



BAZY DANYCH I BIG DATA

SKLEP INTERNETOWY Z URZĄDZENIAMI
ELEKTRONICZNYMI

Bartłomiej Miłkowski 318808

SPIS TREŚCI

- Zakres i cel projektu 2
- Definicja systemu 2
 - Perspektywy użytkowników 2
- Model Konceptualny 3
 - Definicja zbioru encji 3
 - Ustalenie związków między encjami i ich typy 3
 - Opis związków 4
 - Atrybuty i ich dziedziny 5
 - Reguły i założenia biznesowe 14
 - Klucze kandydujące i klucze główne 15
 - Schemat ER na poziomie konceptualnym 15
 - Problem pułapek szczelinowych 16
- Model logiczny 16
 - Charakterystyka modelu relacyjnego 17
 - Analiza właściwości niekompatybilnych z modelem relacyjnym 17
 - Proces normalizacji 18
 - Schemat ER na poziomie modelu logicznego 20
 - Więzy integralności 21
 - Proces denormalizacji – analiza i przykłady 21
- Faza fizyczna 23
 - Projekt transakcji i weryfikacja ich wykonalności 23
 - Strojanie bazy danych – dobór indeksów 25
 - Skrypt SQL 26
 - Przykłady zapytań SQL 45
- Bibliografia 47

Zakres i cel projektu

Celem projektu było stworzenie relacyjnej bazy danych. Projekt składał się z trzech etapów: konceptualnego, relacyjnego i fizycznego. Pierwszy i drugi etap skupiał się na utworzeniu poprawnego schematu ER (diagramu związków encji), usunięciu wszelkich pułapek i normalizacji. Trzeci etap polegał na implementacji fizycznej utworzonej bazy danych.

Realizowany projekt dotyczy obsługi sklepu internetowego z urządzeniami elektronicznymi. Firma zajmuje się internetową sprzedażą produktów takich jak urządzenia mobilne, komputery czy TV i Audio. Zatrudniani są pracownicy na różnych stanowiskach, wśród których można wyróżnić między innymi księgowych i magazynierów. Część z pracowników jest przypisana, i pracuje w placówkach wynajmowanych przez firmę. W skład placówek wchodzi magazyny, w których przechowywane są produkty i punkty obsługi klienta, których zakres obejmuje przyjmowanie reklamacji i słuzenie klientowi w razie pytań odnośnie zamówień. Sklep internetowy ma zawarte umowy z firmami dostawczymi, które oferują transport produktów z magazynów do klientów. Dodatkowo w bazie danych przechowywane są informacje o wszystkich zamówieniach, ich dacie i statusie.

Definicja systemu

Perspektywy użytkowników

Baza sklepu internetowego opiera się na mechanizmie kontroli dostępu, gdzie użytkownicy posiadają różne uprawnienia, a więc dostęp do różnych danych. W systemie wyodrębniono następujące role:

- **Właściciel** – posiada dostęp i możliwość modyfikacji wszystkich danych
- **Księgowy** – ma dostęp do danych pracowników, ich wynagrodzeń, a także do spisu produktów dostępnych w magazynach
- **Magazynier** – posiada dostęp do spisu produktów dostępnych w magazynie, w którym pracuje i możliwość jego modyfikacji
- **Klient** – ma dostęp do listy dostępnych urządzeń, ich atrybutów, natomiast brak uprawnień modyfikacji żadnych danych

Model konceptualny

Definicja zbiorów encji

W projekcie na poziomie konceptualnym zdefiniowano następujące encje:

- **Sklep_Internetowy** – encja zawiera informacje na temat firmy/sklepu.
- **Pracownik** – encja zawiera informacje na temat pracownika sklepu.
- **Ksiegowy** – encja specjalizacji, wprowadza bardziej szczegółowe informacje na temat pracownika, pełniącego rolę księgowego.
- **Magazynier** - encja specjalizacji, wprowadza bardziej szczegółowe informacje na temat pracownika, pełniącego rolę magazyniera.
- **Placowka** – encja zawiera informacje o wynajmowanej przez firmę placówce.
- **Punkt_obsługi** – encja specjalizacji, wprowadza bardziej szczegółowe informacje placówki pełniące rolę punktu obsługi.
- **Magazyn** – encja specjalizacji, wprowadza bardziej szczegółowe informacje placówki pełniące rolę magazynu.
- **Dostawca** – encja zawiera informacje o firmie dostawczej.
- **Produkt** – encja zawiera informacje o produkcie.
- **Klient** – zawiera informacje o kliencie.

Ustalenie związków między encjami i ich typy

Relacja	Nazwa Relacji	Typ związku	Typ uczestnictwa
Sklep_Internetowy - Pracownik	Zatrudnia	1:n	Obowiązkowy - Opcjonalny
Sklep_Internetowy - Produkt	Zaopatruje	1:n	Obowiązkowy - Opcjonalny
Sklep_Internetowy - Dostawca	Korzysta_z_uslug	1:n	Obowiązkowy - Opcjonalny
Sklep_Internetowy - Klient	Posiada_klienta	1:n	Obowiązkowy - Opcjonalny
Sklep_Internetowy - Placowka	Wynajmuje	1:n	Obowiązkowy - Opcjonalny

Sklep_Internetowy - Zamowienie	Realizuje	1:n	Obowiązkowy - Opcjonalny
Placowka - Pracownik	Daje_miejsce_pracy	1:n	Opcjonalny - Obowiązkowy
Placowka - Produkt	Przechowuje	n:m	Obowiązkowy - Opcjonalny
Placowka - Dostawca	Obsluguje	n:m	Obowiązkowy - Opcjonalny
Klient - Zamowienie	Zamawia	n:m	Obowiązkowy - Obowiązkowy
Magazyn - Magazynier	Pracuje_na_magazynie	1:n	Obowiązkowy - Obowiązkowy
Zamowienie - Produkt	Zawiera_produkty	n:m	Opcjonalny - Obowiązkowy

Opis Związków

- **Sklep_Internetowy – Pracownik** – Sklep może zatrudniać wielu pracowników, lub żadnego. Pracownik może przynależeć tylko do jednego sklepu.
- **Sklep_Internetowy - Produkt** – Sklep może mieć wiele produktów, lub żadnego. Produkt przynależy tylko do jednego sklepu.
- **Sklep_Internetowy - Dostawca** – Sklep zawiera umowy z wieloma dostawcami, lub żadnym. Dostawca należy tylko do jednego sklepu.
- **Sklep_Internetowy - Klient** – Sklep ma wielu klientów. Klient kupuje w tylko jednym sklepie.
- **Sklep_Internetowy - Placowka** - Sklep może wynajmować wiele placówek, lub żadnej. Placówka przynależy tylko do jednego sklepu.
- **Sklep_Internetowy - Zamowienie** – Sklep może mieć wiele zamówień, bądź żadnego. Zamówienie może przynależeć tylko do jednego sklepu.
- **Placowka - Pracownik** – Placówka może mieć wielu pracowników, minimum jednego. Pracownik może przynależeć tylko do jednej placówki, nie musi przynależeć do żadnej.
- **Placowka - Produkt** – Placówka może mieć wiele produktów, lub żadnego. Produkt może przynależeć do wielu placówek, minimum jednej.
- **Placowka - Dostawca** – Placówka może obsługiwać wielu dostawców, lub żadnego. Dostawca może być obsługiwanym w wielu placówkach, minimum jednej.

- **Klient - Zamowienie** – Klient może złożyć wiele zamówień, minimum jedno. Zamówienie może przynależeć tylko do jednego klienta.
- **Magazyn - Magazynier** – Magazyn może dawać miejsce pracy wielu magazynierom, minimum jednemu. Magazynier może przynależeć tylko do jednego magazynu.
- **Zamowienie - Produkt** – Zamówienie może zawierać wiele produktów, minimum jeden. Produkt może przynależeć do wielu zamówień, i nie musi do żadnego.

Atrybuty i ich dziedziny

Sklep_Internetowy

Nazwa Atrybutu	Typ	Domena	Obowiązkowy	Opis
Id_sklepu	Integer	-	TAK	Unikalny identyfikator sklepu
Nazwa	VarChar(100)	-	TAK	Nazwa sklepu
Data	Date	-	TAK	Data założenia sklepu
Adres	VarChar(400)	-	TAK	Adres siedziby firmy, Pole segmentowe (miasto, kod pocztowy, ulica, numer domu)
Wlasciciel	VarChar(400)	-	TAK	Właściciel, Pole segmentowe (imię, nazwisko)
Adres_sklepu_www	VarChar(100)	-	TAK	Adres strony internetowej sklepu

Pracownik

Nazwa Atrybutu	Typ	Domena	Obowiązkowy	Opis
Id_pracownika	Integer	-	TAK	Unikalny identyfikator pracownika
Imie	VarChar(20)	-	TAK	Imię pracownika
Drugie_imie	VarChar(20)	-	NIE	Drugie imię pracownika
Nazwisko	VarChar(20)	-	TAK	Nazwisko pracownika
Data_urodzenia	Date	-	TAK	Data urodzenia pracownika
Plec	VarChar(1)	Plec_D ('M', 'K')	TAK	Płeć pracownika (M - mężczyzna, K - kobieta)
PESEL	Character(11)	-	NIE	Numer PESEL pracownika
Adres	VarChar(400)	-	TAK	Adres zamieszkania pracownika, Pole segmentowe (miasto, kod pocztowy, ulica, numer domu)
Nr_telefonu	VarChar(12)	-	TAK	Numer telefonu pracownika
Email	VarChar(100)	-	TAK	Adres Email pracownika

Wykształcenie	VarChar(1)	Wykształcenie_D ('P', 'S', 'Z', 'W')	TAK	Wykształcenie pracownika (P - podstawowe, S - średnie, Z - zasadnicze zawodowe/branżowe, W - wyższe)
Data_zatrudnienia	Date	-	TAK	Data zatrudnienia pracownika
Stanowisko	VarChar(20)	-	TAK	Stanowisko pracownika
Wynagrodzenie	Money	-	TAK	Miesięczne wynagrodzenie pracownika

Placówka

Nazwa Atrybutu	Typ	Domena	Obowiązkowość	Opis
Id_placowki	Integer	-	TAK	Unikalny numer placówki
Adres	VarChar(400)	-	TAK	Adres placówki, Pole segmentowe (miasto, kod pocztowy, ulica, numer domu)
Nr_telefonu	VarChar(12)	-	TAK	Numer telefonu do placówki
Email	VarChar(100)	-	TAK	Adres Email placówki

Dostawca

Nazwa Atrybutu	Typ	Domena	Obowiązkowość	Opis
Id_dostawcy	Integer	-	TAK	Unikalny numer dostawcy
Nazwa	VarChar(100)	-	TAK	Nazwa firmy dostawczej
Adres	VarChar(400)	-	TAK	Adres dostawcy, Pole segmentowe (miasto, kod pocztowy, ulica, numer domu)
Nr_telefonu	VarChar(12)	-	TAK	Numer telefonu dostawcy
Email	VarChar(100)	-	TAK	Email dostawcy

Produkt

Nazwa Atrybutu	Typ	Domena	Obowiązkowość	Opis
Id_produktu	Integer	-	TAK	Unikalny numer produktu
Nazwa	VarChar(100)	-	TAK	Nazwa produktu
Cena	Money	-	TAK	Aktualna cena urządzenia
Specyfikacja	VarChar(1000)	-	TAK	Specyfikacja urządzenia, tekst formatowany o ograniczonej długości
Opis	Text	-	NIE	Opis urządzenia
Ilosc	Integer	-	TAK	Ilość danego produktu

Klient

Nazwa Atrybutu	Typ	Domena	Obowiązkowość	Opis
Id_klienta	Integer	-	TAK	Unikalny identyfikator klienta
Haslo	VarChar(20)	-	TAK	Hasło do konta klienta
Imie	VarChar(20)	-	TAK	Imię klienta
Nazwisko	VarChar(20)	-	TAK	Nazwisko klienta
Email	VarChar(100)	-	TAK	Adres Email klienta
Nr_telefonu	VarChar(12)	-	NIE	Numer telefonu klienta
Adres	VarChar(400)	-	TAK	Adres zamieszkania klienta, Pole segmentowe (miasto, kod pocztowy, ulica, numer domu)
Plec	VarChar(1)	Plec_D ('M', 'K')	NIE	Płeć klienta

Zamowienie

Nazwa Atrybutu	Typ	Domena	Obowiązkowość	Opis
Id_zamowienia	Integer	-	TAK	Unikalny numer zamówienia
Data_zamowienia	Date	-	TAK	Data złożenia zamówienia
Status	VarChar(1)	Status_zamowienia_D ('O', 'P', 'C', 'S', 'D')	TAK	Status zamówienia: O - Zamówione, P - Zapłacone, C - Przygotowane, S- Wysłane, D - Zrealizowane

Ksiegowy

Nazwa Atrybutu	Typ	Domena	Obowiązkowość	Opis
Zaswiadczenie_o_niekaralnosci	VarChar(20)	-	TAK	Unikalny numer zaświadczenia o niekaralności
Certyfikat_ukonczenia_kursu	VarChar(20)	-	TAK	Unikalny numer certyfikatu ukończenia kursu z zakresu finansów

Magazynier

Nazwa Atrybutu	Typ	Domena	Obowiązkowość	Opis
Zas_med_pracy	Integer	-	TAK	Unikalny numer zaświadczenia od lekarza medycyny pracy
Upr_na_wozek_widlowy	VarChar(20)	-	TAK	Numer uprawnienia na wózek widłowy
Prawo jazdy	VarChar(400)	-	TAK	Prawo jazdy, pole segmentowe: numer dokumentu uprawniającego do prowadzenia pojazdów, kategorie: A1, A, B, C, D,

Punkt_obsługi

Nazwa Atrybutu	Typ	Domena	Obowiązkowość	Opis
Nazwa_punktu	VarChar(100)	-	TAK	Nazwa punktu, zawiera nazwę rejonu w jakim się znajduje
Godziny_otwarcia	VarChar(400)	-	TAK	Godziny otwarcia punktu obsługi, Pole segmentowe (godzina otwarcia, godzina zamknięcia)
Ulatwienie_niepełnosprawni	Boolean	-	TAK	Informacja czy punkt odbioru jest dogodny dla osób z niepełnosprawnościami

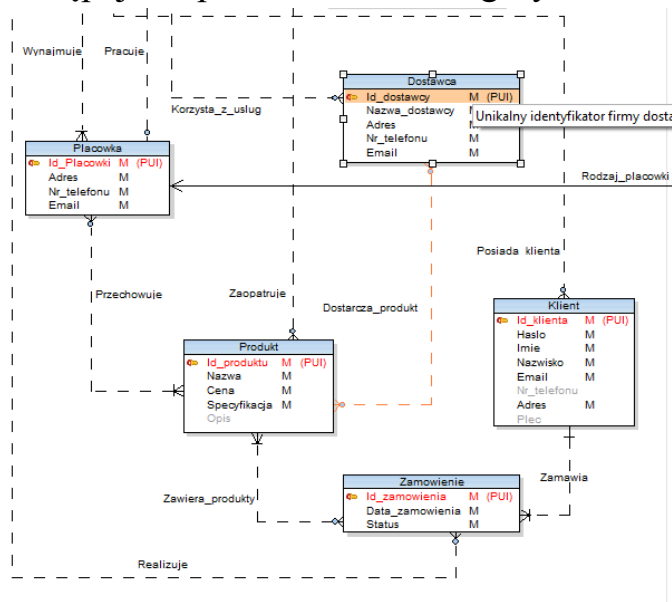
Magazyn

Nazwa Atrybutu	Typ	Domena	Obowiązkowość	Opis
Rodzaj_magazynu	VarChar(1)	Rodzaj_magazynu_D (‘C’, ‘R’, ‘W’)	TAK	Rodzaj magazynu (C - centralny, R - regionalny, W - wysyłkowy)
Powierzchnia	Integer	-	TAK	Powierzchnia magazynu w metrach kwadratowych
Pojazdy	VarChar(400)	-	TAK	Pojazdy, pole segmentowe: nazwa pojazdu, waga pojazdu

Reguły i założenia biznesowe

Aby zachować spójne działanie bazy założono następujące reguły biznesowe:

- W projekcie założono model, w którym klient nie kupuje bezpośrednio produktów, tworzy natomiast zamówienie, do którego dodawane są produkty. Operowanie w takim modelu jest dużo wygodniejsze z perspektywy sklepu, można dzięki temu przechowywać informacje o tym kiedy i przez kogo był dany produkt kupiony.
- Mogą zdarzyć się pracownicy nie-magazynierzy, którzy również będą pracować w magazynie, ale na przykład na stanowisku zarządcy, ochroniarza, księgowego.
- Klient może mieć tylko jeden aktualny adres, na który wysyłane jest zamówienie.
- Wstępnie występowała relacja Dostawca-Produkt, jednak po przeanalizowaniu sytuacji ustalono, że informacja jakie produkty przechowuje dostawca jest zbędna. Związek dostawcy z produktem następuje za pośrednictwem magazynu.

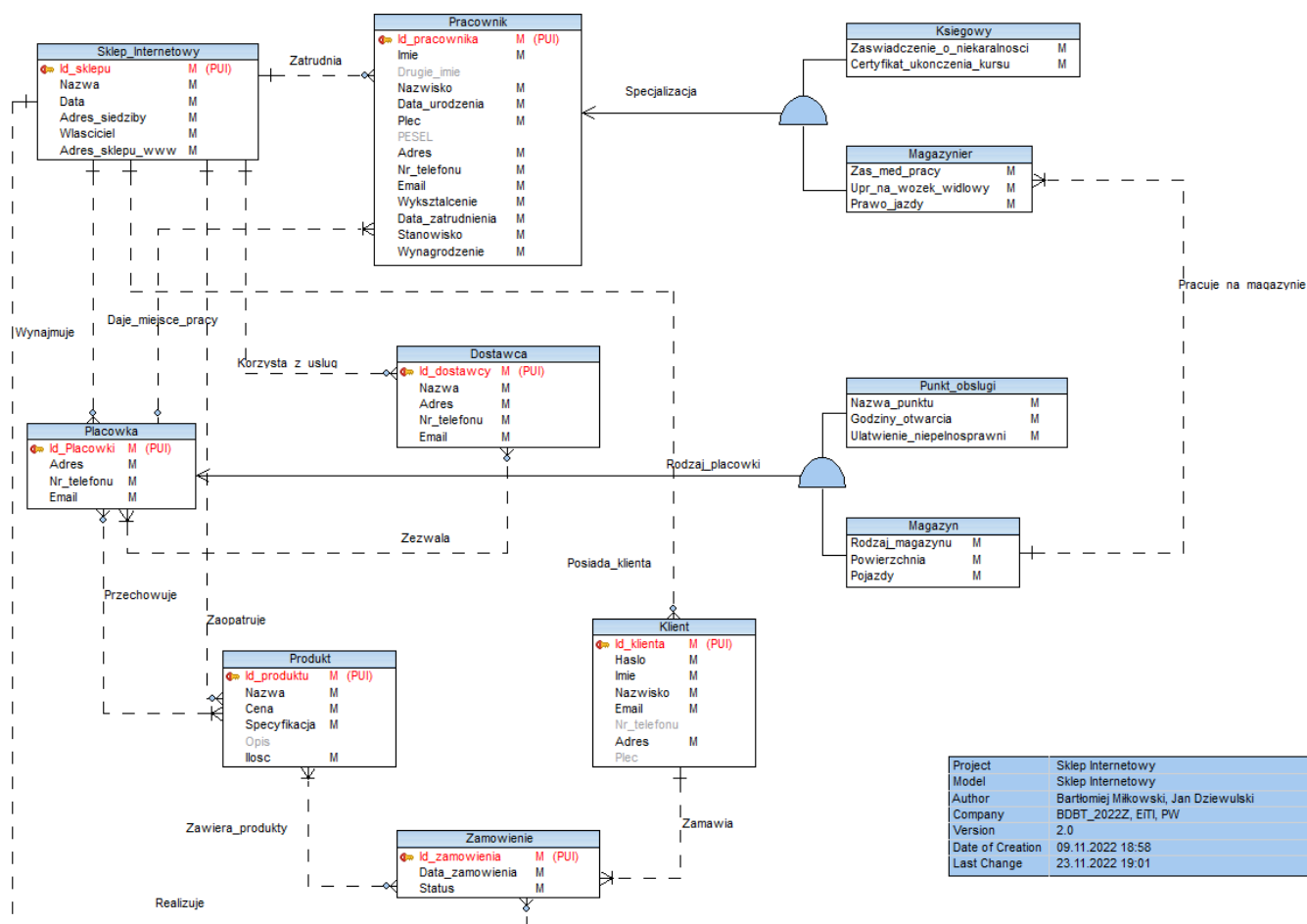


Klucze kandydujące i klucze główne

We wszystkich encjach kluczem głównym są atrybuty z przedrostkiem „Id”. Dzięki temu baza jest czytelna z pewnością, że klucze są unikalne.

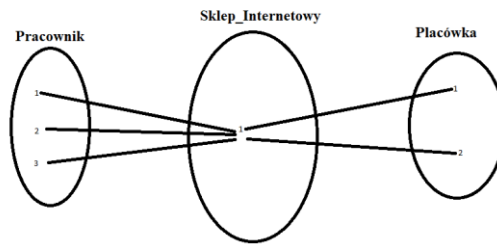
- W encjach **Sklep_Internetowy**, **Dostawca** i **Klient** kluczem kandydującym są atrybuty **Nazwa** (nie ma jednak pewności, że będą one zawsze unikatowe).
- W encji **Pracownik** kluczem kandydującym jest atrybut **PESEL** (jednak w modelu biznesowym zakłada się zatrudnianie osób bez obywatelstwa)

Schemat ER na poziomie konceptualnym



Problem pułapek szczelinowych i wachlarzowych

Na wstępnym etapie projektu conceptualnego występowały pułapki wachlarzowe. Wszystkie encje początkowo zbiegały tylko do encji głównej „Sklep_Internetowy”. Jednak tylko takie związki powodowały niejasności takie jak brak przynależności pracowników do konkretnej placówki.



Jak widać na rysunku modelowym, zarówno pracownicy jak i placówki należą do sklepu internetowego i jest to model poprawny, jednak w modelu biznesowym założono, że pracownicy są przypisani do konkretnych placówek, a w przykładzie powyżej, odczytanie, który pracownik należy do której placówki jest niemożliwe. Aby rozwiązać problem wystarczyło stworzyć nowy związek pomiędzy encjami „Pracownik” i „Placówka”.

Ten sam problem wystąpił również w przypadku encji Klient - Sklep_Internetowy - Zamowienie, w którym to klient powinien być bezpośrednio przypisany do zamówienia, które złożył. Inaczej nie wiadomo byłoby, które zamówienie dotyczy którego klienta. Tak samo w przypadku związku Placówka - Sklep_Internetowy - Produkt, gdzie placówka powinna mieć spis produktów jakie ma na stanie, w przeciwnym razie nie wiadomo, jakie produkty i w jakiej ilości dostępne są w konkretnym magazynie.

Model logiczny

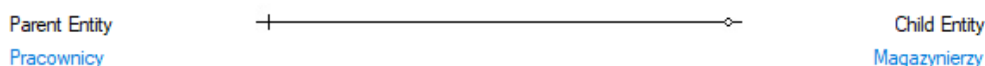
Charakterystyka modelu relacyjnego

Po opracowaniu modelu konceptualnego, projekt został przekonwertowany na model logiczny (relacyjny). Model ten odwzorowuje już rzeczywistą postać bazy. Charakteryzuje się on:




- Atrybuty identyfikujące encje stały się kluczami głównymi tabel
- Relacje n:m zostały zmienione na dwie relacje 1:n z tabelą łączącą
- Typy atrybutów zostały dostosowane do używanego silnika bazy danych (Oracle 12c Release 2)
- Relacje w encjach specjalizacji zostały zmienione na relacje 1:n

Analiza właściwości niekompatybilnych z modelem relacyjnym

- Nazwy relacji zostały zmienione z liczby pojedynczej, na mnogą. Na przykład: Pracownik - > Pracownicy
- Związki 1:n przy specjalizacjach zostały zmienione na związki 1:1. Na przykład:



- Poprawienie krotności minimalnych w związkach przy tablicach łączących, tak aby zgadzały się z krotnościami związków m:n w modelu konceptualnym
- Odkomentowanie Check Constraint Rules
- Dodatkowa operacja: Przeniesienie atrybutu „Ilosc” z encji „Produkty” do tabeli łączącej „Produkty_w_placowkach”. Zmiana ta wynika z modelu biznesowego, w którym produkty są w różnej ilości, w zależności od magazynu

Produkt			Produkty_w_placowkach		
 Id_produktu	M	(PUI)	 Id_produktu	Integer NN (PFK)	
Nazwa	M		 Id_Placowki	Integer NN (PFK)	
Cena	M		Ilosc	Integer NN	
Specyfikacja	M				
Opis					
Ilosc	M				

Proces normalizacji – analiza i przykłady

Normalizacja jest procesem, który ma na celu eliminację powtarzających się danych w bazie danych. Dzięki takiemu tworzeniu baz danych zwiększa się bezpieczeństwo danych i zmniejsza ryzyko powstania niespójności.

Normalizacja nie usuwa danych, zmienia się jedynie ich układ.


Istnieje pięć postaci normalnych, każda następna zawiera w sobie poprzednie. W większości projektów, jak również w tym, projekt jest sprowadzany do trzeciej postaci normalnej.

- **Pierwsza postać normalna (1PN)** polega na tym aby każda wartość atrybutu w krotce relacji była wartością elementarną i nie było powtarzających się grup.
- **Druga postać normalna (2PN)** jest spełniona gdy: jest w 1PN, każdy atrybut tej relacji nie wchodzący w skład żadnego klucza potencjalnego jest w pełni funkcyjnie zależny od wszystkich potencjalnych tej relacji, każdy atrybut nie wchodzący w skład klucza zależy od klucza a nie jego części, wszystkie klucze potencjalne relacji są kluczami prostymi.
- **Trzecia postać normalna (3PN)** jest spełniona gdy: jest w 1PN, jest w 2PN, gdy każdy atrybut relacji, który wchodzi w skład żadnego klucza potencjalnego nie jest przechodnio-funkcyjnie zależny od żadnego klucza potencjalnego tej relacji.

W projekcie, aby osiągnąć trzecią postać normalną, usunięto między innymi pola segmentowe i sprawdzono czy żaden niekluczowy atrybut nie jest częściowo zależny od któregoś z kluczy potencjalnych.

Podczas tego procesu, w projekcie nastąpiło kilka modyfikacji:

- Wszystkie pola „Adres” usunięto i stworzono nową relację „Adresy”, która zawiera wszystkie atrybuty proste składające się na pełny adres. Wcześniej, atrybut „Adres” powtarzał się w wielu encjach, po normalizacji, wszystkie łączą się z tylko jedną relacją.

Adresy		
 Id_adresu	Integer	NN (PK)
Miasto	Varchar2(20)	NN
Kod_pocztowy	Char(6)	NN
Ulica	Varchar2(20)	NN
Numer_domu	Varchar2(10)	NN
Numer_mieszkania	Varchar2(10)	

- Zdecydowano się na rozbicie atrybutu „Wlasciciel” w encji „Sklep_Internetowy” na tabelę „Wlasciciele”, gdyż założono, iż właścicieli może być wielu. Ponadto pole to zawierało informację o imieniu i nazwisku, co sprawiło, że atrybut stał się wielowartościowym i było wymagane jego usunięcie.

Wlasciciele		
Id_Wlasciciela	Integer	NN (PK)
Imie	Varchar2(20)	NN
Nazwisko	Varchar2(20)	NN
Id_sklepu	Integer	NN (FK) (IX1)
IX_Relationship1 (IX1)		

- W encji specjalizacji „Punkty_Obslugi” pojawiło się pole „Godziny_otwarcia”, które jest wielowartościowe, gdyż zawiera informacje nie tylko o godzinie otwarcia i zamknięcia, ale również o tym, w jaki jest to dzień tygodnia. Zdecydowano się na stworzenie zamiast tego relacji „Godziny_otwarc” ze związkiem n:m, która zawiera wszystkie informacje o godzinach otwarcia i zamknięcia sklepu.

Godziny_otwarc		
Id_godzin_otwarcia	Integer	NN (PK)
Godzina_otwarcia	Date	NN
Godzina_zamknienia	Date	NN
Dzien_tygodnia	Dzien_tygodnia_D	NN

- W przypadku atrybutu „Prawo_jazdy”, dodano atrybut „Id_prawa_jazdy”, jak również utworzono tabelę z kategoriami, co ma na celu ułatwienie w znajdowaniu pracownika z odpowiednią kategorią (pracownik może mieć tych kategorii wiele).

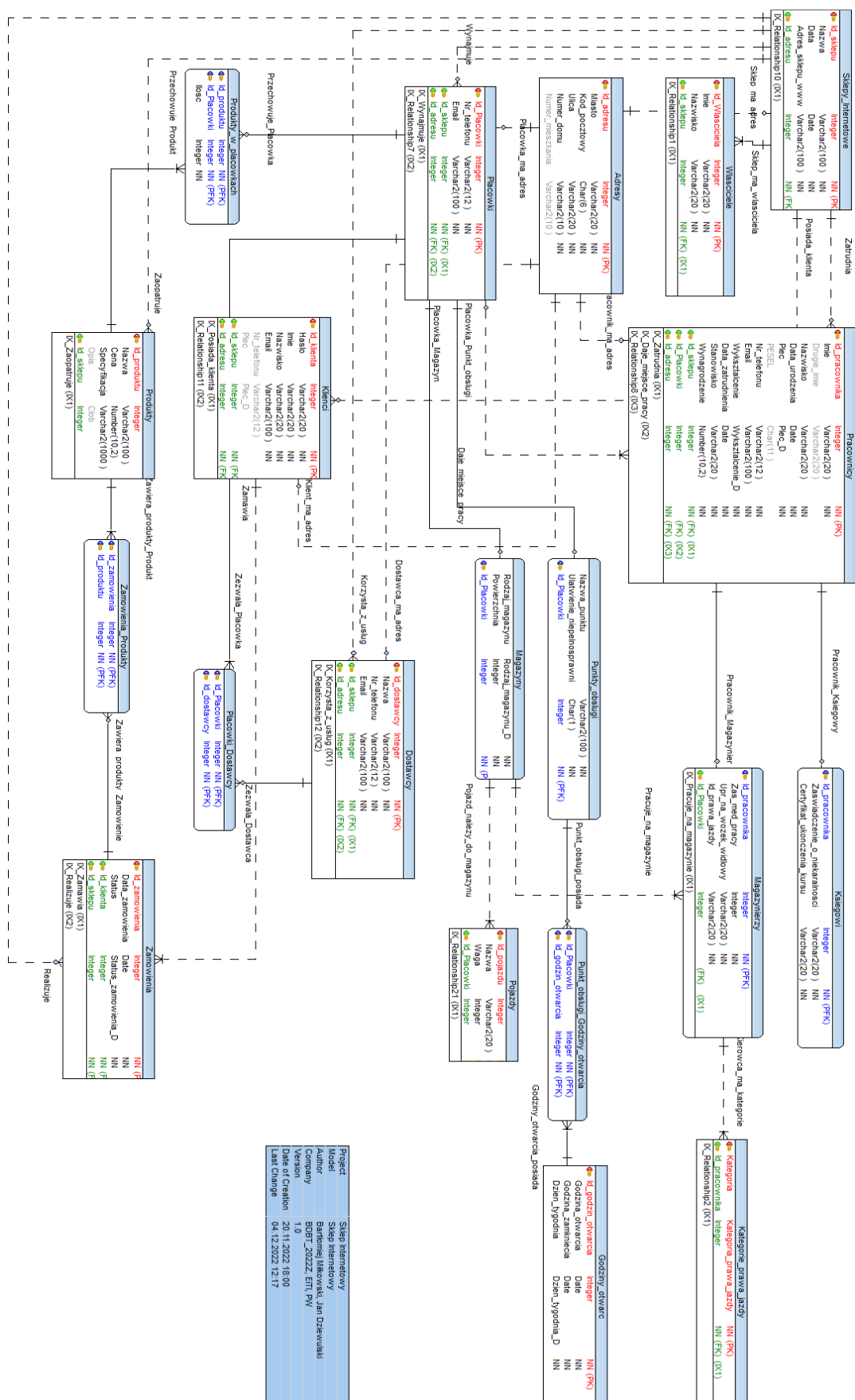
Magazynierzy		
Id_pracownika	Integer	NN (PFK)
Zas_med_pracy	Integer	NN
Upr_na_wozek_widlowy	Varchar2(20)	NN
Id_prawa_jazdy	Varchar2(20)	NN
Id_Placowki	Integer	(FK) (IX1)
IX_Pracuje_na_magazynie (IX1)		

Kategorie_prawa_jazdy		
Kategoria	Kategoria_prawa_jazdy	NN (PK)
Id_pracownika	Integer	NN (FK) (IX1)
IX_Relationship2 (IX1)		

- W „Magazyny” znajdował się atrybut „Pojazdy”, który miał spisywać listę pojazdów przypisanych do magazynu. Jednak przez to, iż pojazdów jest wiele i mają wiele czynników ich identyfikacji, utworzono tabelę „Pojazdy” powiązaną z Magazynem związkiem 1:n.

Pojazdy		
Id_pojazdu	Integer	
Nazwa	Varchar2(20)	
Waga	Integer	
Id_Placowki	Integer	
IX_Relationship21 (IX1)		

Schemat ER na poziomie modelu logicznego



Więzy integralności

Wszystkie typy pól zostały stworzone w taki sposób, że są integralne z ich atrybutami. To znaczy, że atrybuty mają swoje własne unikatowe typy pasujące do odgrywanej przez nich roli. Przykładowo Wynagrodzenie jest określone typem Number(10,2) specjalnie przeznaczonym do zapisu ilości pieniędzy i przewidującej część ułamkową. Innym mniej trywialnym przykładem mógłby być typ VarChar2(20), za pomocą którego zapisywane jest imię. Założono, że imię może być różnej długości, ale nie będzie przekraczać 20 znaków. Atrybuty nieobowiązkowe, mogą przyjmować wartość NULL. Jest to wartość, która ma imitować puste miejsce w przypadku braku wypełnienia pola.

Proces denormalizacji – analiza i przykłady

Proces denormalizacji polega na wprowadzeniu kontrolowanej redundancji poprzez złagodzenie zakresu stosowania normalizacji.

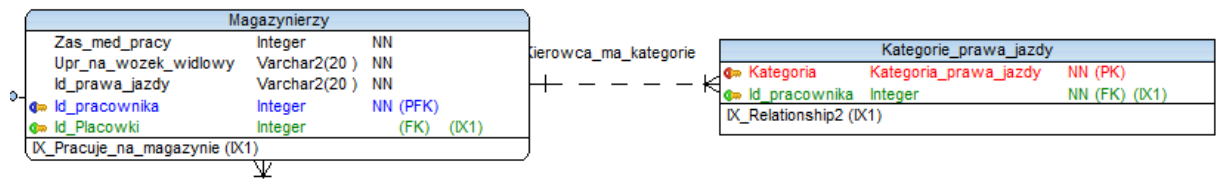
W przypadku tego projektu, denormalizacja okazała się być w większości przypadków zbędna. W jednym przypadku proces ten jednak został zastosowany, a w dwóch innych rozważany:

- W jednym z przypadków pogodzono się z redundancją danych, na rzecz potrzeb biznesowych. Encja „Pracownicy” przechowuje klucz obcy z „Id_Placowki”, do której pracownik może, aczkolwiek nie musi być przypisany. W specjalizacji pracowników, magazynierów, występuje związek pomiędzy nimi a magazynami, co powoduje dziedziczenie po raz drugi klucza obcego „Id_Placowki”. Operacja ta była jednak niezbędna, gdyż każdy magazynier jest z góry, w dniu zatrudnienia, przypisywany do konkretnego magazynu, i jest to w tym przypadku pole obowiązkowe.

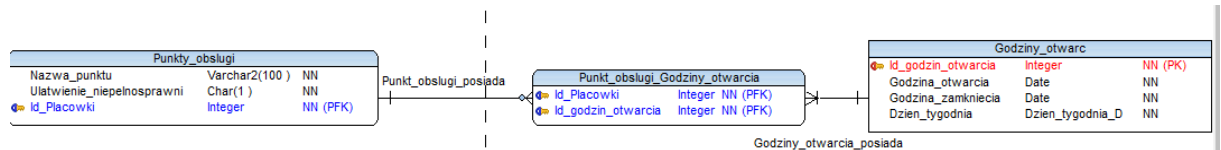
Pracownicy		
Id_pracownika	Integer	NN (F)
Imie	Varchar2(20)	NN
Drugie_imie	Varchar2(20)	
Nazwisko	Varchar2(20)	NN
Data_urodzenia	Date	NN
Plec	Plec_D	NN
PESEL	Char(11)	
Nr_telefonu	Varchar2(12)	NN
Email	Varchar2(100)	NN
Wykształcenie	Wykształcenie_D	NN
Data_zatrudnienia	Date	NN
Stanowisko	Varchar2(20)	NN
Wynagrodzenie	Number(10,2)	NN
Id_sklepu	Integer	NN (F)
Id_Placowki	Integer	(F)
Id_adresu	Integer	NN (F)
IX_Zatrudnia (IX1)		
IX_Daje_miejsce_pracy (IX2)		
IX_Relationship8 (IX3)		

Magazynierzy		
Zas_med_pracy	Integer	NN
Upr_na_wozek_widlowy	Varchar2(20)	NN
Id_prawa_jazdy	Varchar2(20)	NN
Id_pracownika	Integer	NN (PFK)
Id_Placowki	Integer	(FK) (IX1)
IX_Pracuje_na_magazynie (IX1)		

- Z tabeli Kategorie_prawa_jazdy korzystają jedynie Magazynierzy, tak więc połączenie tabel nie powodowało by dużej redundancji. Możliwe byłoby wprowadzenie do tabeli Magazynierzy kategorii prawa jazdy, które przyjmowały by dwie wartości: 1 i 0 (prawda lub fałsz). Zdecydowano się jednak na stworzenie dwóch tabel w celu większej czytelności.



- W tym przypadku Punkty_obsługi posiadające godziny otwarcia mogłyby przechowywać jedynie atrybuty z informacją o godzinach otwarcia i zamknięcia punktu, jednak w modelu biznesowym założono, że punkty mogą pracować różnie w zależności od dnia tygodnia czy wypadającego w danym dniu święta.



Faza Fizyczna

Projekt transakcji i weryfikacja ich wykonalności

Transakcja	Relacje biorące udział	Czy wykonalne
Podgląd informacji o Sklepie Internetowym	Sklepy_Internetowe, Adresy, Wlasciciele	TAK
Modyfikacja informacji o Sklepie Internetowym	Sklepy_Internetowe, Adresy, Wlasciciele	TAK
Podgląd informacji o Pracownikach	Pracownicy, Sklepy_Internetowe, Adresy, Placowki	TAK
Modyfikacja informacji o Pracownikach	Pracownicy, Sklepy_Internetowe, Adresy, Placowki	TAK
Podgląd informacji o Placówkach	Placowki, Adresy, Sklepy_Internetowe	TAK
Modyfikacja informacji o Placówkach	Placowki, Adresy, Sklepy_Internetowe	TAK
Podgląd informacji o Dostawcach	Dostawcy, Sklepy_Internetowe, Adresy	TAK
Modyfikacja informacji o Dostawcach	Dostawcy, Sklepy_Internetowe, Adresy	TAK
Podgląd informacji o Produktach	Produkty, Sklepy_Internetowe	TAK
Modyfikacja informacji o Produktach	Produkty, Sklepy_Internetowe	TAK
Podgląd informacji o Klientach	Klienci, Sklepy_Internetowe, Adresy	TAK
Modyfikacja informacji o Klientach	Klienci, Sklepy_Internetowe, Adresy	TAK
Podgląd informacji o Zamówieniach	Zamowienia, Sklepy_Internetowe, Klienci	TAK
Modyfikacja informacji o Zamówieniach	Zamowienia, Sklepy_Internetowe, Klienci	TAK

Podgląd informacji o Księgowych	Pracownicy, Księgowi	TAK
Modyfikacja informacji o Księgowych	Pracownicy, Księgowi	TAK
Podgląd informacji o Magazynierach	Pracownicy, Magazynierzy, Placowki	TAK
Modyfikacja informacji o Magazynierach	Pracownicy, Magazynierzy, Placowki	TAK
Podgląd informacji o Punktach Obsługi	Placowki, Punkty_Obslugi	TAK
Modyfikacja informacji o Punktach Obsługi	Placowki, Punkty_Obslugi	TAK
Podgląd informacji o Magazynach	Placowki, Magazyny	TAK
Modyfikacja informacji o Magazynach	Placowki, Magazyny	TAK
Podgląd informacji o Ilości danego produktu w placówce	Produkty_w_placowkach, Placowki, Produkty	TAK
Modyfikacja informacji o ilości danego produktu w placówce	Produkty_w_placowkach, Placowki, Produkty	TAK
Podgląd informacji o Pojazdach	Pojazdy, Placowki	TAK
Modyfikacja informacji o Pojazdach	Pojazdy, Placowki	TAK
Podgląd informacji o Godzinach otwarć	Godziny_otwarc	TAK
Modyfikacja informacji o Godzinach otwarć	Godziny_otwarc	TAK
Podgląd informacji o Kategoriach prawa jazdy	Kategorie_prawa_jazdy, Pracownicy	TAK
Modyfikacja informacji o Kategoriach prawa jazdy	Kategorie_prawa_jazdy, Pracownicy	TAK

Strojenie baz danych – dobór indeksów

Potrzebne do strojenia bazy danych indeksy zostały wygenerowane automatycznie:

- CREATE INDEX IX_Sklepy_maja ON Sklepy_Internetowe (Id_adresu)
- CREATE INDEX IX_Zatrudnia ON Pracownicy (Id_sklepu)
- CREATE INDEX IX_Daje_miejsce_pracy ON Pracownicy (Id_Placowki)
- CREATE INDEX IX_Pracownicy_maja_adres ON Pracownicy (Id_adresu)
- CREATE INDEX IX_Pracuje_na_magazynie ON Magazynierzy (Id_Placowki)
- CREATE INDEX IX_Zaopatruje ON Produkty (Id_sklepu)
- CREATE INDEX IX_Posiada_klienta ON Klienci (Id_sklepu)
- CREATE INDEX IX_Klienci_maja_adres ON Klienci (Id_adresu)
- CREATE INDEX IX_dostawcy_maja_adres ON Dostawcy (Id_adresu)
- CREATE INDEX IX_Wynajmuje ON Placowki (Id_sklepu)
- CREATE INDEX IX_Placowki_maja_adres ON Placowki (Id_adresu)
- CREATE INDEX IX_Realizuje ON Zamowienia (Id_sklepu)
- CREATE INDEX IX_Sklep_ma_wlascicieli ON Wlasciciele (Id_sklepu)
- CREATE INDEX IX_Pracownik_ma_kategorie ON Kategorie_prawa_jazdy (Id_pracownika)
- CREATE INDEX IX_Placowki_maja_pojazdy ON Pojazdy (Id_Placowki)

Skrypt SQL

-- Create sequences section -----

CREATE SEQUENCE SklepS1

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE PracownikS2

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE PlacowkaS3

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE DostawcaS4

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE ProduktS5

```

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE ZamowienieS6

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE KlientS7

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE AdresS8

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE Godz_otwS9

NOMAXVALUE

NOMINVALUE

CACHE 20

/

```

```

CREATE SEQUENCE Kategoria_pr_jazdyS10

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

CREATE SEQUENCE WlascicielS11

INCREMENT BY 1

START WITH 1

NOMAXVALUE

NOMINVALUE

CACHE 20

/

-- Create tables section -----

-- Table Sklepy_Internetowe

CREATE TABLE Sklepy_Internetowe(

    Id_sklepu Integer NOT NULL,

    Nazwa Varchar2(100 ) NOT NULL,

    Data Date NOT NULL,

    Adres_sklepu_www Varchar2(100 ) NOT NULL,

    Id_adresu Integer NOT NULL

)

/

-- Create indexes for table Sklepy_Internetowe

CREATE INDEX IX_Relationship10 ON Sklepy_Internetowe (Id_adresu)

/

-- Add keys for table Sklepy_Internetowe

ALTER TABLE Sklepy_Internetowe ADD CONSTRAINT Sklep_PK PRIMARY KEY (Id_sklepu)

/

-- Table Pracownicy

```

```

CREATE TABLE Pracownicy(
    Id_pracownika Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Drugie_imie Varchar2(20 ),
    Nazwisko Varchar2(20 ) NOT NULL,
    Data_urodzenia Date NOT NULL,
    Plec Varchar2(1 ) NOT NULL
        CHECK (Plec IN ('M','K')),
    PESEL Varchar2(11 ),
    Nr_telefonu Varchar2(12 ) NOT NULL,
    Email Varchar2(100 ) NOT NULL,
    Wykształcenie Varchar2(1 ) NOT NULL
        CHECK (Wykształcenie IN ('P','S','Z','W')),
    Data_zatrudnienia Date NOT NULL,
    Stanowisko Varchar2(20 ) NOT NULL,
    Wynagrodzenie Number(10,2) NOT NULL,
    Id_sklepu Integer NOT NULL,
    Id_Placowki Integer,
    Id_adresu Integer NOT NULL
)
/
-- Create indexes for table Pracownicy
CREATE INDEX IX_Zatrudnia ON Pracownicy (Id_sklepu)
/
CREATE INDEX IX_Daje_miejsce_pracy ON Pracownicy (Id_Placowki)
/
CREATE INDEX IX_Relationship8 ON Pracownicy (Id_adresu)
/
-- Add keys for table Pracownicy
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_PK PRIMARY KEY (Id_pracownika)
/
-- Table Ksiegowi

```

```

CREATE TABLE Ksiegowi(

    Zaswiadczenie_o_niekaralnosci Varchar2(20 ) NOT NULL,

    Certyfikat_ukonczenia_kursu Varchar2(20 ) NOT NULL,

    Id_pracownika Integer NOT NULL

)

/

-- Add keys for table Ksiegowi

ALTER TABLE Ksiegowi ADD CONSTRAINT Unique_Identifier1 PRIMARY KEY (Id_pracownika)

/

-- Table Magazynierzy

CREATE TABLE Magazynierzy(

    Zas_med_pracy Integer NOT NULL,

    Upr_na_wozek_widlowy Varchar2(20 ) NOT NULL,

    Id_prawa_jazdy Varchar2(20 ) NOT NULL,

    Id_pracownika Integer NOT NULL,

    Id_Placowki Integer

)

/

-- Create indexes for table Magazynierzy

CREATE INDEX IX_Pracuje_na_magazynie ON Magazynierzy (Id_Placowki)

/

-- Add keys for table Magazynierzy

ALTER TABLE Magazynierzy ADD CONSTRAINT Unique_Identifier2 PRIMARY KEY (Id_pracownika)

/

-- Table and Columns comments section

COMMENT ON COLUMN Magazynierzy.Zas_med_pracy IS 'Unikalny numer zaÅwiadczenia od lekarza medycyny pracy'

/

COMMENT ON COLUMN Magazynierzy.Id_prawa_jazdy IS 'Unikalny numer prawa jazdy'

/

-- Table Produkty

```

```

CREATE TABLE Produkty(
    Id_produktu Integer NOT NULL,
    Nazwa Varchar2(100 ) NOT NULL,
    Cena Number(10,2) NOT NULL,
    Specyfikacja Varchar2(1000 ) NOT NULL,
    Opis Clob,
    Id_sklepu Integer NOT NULL
)
/

-- Create indexes for table Produkty

CREATE INDEX IX_Zaopatrzuje ON Produkty (Id_sklepu)

/

-- Add keys for table Produkty

ALTER TABLE Produkty ADD CONSTRAINT Produkt_PK PRIMARY KEY (Id_produktu)

/

-- Table Klienci

CREATE TABLE Klienci(
    Id_klienta Integer NOT NULL,
    Haslo Varchar2(20 ) NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(20 ) NOT NULL,
    Email Varchar2(100 ) NOT NULL,
    Nr_telefonu Varchar2(12 ),
    Plec Varchar2(1 )
        CHECK (Plec IN ('M','K')),
    Id_sklepu Integer NOT NULL,
    Id_adresu Integer NOT NULL
)
/

-- Create indexes for table Klienci

CREATE INDEX IX_Posiada_klienta ON Klienci (Id_sklepu)

/

```



```

CREATE INDEX IX_Relationship11 ON Klienci (Id_adresu)

/

-- Add keys for table Klienci

ALTER TABLE Klienci ADD CONSTRAINT Klient_PK PRIMARY KEY (Id_klienta)

/

-- Table Dostawcy

CREATE TABLE Dostawcy(

    Id_dostawcy Integer NOT NULL,

    Nazwa Varchar2(100 ) NOT NULL,

    Nr_telefonu Varchar2(12 ) NOT NULL,

    Email Varchar2(100 ) NOT NULL,

    Id_sklepu Integer NOT NULL,

    Id_adresu Integer NOT NULL

)

/

-- Create indexes for table Dostawcy

CREATE INDEX IX_Korzysta_z_uslug ON Dostawcy (Id_sklepu)

/

CREATE INDEX IX_Relationship12 ON Dostawcy (Id_adresu)

/

-- Add keys for table Dostawcy

ALTER TABLE Dostawcy ADD CONSTRAINT Dostawca_PK PRIMARY KEY (Id_dostawcy)

/

-- Table Placowki

CREATE TABLE Placowki(

    Id_Placowki Integer NOT NULL,

    Nr_telefonu Varchar2(12 ) NOT NULL,

    Email Varchar2(100 ) NOT NULL,

    Id_sklepu Integer NOT NULL,

    Id_adresu Integer NOT NULL

)

```

```

/

-- Create indexes for table Placowki

CREATE INDEX IX_Wynajmuje ON Placowki (Id_sklepu)

/

CREATE INDEX IX_Relationship7 ON Placowki (Id_adresu)

/

-- Add keys for table Placowki

ALTER TABLE Placowki ADD CONSTRAINT Placowka_PK PRIMARY KEY (Id_Placowki)

/

-- Table Punkty_obsługi

CREATE TABLE Punkty_obsługi(

    Nazwa_punktu Varchar2(100 ) NOT NULL,

    Ulatwienie_niepełnosprawni Char(1 ) NOT NULL,

    Id_Placowki Integer NOT NULL

)

/

-- Add keys for table Punkty_obsługi

ALTER TABLE Punkty_obsługi ADD CONSTRAINT Unique_Identifier3 PRIMARY KEY (Id_Placowki)

/

-- Table Magazyny

CREATE TABLE Magazyny(

    Rodzaj_magazynu Varchar2(1 ) NOT NULL

    CHECK (Rodzaj_magazynu IN ('C', 'R', 'W')),

    Powierzchnia Integer NOT NULL,

    Id_Placowki Integer NOT NULL

)

/

-- Add keys for table Magazyny

ALTER TABLE Magazyny ADD CONSTRAINT Unique_Identifier4 PRIMARY KEY (Id_Placowki)

/

-- Table Zamowienia

CREATE TABLE Zamowienia(

```

```

    Id_zamowienia Integer NOT NULL,

    Data_zamowienia Date NOT NULL,

    Status Varchar2(1 ) NOT NULL

        CHECK (Status IN ('O','P','C','S','D')),

    Id_klienta Integer NOT NULL,

    Id_sklepu Integer NOT NULL

)

/

-- Create indexes for table Zamowienia

CREATE INDEX IX_Zamawia ON Zamowienia (Id_klienta)

/

CREATE INDEX IX_Realizuje ON Zamowienia (Id_sklepu)

/

-- Add keys for table Zamowienia

ALTER TABLE Zamowienia ADD CONSTRAINT Zamowienie_PK PRIMARY KEY (Id_zamowienia)

/

-- Table Produkt_Placowka

CREATE TABLE Produkt_Placowka(

    Id_produktu Integer NOT NULL,

    Id_Placowki Integer NOT NULL,

    Ilosc Integer NOT NULL

)

/

-- Table and Columns comments section

COMMENT ON COLUMN Produkt_Placowka.Ilosc IS 'Ilość danego produktu na magazynie'

/

-- Table Zamowienie_Produkt

CREATE TABLE Zamowienie_Produkt(

    Id_zamowienia Integer NOT NULL,

    Id_produktu Integer NOT NULL

)

```

```

/

-- Table Placowka_Dostawca

CREATE TABLE Placowka_Dostawca(

    Id_Placowki Integer NOT NULL,

    Id_dostawcy Integer NOT NULL

)

/

-- Table Wlasciciele

CREATE TABLE Wlasciciele(

    Id_Wlasciciela Integer NOT NULL,

    Imie Varchar2(20 ) NOT NULL,

    Nazwisko Varchar2(20 ) NOT NULL,

    Id_sklepu Integer NOT NULL

)

/

-- Create indexes for table Wlasciciele

CREATE INDEX IX_Relationship1 ON Wlasciciele (Id_sklepu)

/

-- Add keys for table Wlasciciele

ALTER TABLE Wlasciciele ADD CONSTRAINT PK_Wlasciciele PRIMARY KEY (Id_Wlasciciela)

/

-- Table and Columns comments section

COMMENT ON TABLE Wlasciciele IS 'Właściciele sklepu'

/

COMMENT ON COLUMN Wlasciciele.Id_Wlasciciela IS 'Unikalny numer właściciela'

/

COMMENT ON COLUMN Wlasciciele.Imie IS 'Imię właściciela'

/

COMMENT ON COLUMN Wlasciciele.Nazwisko IS 'Nazwisko właściciela'

/

-- Table Kategorie_prawa_jazdy

CREATE TABLE Kategorie_prawa_jazdy(

```

```

Kategoria Char(20 ) NOT NULL

    CHECK (Kategoria IN ('A1', 'A2', 'A', 'B1', 'B', 'C1', 'C', 'D1', 'D' )),

Id_pracownika Integer NOT NULL

)

/

-- Create indexes for table Kategorie_prawa_jazdy

CREATE INDEX IX_Relationship2 ON Kategorie_prawa_jazdy (Id_pracownika)

/

-- Add keys for table Kategorie_prawa_jazdy

ALTER TABLE Kategorie_prawa_jazdy ADD CONSTRAINT PK_Kategorie_prawa_jazdy PRIMARY KEY (Kategoria)

/

-- Table and Columns comments section

COMMENT ON TABLE Kategorie_prawa_jazdy IS 'Kategorie prawa jazdy'

/

COMMENT ON COLUMN Kategorie_prawa_jazdy.Kategoria IS 'Kategoria prawa jazdy: A1, A2, A, B1, B, C1, C, D1, D'

/

-- Table Adresy

CREATE TABLE Adresy(

    Id_adresu Integer NOT NULL,

    Miasto Varchar2(20 ) NOT NULL,

    Kod_pocztowy Varchar2(6 ) NOT NULL,

    Ulica Varchar2(20 ) NOT NULL,

    Numer_domu Varchar2(10 ) NOT NULL,

    Numer_mieszkania Varchar2(10 )

)

/

-- Add keys for table Adresy

ALTER TABLE Adresy ADD CONSTRAINT PK_Adresy PRIMARY KEY (Id_adresu)

/

-- Table and Columns comments section

COMMENT ON TABLE Adresy IS 'Adresy'

/

```

```

COMMENT ON COLUMN Adresy.Id_adresu IS 'Unikalny identyfikator adresu'

/

COMMENT ON COLUMN Adresy.Miasto IS 'Nazwa miasta'

/

COMMENT ON COLUMN Adresy.Kod_pocztowy IS 'Kod pocztowy: xx-xxx'

/

COMMENT ON COLUMN Adresy.Ulica IS 'Nazwa ulicy'

/

COMMENT ON COLUMN Adresy.Numer_domu IS 'Numer domu'

/

COMMENT ON COLUMN Adresy.Numer_mieszkania IS 'Numer mieszkania'

/

-- Table Godziny_otwarc
CREATE TABLE Godziny_otwarc(
    Id_godzin_otwarcia Integer NOT NULL,
    Godzina_otwarcia Date NOT NULL,
    Godzina_zamknienia Date NOT NULL,
    Dzień_tygodnia Varchar2(1 ) NOT NULL
        CHECK (Dzień_tygodnia IN ('M','T','W','H','F','S','D'))
)

/

-- Add keys for table Godziny_otwarc
ALTER TABLE Godziny_otwarc ADD CONSTRAINT PK_Godziny_otwarc PRIMARY KEY (Id_godzin_otwarcia)

/

-- Table and Columns comments section
COMMENT ON TABLE Godziny_otwarc IS 'Godziny otwarcia punktu obsÅugi'

/

COMMENT ON COLUMN Godziny_otwarc.Id_godzin_otwarcia IS 'Unikatowy numer godzin otwarcia punktu obsÅugi klienta'

/

COMMENT ON COLUMN Godziny_otwarc.Godzina_otwarcia IS 'Godzina otwarcia punktu'

/

```

```

COMMENT ON COLUMN Godziny_otwarc.Godzina_zamknienia IS 'Godzina zamknięcia punktu'

/

COMMENT ON COLUMN Godziny_otwarc.Dzien_tygodnia IS 'Dzień, tygodnia: M - Poniedziałek, T - Wtorek, W -
Środa, H - Czwartek, F - Czwartek, S - Sobota, D - Niedziela'

/

-- Table Punkt_obsługi_Godziny_otwarcia

CREATE TABLE Punkt_obsługi_Godziny_otwarcia(

    Id_Placowki Integer NOT NULL,

    Id_godzin_otwarcia Integer NOT NULL

)

/

-- Add keys for table Punkt_obsługi_Godziny_otwarcia

ALTER TABLE Punkt_obsługi_Godziny_otwarcia ADD CONSTRAINT PK_Punkt_obsługi_Godziny_otwarcia PRIMARY
KEY (Id_Placowki,Id_godzin_otwarcia)

/

-- Table Pojazdy

CREATE TABLE Pojazdy(

    Id_pojazdu Integer NOT NULL,

    Nazwa Varchar2(20 ) NOT NULL,

    Waga Integer NOT NULL,

    Id_Placowki Integer NOT NULL

)

/

-- Create indexes for table Pojazdy

CREATE INDEX IX_Relationship21 ON Pojazdy (Id_Placowki)

/

-- Add keys for table Pojazdy

ALTER TABLE Pojazdy ADD CONSTRAINT PK_Pojazdy PRIMARY KEY (Id_pojazdu)

/

-- Table and Columns comments section

COMMENT ON TABLE Pojazdy IS 'Pojazdy'

/

COMMENT ON COLUMN Pojazdy.Id_pojazdu IS 'Unikalny numer pojazdu należący do magazynu'

```

```

/

COMMENT ON COLUMN Pojazdy.Nazwa IS 'Nazwa pojazdu'

/

COMMENT ON COLUMN Pojazdy.Waga IS 'Waga pojazdu'

/


-- Trigger for sequence SklepS1 for column Id_sklepu in table Sklepy_Internetowe -----
CREATE OR REPLACE TRIGGER ts_Sklepy_Internetowe_SklepS1 BEFORE INSERT
ON Sklepy_Internetowe FOR EACH ROW
BEGIN
    :new.Id_sklepu := SklepS1.nextval;
END;

/

CREATE OR REPLACE TRIGGER tsu_Sklepy_Internetowe_SklepS1 AFTER UPDATE OF Id_sklepu
ON Sklepy_Internetowe FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_sklepu in table Sklepy_Internetowe as it uses
sequence.');
```

```

END;

/

-- Trigger for sequence PracownikS2 for column Id_pracownika in table Pracownicy -----
CREATE OR REPLACE TRIGGER ts_Pracownicy_PracownikS2 BEFORE INSERT
ON Pracownicy FOR EACH ROW
BEGIN
    :new.Id_pracownika := PracownikS2.nextval;
END;

/

CREATE OR REPLACE TRIGGER tsu_Pracownicy_PracownikS2 AFTER UPDATE OF Id_pracownika
ON Pracownicy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_pracownika in table Pracownicy as it uses sequence.');
```

```

END;
```



```

/

-- Trigger for sequence ProduktS5 for column Id_produktu in table Produkty -----
CREATE OR REPLACE TRIGGER ts_Produkty_ProduktS5 BEFORE INSERT
ON Produkty FOR EACH ROW
BEGIN

:new.Id_produktu := ProduktS5.nextval;

END;

/

CREATE OR REPLACE TRIGGER tsu_Produkty_ProduktS5 AFTER UPDATE OF Id_produktu
ON Produkty FOR EACH ROW
BEGIN

RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_produktu in table Produkty as it uses sequence.');
```

```

END;

/

-- Trigger for sequence KlientS7 for column Id_klienta in table Klienci -----
CREATE OR REPLACE TRIGGER ts_Klienci_KlientS7 BEFORE INSERT
ON Klienci FOR EACH ROW
BEGIN

:new.Id_klienta := KlientS7.nextval;

END;

/

CREATE OR REPLACE TRIGGER tsu_Klienci_KlientS7 AFTER UPDATE OF Id_klienta
ON Klienci FOR EACH ROW
BEGIN

RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_klienta in table Klienci as it uses sequence.');
```

```

END;

/

-- Trigger for sequence DostawcaS4 for column Id_dostawcy in table Dostawcy -----
CREATE OR REPLACE TRIGGER ts_Dostawcy_DostawcaS4 BEFORE INSERT
ON Dostawcy FOR EACH ROW
BEGIN

:new.Id_dostawcy := DostawcaS4.nextval;
```

```

END;

/

CREATE OR REPLACE TRIGGER tsu_Dostawcy_DostawcaS4 AFTER UPDATE OF Id_dostawcy
ON Dostawcy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_dostawcy in table Dostawcy as it uses sequence.');
```

```

END;

/

-- Trigger for sequence PlacowkaS3 for column Id_Placowki in table Placowki -----
CREATE OR REPLACE TRIGGER ts_Placowki_PlacowkaS3 BEFORE INSERT
ON Placowki FOR EACH ROW
BEGIN
    :new.Id_Placowki := PlacowkaS3.nextval;
END;

/

CREATE OR REPLACE TRIGGER tsu_Placowki_PlacowkaS3 AFTER UPDATE OF Id_Placowki
ON Placowki FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_Placowki in table Placowki as it uses sequence.');
```

```

END;

/

-- Trigger for sequence ZamowienieS6 for column Id_zamowienia in table Zamowienia -----
CREATE OR REPLACE TRIGGER ts_Zamowienia_ZamowienieS6 BEFORE INSERT
ON Zamowienia FOR EACH ROW
BEGIN
    :new.Id_zamowienia := ZamowienieS6.nextval;
END;

/

CREATE OR REPLACE TRIGGER tsu_Zamowienia_ZamowienieS6 AFTER UPDATE OF Id_zamowienia
ON Zamowienia FOR EACH ROW
BEGIN
```

```

    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_zamowienia in table Zamowienia as it uses
sequence.');
```

END;

/

-- Trigger for sequence WlascicielS11 for column Id_Wlasciciela in table Wlascieciele -----

```

CREATE OR REPLACE TRIGGER ts_Wlascieciele_WlascicielS11 BEFORE INSERT
ON Wlascieciele FOR EACH ROW
BEGIN
    :new.Id_Wlasciciela := WlascicielS11.nextval;
END;
```

/

```

CREATE OR REPLACE TRIGGER tsu_Wlascieciele_WlascicielS11 AFTER UPDATE OF Id_Wlasciciela
ON Wlascieciele FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_Wlasciciela in table Wlascieciele as it uses sequence.');
```

END;

/

-- Trigger for sequence Kategoria_pr_jazdyS10 for column Kategoria in table Kategoria_prawa_jazdy -----

```

CREATE OR REPLACE TRIGGER ts_Kategoria_prawa_jazdy_Kategoria_pr_jazdyS10 BEFORE INSERT
ON Kategoria_prawa_jazdy FOR EACH ROW
BEGIN
    :new.Kategoria := Kategoria_pr_jazdyS10.nextval;
END;
```

/

```

CREATE OR REPLACE TRIGGER tsu_Kategoria_prawa_jazdy_Kategoria_pr_jazdyS10 AFTER UPDATE OF Kategoria
ON Kategoria_prawa_jazdy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Kategoria in table Kategoria_prawa_jazdy as it uses
sequence.');
```

END;

/

-- Trigger for sequence AdresS8 for column Id_adresu in table Adresy -----

```

CREATE OR REPLACE TRIGGER ts_Adresy_AdresS8 BEFORE INSERT
```

```

ON Adresy FOR EACH ROW

BEGIN

    :new.Id_adresu := AdresS8.nextval;

END;

/

CREATE OR REPLACE TRIGGER tsu_Adresy_AdresS8 AFTER UPDATE OF Id_adresu
ON Adresy FOR EACH ROW

BEGIN

    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_adresu in table Adresy as it uses sequence.');
```

```

END;

/

-- Trigger for sequence Godz_otwS9 for column Id_godzin_otwarcia in table Godziny_otwarc -----

CREATE OR REPLACE TRIGGER ts_Godziny_otwarc_Godz_otwS9 BEFORE INSERT
ON Godziny_otwarc FOR EACH ROW

BEGIN

    :new.Id_godzin_otwarcia := Godz_otwS9.nextval;

END;

/

CREATE OR REPLACE TRIGGER tsu_Godziny_otwarc_Godz_otwS9 AFTER UPDATE OF Id_godzin_otwarcia
ON Godziny_otwarc FOR EACH ROW

BEGIN

    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Id_godzin_otwarcia in table Godziny_otwarc as it uses
sequence.');
```

```

END;

/

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

ALTER TABLE Pracownicy ADD CONSTRAINT Zatrudnia FOREIGN KEY (Id_sklepu) REFERENCES
Sklepy_Internetowe (Id_sklepu)

/

ALTER TABLE Dostawcy ADD CONSTRAINT Korzysta_z_uslug FOREIGN KEY (Id_sklepu) REFERENCES
Sklepy_Internetowe (Id_sklepu)

/
```

ALTER TABLE Pracownicy ADD CONSTRAINT Daje_miejsce_pracy FOREIGN KEY (Id_Placowki) REFERENCES Placowki (Id_Placowki)

/

ALTER TABLE Magazynierzy ADD CONSTRAINT Pracuje_na_magazynie FOREIGN KEY (Id_Placowki) REFERENCES Magazyny (Id_Placowki)

/

ALTER TABLE Placowki ADD CONSTRAINT Wynajmuje FOREIGN KEY (Id_sklepu) REFERENCES Sklepy_Internetowe (Id_sklepu)

/

ALTER TABLE Produkty ADD CONSTRAINT Zaopatruje FOREIGN KEY (Id_sklepu) REFERENCES Sklepy_Internetowe (Id_sklepu)

/

ALTER TABLE Zamowienia ADD CONSTRAINT Zamawia FOREIGN KEY (Id_klienta) REFERENCES Klienci (Id_klienta)

/

ALTER TABLE Klienci ADD CONSTRAINT Posiada_klienta FOREIGN KEY (Id_sklepu) REFERENCES Sklepy_Internetowe (Id_sklepu)

/

ALTER TABLE Zamowienia ADD CONSTRAINT Realizuje FOREIGN KEY (Id_sklepu) REFERENCES Sklepy_Internetowe (Id_sklepu)

/

ALTER TABLE Wlasciele ADD CONSTRAINT Sklep_ma_wlasciciela FOREIGN KEY (Id_sklepu) REFERENCES Sklepy_Internetowe (Id_sklepu)

/

ALTER TABLE Kategorie_prawa_jazdy ADD CONSTRAINT Kierowca_ma_kategorie FOREIGN KEY (Id_pracownika) REFERENCES Magazynierzy (Id_pracownika)

/

ALTER TABLE Placowki ADD CONSTRAINT Placowka_ma_adres FOREIGN KEY (Id_adresu) REFERENCES Adresy (Id_adresu)

/

ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_ma_adres FOREIGN KEY (Id_adresu) REFERENCES Adresy (Id_adresu)

/

ALTER TABLE Sklepy_Internetowe ADD CONSTRAINT Sklep_ma_adres FOREIGN KEY (Id_adresu) REFERENCES Adresy (Id_adresu)

/

ALTER TABLE Klienci ADD CONSTRAINT Klient_ma_adres FOREIGN KEY (Id_adresu) REFERENCES Adresy (Id_adresu)

/

```
ALTER TABLE Dostawcy ADD CONSTRAINT Dostawca_ma_adres FOREIGN KEY (Id_adresu) REFERENCES Adresy (Id_adresu)
```

/

```
ALTER TABLE Punkt_obsługi_Godziny_otwarcia ADD CONSTRAINT Punkt_obsługi_posiada FOREIGN KEY (Id_Placowki) REFERENCES Punkty_obsługi (Id_Placowki)
```

/

```
ALTER TABLE Punkt_obsługi_Godziny_otwarcia ADD CONSTRAINT Godziny_otwarcia_posiada FOREIGN KEY (Id_godzin_otwarcia) REFERENCES Godziny_otwarci (Id_godzin_otwarcia)
```

/

```
ALTER TABLE Pojazdy ADD CONSTRAINT Pojazd_nalezy_do_magazynu FOREIGN KEY (Id_Placowki) REFERENCES Magazyny (Id_Placowki)
```

/

Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

Aby móc przetestować działanie bazy, wprowadzono do niej przykładowe dane.

```
INSERT INTO Adresy (miasto, kod_pocztowy, ulica, numer_domu, numer_mieszkania) VALUES ('Warszawa', '21-371', 'Generala', '1', '99');
INSERT INTO Adresy (miasto, kod_pocztowy, ulica, numer_domu, numer_mieszkania) VALUES ('Lodz', '21-000', 'Maklowa', '2', '87C');
INSERT INTO Adresy (miasto, kod_pocztowy, ulica, numer_domu, numer_mieszkania) VALUES ('Warszawa', '21-371', 'Gen. L. Okulickiego', '1', '99');
INSERT INTO Adresy (miasto, kod_pocztowy, ulica, numer_domu) VALUES ('Lwow', '01-374', 'Ewerowodzka', '93A');
INSERT INTO Adresy (miasto, kod_pocztowy, ulica, numer_domu, numer_mieszkania) VALUES ('Krakow', '22-222', 'Generala', '4', '54');
INSERT INTO Adresy (miasto, kod_pocztowy, ulica, numer_domu) VALUES ('Warszawa', '12-371', 'Gen. T. Bora', '3');
```

	ID_ADRESU	MIASTO	KOD_POCZTOWY	ULICA	NUMER_DOMU	NUMER_MIESZKANIA
1	64	Warszawa	21-371	Generala	1	99
2	66	Lodz	21-000	Maklowa	2	87C
3	68	Warszawa	21-371	Gen. L. Okulickiego	1	99
4	69	Lwow	01-374	Ewerowodzka	93A	(null)
5	70	Krakow	22-222	Generala	4	54
6	71	Warszawa	12-371	Gen. T. Bora	3	(null)

Przykłady poleceń:

- Wyszukanie adresów, które znajdują się w Warszawie.

```
SELECT * FROM Adresy WHERE Miasto = 'Warszawa';
```

	ID_ADRESU	MIASTO	KOD_POCZTOWY	ULICA	NUMER_DOMU	NUMER_MIESZKANIA
1	64	Warszawa	21-371	Generala	1	99
2	68	Warszawa	21-371	Gen. L. Okulickiego	1	99
3	71	Warszawa	12-371	Gen. T. Bora	3	(null)
4	81	Warszawa	67-677	Ryzowa	43d	20

- Wyszukanie osób z wynagrodzeniem większym niż 4000 złotych.

```
SELECT Imie, Nazwisko, Stanowisko, Wynagrodzenie FROM PRACOWNICY  
WHERE Wynagrodzenie > 4000;
```

	IMIE	NAZWISKO	STANOWISKO	WYNAGRODZENIE
1	Malek	Kozidroblowski	Administrator baz	4705
2	Ewelina	LaStrange	Ksiegowia	4320

- Informacje o adresach zamieszkania pracowników.

```
SELECT Imie, Nazwisko, Miasto, Ulica, adresy.numer_domu, adresy.numer_mieszkania FROM Pracownicy  
NATURAL JOIN Adresy;
```

	IMIE	NAZWISKO	MIASTO	ULICA	NUMER_DOMU	NUMER_MIESZKANIA
1	Jagienka	Anastazjowa	Warszawa	Gen. L. Okulickiego	1	99
2	Marek	Tracz	Krakow	Generala	4	54
3	Malek	Kozidroblowski	Warszawa	Gen. T. Bora	3	(null)
4	Ewelina	LaStrange	Warszawa	Ryzowa	43d	20

- Wyszukanie informacji na temat tego ile adresów w bazie jest z danego miasta.

```
SELECT Miasto, COUNT(*) AS Ilosc_adresow FROM Adresy GROUP BY Miasto;
```

	MIASTO	ILOSC_ADRESOW
1	Warszawa	4
2	Lodz	1
3	Lwow	1
4	Krakow	1

Bibliografia

- Prezentacje wykładowe i projektowe
- <https://wdx.pl/2020/05/29/klasyfikacja-i-rodzaje-magazynow/>