nr_pracownika) REFERENCES Pracownicy (Nr_pracownika)
788 /
789
790
791
792 ALTER TABLE Zlecenia ADD CONSTRAINT ma_zlecenia FOREIGN KEY (Nr_klienta)
793 REFERENCES Klienci (Nr_klienta)
794 /
795
796
797
798 ALTER TABLE Ladunki ADD CONSTRAINT zawiera_ladunek FOREIGN KEY (Nr_zlecenia)
799 REFERENCES Zlecenia (Nr_zlecenia)
800 /
801
802
803
804 ALTER TABLE Modele ADD CONSTRAINT posiada_model FOREIGN KEY (Nr_marki)
805 REFERENCES Marki (Nr_marki)
806 /
807
808
809
810 ALTER TABLE Wozki_widlowe ADD CONSTRAINT wozek_ma_model FOREIGN KEY (Nr_modelu)
811 REFERENCES Modele (Nr_modelu)
812 /
813
814
815
816 ALTER TABLE Wozki_widlowe ADD CONSTRAINT posiada_wozek FOREIGN KEY (Nr_portu)
817 REFERENCES Porty (Nr_portu)
818 /
819
820
821
822 ALTER TABLE Wozki_widlowe ADD CONSTRAINT przynalezy FOREIGN KEY (Nr_magazynu)
823 REFERENCES Magazyny (Nr_magazynu)
824 /

```

```

825
826
827
828 ALTER TABLE Skladowania ADD CONSTRAINT jest_skladowany FOREIGN KEY (Nr_ladunku)
829 REFERENCES Ladunki (Nr_ladunku)
830 /
831
832
833
834 ALTER TABLE Miejsca_sklad ADD CONSTRAINT ma_miejsce_skladowania FOREIGN KEY
835 (Nr_magazynu) REFERENCES Magazyny (Nr_magazynu)
836 /
837
838
839
840 ALTER TABLE Skladowania ADD CONSTRAINT jest_na_miejscu FOREIGN KEY (Nr_miejsca)
841 REFERENCES Miejsca_sklad (Nr_miejsca)
842 /
843
844 --create indexes
845 CREATE INDEX IX_adres_portu ON Porty (Nr_adresu);
846 CREATE INDEX IX_zawiera_ladunek ON Ladunki (Nr_zlecenia);
847 CREATE INDEX IX_ma_miejsce_skladowania ON Miejsca_sklad (Nr_magazynu);
848 CREATE INDEX IX_zatrudnia ON Pracownicy (Nr_portu);
849 CREATE INDEX IX_pracownik_ma_stanowisko ON Pracownicy (Nr_stanowiska);
850 CREATE INDEX IX_ma_klienta ON Klienci (Nr_portu);
851 CREATE INDEX IX_adres_klienta ON Klienci (Nr_adresu);

```

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

5.4.1. Zapełnienie danymi

```

1  /*port*/
2  INSERT INTO Adresy(miasto, ulica, nr_lokalu, kod_pocztowy, kraj)
3  VALUES ('Kutno', 'Oporowska', '7', '99-300', 'Polska');
4  INSERT INTO Porty(nazwa, email, nr_telefonu, nr_adresu)
5  VALUES ('DobryPort', 'portdobry@port.pl', '801-802-803', 1);
6
7  /*wlascciele*/
8  INSERT INTO Wlasciciele(imie, nazwisko, nr_portu) VALUES
9  ('Robert', 'Lewandowski', 1);
10
11 /*klienci*/
12
13 -- adresy klientow
14 INSERT INTO Adresy(miasto, ulica, nr_lokalu, kod_pocztowy, kraj) VALUES
15 ('Bialystok', 'Legionowa', '8', '15-194', 'Polska');
16 INSERT INTO Adresy(miasto, ulica, nr_lokalu, kod_pocztowy, kraj) VALUES
17 ('Plock', 'Czwartakow', '3', '09-400', 'Polska');
18 INSERT INTO Adresy(miasto, ulica, nr_lokalu, kod_pocztowy, kraj) VALUES
19 ('Monachium', 'Hundestrasse', '3', '81-675', 'Niemcy');
20
21 -- klienci
22 INSERT INTO Klienci(nazwa_firmy, nr_telefonu, email, nr_konta, nr_portu, nr_adresu)
23 VALUES ('Leczko_jest_pyszne', '505-876-045', 'leczojestpyszne@leczo.pl',
24 '63179490636015478653657360', 1, 2);

```

```

25 INSERT INTO Klienci(nazwa_firmy, nr_telefonu, email, nr_konta, nr_portu, nr_adresu)
26 VALUES ('Szybkieauta', '575-235-067', 'szybkieauta@bardzoszybkie.pl',
27 '72532204635141553531864043', 1, 3);
28 INSERT INTO Klienci(nazwa_firmy, nr_telefonu, email, nr_konta, nr_portu, nr_adresu)
29 VALUES ('ObstundGemuse', '+49894-131-70', 'meinegemuse@obst.de',
30 '24648126919520194795208564', 1, 4);
31
32 /*pracownicy*/
33
34 --stanowiska
35 INSERT INTO Stanowiska(Nazwa, Opis) VALUES ('Magazynier',
36 'Zajmujesie transportem ladunkow w obrebie portu. ');
37 INSERT INTO Stanowiska(Nazwa, Opis) VALUES ('Pracownikbiurowy',
38 'Zajmujesie tabelkami w Excelu oraz kontaktem z klientami. ');
39
40 -- adresy pracownikow
41 INSERT INTO Adresy(miasto, ulica, nr_lokalu, kod_pocztowy, kraj) VALUES
42 ('Pruszkow', 'Wysockiego', '4', '05-643', 'Polska');
43 INSERT INTO Adresy(miasto, ulica, nr_lokalu, kod_pocztowy, kraj) VALUES
44 ('Kutno', 'ArmiiCzerwonej', '98', '99-301', 'Polska');
45
46 -- pracownicy
47 INSERT INTO Pracownicy(Imie, Nazwisko, Plec, Nr_telefonu, Email, PESEL,
48 Data_urodzenia, Data_zatrudnienia, Nr_konta, Nr_portu, Nr_adresu, Nr_stanowiska)
49 VALUES ('Andrzej', 'Marian', 'M', '603-136-747', 'andrzejekkk96@onet.pl',
50 '96020356463', TO_DATE('1996/03/02', 'yyyy/mm/dd'),
51 TO_DATE('2014/05/02', 'yyyy/mm/dd'), '90039309522289322135779473', 1, 5, 1);
52
53 INSERT INTO Pracownicy(Imie, Nazwisko, Plec, Nr_telefonu, Email, PESEL,
54 Data_urodzenia, Data_zatrudnienia, Nr_konta, Nr_portu, Nr_adresu, Nr_stanowiska)
55 VALUES ('Beata', 'Nowak', 'K', '614-569-947', 'beata.nowak@wp.pl',
56 '91072098364', TO_DATE('1991/07/20', 'yyyy/mm/dd'),
57 TO_DATE('2016/04/01', 'yyyy/mm/dd'), '66443017694170038232221516', 1, 6, 2);
58
59 -- specjalizacje
60 INSERT INTO Magazynierzy(Nr_pracownika, Data_badania, Data_uprawnien)
61 VALUES (1, TO_DATE('2021/03/02', 'yyyy/mm/dd'), TO_DATE('2030/03/02', 'yyyy/mm/dd'));
62
63 -- wynagrodzenia
64 INSERT INTO Wynagrodzenia(data, kwota_podstawowa, kwota_dodatkowa, nr_pracownika)
65 VALUES (TO_DATE('28', 'dd'), '4200', '500', 1);
66 INSERT INTO Wynagrodzenia(data, kwota_podstawowa, kwota_dodatkowa, nr_pracownika)
67 VALUES (TO_DATE('27', 'dd'), '3400', NULL, 2);
68
69 -- magazyny
70 INSERT INTO Magazyny(pojemnosc, nr_telefonu, nr_portu) VALUES
71 ('600', '406-245-134', 1);
72 INSERT INTO Magazyny(pojemnosc, nr_telefonu, nr_portu) VALUES
73 ('5400', '406-215-118', 1);
74
75 -- marki wozkow
76 INSERT INTO Marki(nazwa_marki) VALUES('SolidHub');
77 INSERT INTO Marki(nazwa_marki) VALUES('Toyota');
78
79 -- modele wozkow
80 INSERT INTO Modele(nazwa_modelu, nr_marki) VALUES ('GSE2500/5', 1);

```

```

81 INSERT INTO Modele(nazwa_modelu, nr_marki) VALUES ('8FGF25',2);
82 INSERT INTO Modele(nazwa_modelu, nr_marki) VALUES ('8FGF18',2);
83
84 -- wozki widlowe
85 INSERT INTO Wozki_widlowe(nr_fabryczny, rok_produkcji, udzwig,
86 wysokosc_podnoszenia, nr_modelu, nr_portu, nr_magazynu)
87 VALUES('A21F6D51ADG65', '2019', '2500', '6', 1, 1, 1);
88
89 INSERT INTO Wozki_widlowe(nr_fabryczny, rok_produkcji, udzwig,
90 wysokosc_podnoszenia, nr_modelu, nr_portu, nr_magazynu)
91 VALUES('H4FJ1GF4J1G8G', '2016', '3000', '5', 2, 1, 2);
92
93 INSERT INTO Wozki_widlowe(nr_fabryczny, rok_produkcji, udzwig,
94 wysokosc_podnoszenia, nr_modelu, nr_portu, nr_magazynu)
95 VALUES('4JK1H1D1G1H4J', '2020', '1750', '3', 3, 1, NULL);
96
97 -- zlecenia
98 INSERT INTO Zlecenia(cena, data_przyjecia_zlecenia, status, data_dostawy,
99 data_odbioru, opis, nr_klienta)
100 VALUES ('11900', TO_DATE('2020/10/18', 'yyyy/mm/dd'), 'Zrealizowane',
101 TO_DATE('2020/10/21', 'yyyy/mm/dd'), TO_DATE('2020/11/24', 'yyyy/mm/dd'),
102 'Kielbaska', 1);
103
104 INSERT INTO Zlecenia(cena, data_przyjecia_zlecenia, status, data_dostawy,
105 data_odbioru, opis, nr_klienta)
106 VALUES ('24000', TO_DATE('2020/10/24', 'yyyy/mm/dd'), 'Zrealizowane',
107 TO_DATE('2020/10/29', 'yyyy/mm/dd'), TO_DATE('2020/11/04', 'yyyy/mm/dd'),
108 'Cegly i inne materialy budowlane', 1);
109
110 INSERT INTO Zlecenia(cena, data_przyjecia_zlecenia, status, data_dostawy,
111 data_odbioru, opis, nr_klienta)
112 VALUES ('9000', TO_DATE('2020/11/09', 'yyyy/mm/dd'), 'Aktywne',
113 TO_DATE('2020/11/26', 'yyyy/mm/dd'), TO_DATE('2020/11/30', 'yyyy/mm/dd'),
114 'Czesci samochodowe', 2);
115
116 INSERT INTO Zlecenia(cena, data_przyjecia_zlecenia, status, data_dostawy,
117 data_odbioru, opis, nr_klienta)
118 VALUES ('4600', TO_DATE('2020/11/25', 'yyyy/mm/dd'), 'Aktywne',
119 TO_DATE('2020/12/01', 'yyyy/mm/dd'), TO_DATE('2020/12/03', 'yyyy/mm/dd'),
120 'Owoce i warzywa od jakiegos Niemca', 3);
121
122 -- ladunki
123 INSERT INTO Ladunki(Objetosc, Masa, Nr_zlecenia) VALUES ('12', '9000', 1);
124 INSERT INTO Ladunki(Objetosc, Masa, Nr_zlecenia) VALUES ('38', '42000', 2);
125 INSERT INTO Ladunki(Objetosc, Masa, Nr_zlecenia) VALUES ('10', '5400', 3);
126 INSERT INTO Ladunki(Objetosc, Masa, Nr_zlecenia) VALUES ('4', '1800', 4);
127 INSERT INTO Ladunki(Objetosc, Masa, Nr_zlecenia) VALUES ('5', '173', 4);
128
129 -- miejsca_sklad
130 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
131 VALUES ('1', '1', '0', 1);
132 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
133 VALUES ('1', '2', '2', 1);
134 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
135 VALUES ('1', '3', '4', 1);
136 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)

```



```

137 VALUES ('1','4','6',1);
138 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
139 VALUES ('2','1','0',1);
140 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
141 VALUES ('2','2','2',1);
142 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
143 VALUES ('2','3','4',1);
144 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
145 VALUES ('2','4','6',1);
146
147 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
148 VALUES ('1','1','0',2);
149 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
150 VALUES ('1','2','2',2);
151 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
152 VALUES ('1','3','4',2);
153 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
154 VALUES ('1','4','6',2);
155 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
156 VALUES ('2','1','0',2);
157 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
158 VALUES ('2','2','2',2);
159 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
160 VALUES ('2','3','4',2);
161 INSERT INTO Miejsca_sklad(nr_regalu, nr_polki, wysokosc, nr_magazynu)
162 VALUES ('2','4','6',2);
163
164 -- skladowania
165 INSERT INTO Skladowania(data_poczatku, nr_ladunku, nr_miejsca, data_konca) VALUES
166 (TO_DATE('2020/10/21', 'yyyy/mm/dd'), 1, 1, TO_DATE('2020/10/24', 'yyyy/mm/dd'));
167 INSERT INTO Skladowania(data_poczatku, nr_ladunku, nr_miejsca, data_konca) VALUES
168 (TO_DATE('2020/10/21', 'yyyy/mm/dd'), 1, 2, TO_DATE('2020/10/24', 'yyyy/mm/dd'));
169 INSERT INTO Skladowania(data_poczatku, nr_ladunku, nr_miejsca, data_konca) VALUES
170 (TO_DATE('2020/10/21', 'yyyy/mm/dd'), 1, 3, TO_DATE('2020/10/24', 'yyyy/mm/dd'));
171 INSERT INTO Skladowania(data_poczatku, nr_ladunku, nr_miejsca, data_konca) VALUES
172 (TO_DATE('2020/10/21', 'yyyy/mm/dd'), 1, 4, TO_DATE('2020/10/24', 'yyyy/mm/dd'));
173
174 INSERT INTO Skladowania(data_poczatku, nr_ladunku, nr_miejsca, data_konca) VALUES
175 (TO_DATE('2020/10/29', 'yyyy/mm/dd'), 2, 8, TO_DATE('2020/11/04', 'yyyy/mm/dd'));
176 INSERT INTO Skladowania(data_poczatku, nr_ladunku, nr_miejsca, data_konca) VALUES
177 (TO_DATE('2020/10/29', 'yyyy/mm/dd'), 2, 9, TO_DATE('2020/11/04', 'yyyy/mm/dd'));
178 INSERT INTO Skladowania(data_poczatku, nr_ladunku, nr_miejsca, data_konca) VALUES
179 (TO_DATE('2020/10/29', 'yyyy/mm/dd'), 2, 10, TO_DATE('2020/11/04', 'yyyy/mm/dd'));
180 INSERT INTO Skladowania(data_poczatku, nr_ladunku, nr_miejsca, data_konca) VALUES
181 (TO_DATE('2020/10/29', 'yyyy/mm/dd'), 2, 11, TO_DATE('2020/11/04', 'yyyy/mm/dd'));
182 INSERT INTO Skladowania(data_poczatku, nr_ladunku, nr_miejsca, data_konca) VALUES
183 (TO_DATE('2020/10/29', 'yyyy/mm/dd'), 2, 12, TO_DATE('2020/11/04', 'yyyy/mm/dd'));
184
185 INSERT INTO Skladowania(data_poczatku, nr_ladunku, nr_miejsca, data_konca) VALUES
186 (TO_DATE('2020/11/26', 'yyyy/mm/dd'), 3, 2, TO_DATE('2020/11/30', 'yyyy/mm/dd'));
187 INSERT INTO Skladowania(data_poczatku, nr_ladunku, nr_miejsca, data_konca) VALUES
188 (TO_DATE('2020/11/26', 'yyyy/mm/dd'), 3, 3, TO_DATE('2020/11/30', 'yyyy/mm/dd'));
189
190 INSERT INTO Skladowania(data_poczatku, nr_ladunku, nr_miejsca, data_konca) VALUES
191 (TO_DATE('2020/12/01', 'yyyy/mm/dd'), 4, 15, TO_DATE('2020/12/03', 'yyyy/mm/dd'));
192

```

193 INSERT INTO Magazynierzy_Magazynu(nr_magazynu, nr_pracownika) VALUES (1,1);

5.4.2. Przykładowe zapytania

```
--masa i objetosc ladunkow ktore zostana dostarczone w przyszlosci, wraz z data dostarczenia, nazwa firmy od zleceniam, oraz opisem zlecenia do ktorego naleza
SELECT l.objetosc as Objetosc_Ladunku, l.masa as Masa_Ladunku, k.nazwa_firmy, z.nr_zlecenia, data_dostawy, opis FROM zlecenia z, klienci k, ladunki l
WHERE (data_dostawy > (SELECT current_date FROM dual) and z.nr_klienta = k.nr_klienta and l.nr_zlecenia = z.nr_zlecenia)
```

OBJETOSC_LADUNKU	MASA_LADUNKU	NAZWA_FIRMY	NR_ZLECENIA	DATA_DOSTAWY	OPIS
1	4	1800 Obst und Gemuse	420/12/01	Owoce i warzywa od jakiegos Niemca	
2	5	173 Obst und Gemuse	420/12/01	Owoce i warzywa od jakiegos Niemca	

Masa i objętość ładunków które zostaną dostarczone w przyszłości, wraz z data dostarczenia, nazwa firmy od zlecenia, oraz opisem zlecenia do którego należą

```
--pokaz konkretne dane dotyczace posiadanych wozkow widlowych
SELECT nr_modelu, nazwa_marki, nazwa_modelu, nr_fabryczny, udzwig, wysokosc_podnoszenia, nr_magazynu FROM marki INNER JOIN modele USING (nr_marki)
INNER JOIN wozki_widlowe USING (nr_modelu);
```

NR_MODELU	NAZWA_MARKI	NAZWA_MODELU	NR_FABRYCZNY	UDZWIG	WYSOKOSC_PODNOSZENIA	NR_MAGAZYNU
1	1 SolidHub	GSE2500/5	A21F6D51ADG65	2500	6	1
2	2 Toyota	8FGF25	H4FJ1GF4J1G8G	3000	5	2
3	3 Toyota	8FGF18	4JK1H1D1G1H4J	1750	3	(null)

Pokaż wskazane dane dotyczące posiadanych wózków widłowych

```
--pokaz wozki widlowe jeszcze nie przypisane do zadnego magazynu
SELECT nr_fabryczny, rok_produkcji, udzwig, wysokosc_podnoszenia, nr_portu FROM wozki_widlowe WHERE nr_magazynu is null;
```

NR_FABRYCZNY	ROK_PRODUKCJI	UDZWIG	WYSOKOSC_PODNOSZENIA	NR_PORTU
1 4JK1H1D1G1H4J	2020	1750	3	1

Pokaż wózki widłowe które nie zostały jeszcze przypisane do żadnego magazynu

```
--co komu i do kiedy trzeba wyplacic
SELECT imie, nazwisko, PESEL, nazwa AS nazwa_stanowiska, nr_konta, kwota_podstawowa, kwota_dodatkowa, data FROM wynagrodzenia INNER JOIN pracownicy USING (nr_pracownika)
INNER JOIN stanowiska USING (nr_stanowiska);
```

Script Output x Query Result x Query Result 1 x

SQL | All Rows Fetched: 2 in 0,016 seconds

IMIE	NAZWISKO	PESEL	NAZWA_STANOWISKA	NR_KONTA	KWOTA_PODSTAWOWA	KWOTA_DODATKOWA	DATA
1	Andrzej	Marian	96020356463	Magazynier	90039309522289322135779473	4200	500 20/11/28
2	Beata	Nowak	91072098364	Pracownik biurowy	66443017694170038232221516	3400	(null) 20/11/27

Co, komu i do kiedy trzeba zapłacić

```
-- Wyświetlanie informacji o klientach oraz ich zamówieniach sortując po łącznej kwocie zamówien
SELECT Nazwa_firmy, Miasto, Ulica, Kraj, sum(cena) as Laczna_wartosc_zleceń, avg(objetosc) as Srednia_objetosc_zamowienia FROM klienci, adresy, zlecenia, ladunki
WHERE (klienci.nr_adresu=adresy.nr_adresu AND zlecenia.nr_klienta=klienci.nr_klienta AND zlecenia.nr_zlecenia=ladunki.nr_zlecenia)
GROUP BY Nazwa_firmy, Miasto, Ulica, Kraj
ORDER BY sum(cena) DESC
```

NAZWA_FIRMY	MIASTO	ULICA	KRAJ	LACZNA_WARTOSC_ZLECEN	SREDNIA_OBJETOSC_ZAMOWIENIA
1 Lecsko jest pyszne	Białystok	Legionowa	Polska	35900	25
2 Szybkie auta	Płock	Czwartaków	Polska	9000	10
3 Obst und Gemuse	Monachium	Hundestrasse	Niemcy	4600	4

Wyświetlanie informacji o ładunkach, które są aktualnie składowane w magazynach portu

Worksheet

Query Builder

-- wyswietl informacje i lokalizacje ladunkow ktore sa aktualnie skladowane w magazynach portu
SELECT nr_magazynu, nr_regalu, nr_polki, l.nr_ladunku, l.nr_zlecenia, opis
FROM skladowania s, miejsca_sklad m, ladunki l, zlecenia z
WHERE (s.nr_miejsca=m.nr_miejsca AND data_konca>(SELECT current_date FROM dual) AND data_poczatku<(SELECT current_date FROM dual) AND s.nr_ladunku=l.nr_ladunku AND z.nr_zlecenia=l.nr_zlecenia);

Script Output x

Query Result x

All Rows Fetched: 2 in 0.074 seconds

	NR_MAGAZYN	NR_REGALU	NR_POLKI	NR_LADUNKU	NR_ZLECENIA	OPIS
1	1	1	2	3	3	Czesci samochodowe
2	1	1	3	3	3	Czesci samochodowe

Wyświetlanie informacji o klientach oraz ich zamówieniach sortując po łącznej kwocie zamówień

Worksheet

Query Builder

```
-- wyswietlanie zrealizowanych zleceń
SELECT nazwa_firmy, z.nr_zlecenia, data_przyjecia_zlecenia, data_dostawy, data_odbioru, opis, sum(objetosc) as objetosc_ladunkow, sum(masa) as masa_ladunkow, ce
FROM klienci k, zlecenia z, ladunki l
WHERE (z.nr_klienta=k.nr_klienta AND l.nr_zlecenia=z.nr_zlecenia AND status='Zrealizowane')
GROUP BY nazwa_firmy, z.nr_zlecenia, data_przyjecia_zlecenia, data_dostawy, data_odbioru, opis, cena
```

Script Output x

Query Result x

Query Result 1 x

Query Result 2 x

Query Result 3 x


Query Result 4 x

Query Result 5 x

Query Result 6 x

Query Result 7 x

Query Result 8 x

 All Rows Fetched: 2 in 0.056 seconds

	NAZWA_FIRMY	NR_ZLECENIA	DATA_PRZYJECIA_ZLECENIA	DATA_DOSTAWY	DATA_ODBIORU	OPIS	OBJETOSC_LADUNKOW	MASA_LADUNKOW	CENA
1	Leczko jest pyszne	1	18-OCT-20	21-OCT-20	24-NOV-20	Kielbaska	12	9000	11900
2	Leczko jest pyszne	2	24-OCT-20	29-OCT-20	04-NOV-20	Cegly i inne materialy budowlane	38	42000	24000

Wyświetlanie zrealizowanych zleceń

Worksheet		Query Builder									
<pre>-- sprawdzanie srednich pensji dla poszczegolnych stanowisk SELECT nazwa as stanowisko, avg(kwota_podstawowa) as srednia_pensja FROM stanowiska s, pracownicy p, wynagrodzenia w WHERE s.nr_stanowiska = p.nr_stanowiska AND w.nr_pracownika=p.nr_pracownika GROUP BY nazwa</pre>											
<p>Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x</p> <p>All Rows Fetched: 2 in 0.055 seconds</p> <table> <tr> <th></th><th>STANOWISKO</th><th>SREDNIA_PENSJA</th></tr> <tr> <td>1</td><td>Magazynier</td><td>4200</td></tr> <tr> <td>2</td><td>Pracownik biurowy</td><td>3400</td></tr> </table>				STANOWISKO	SREDNIA_PENSJA	1	Magazynier	4200	2	Pracownik biurowy	3400
	STANOWISKO	SREDNIA_PENSJA									
1	Magazynier	4200									
2	Pracownik biurowy	3400									

Sprawdzanie średnich pensji dla poszczególnych stanowisk