

***Dokument przygotował:***

***Łukasz Michowski***



***GIORGIO ULANI NUTRITION® jest wiodącym sklepem z odżywkami oraz suplementami diety dla sportowców.***

# **Baza danych**

## **Spis treści**

1. Zakres i cel projektu (opis założeń funkcjonalnych projektowanej bazy danych)
2. Definicja systemu
  - 2.1 Perspektywy użytkowników
3. Model konceptualny
  - 3.1 Definicja zbiorów encji określonych w projekcie (decyzje projektowe) oraz określenie atrybutów i ich dziedzin
  - 3.2 Ustalenie związków i ich typów między encjami
  - 3.3 Klucze kandydujące i główne (decyzje projektowe)
  - 3.4 Schemat ER na poziomie konceptualnym
  - 3.5 Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady
4. Model logiczny
  - 4.1 Charakterystyka modelu relacyjnego
  - 4.2 Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady
  - 4.3 Proces normalizacji – analiza i przykłady
  - 4.4 Schemat ER na poziomie modelu logicznego
  - 4.5 Więzy integralności
  - 4.6 Proces denormalizacji – analiza i przykłady
5. Faza fizyczna
  - 5.1 Skrypt SQL zakładający bazę danych oraz dobór indeksów
  - 5.2 Skrypt uzupełniający bazę danych
  - 5.3 Przykłady zapytań SQL odnoszących się do bazy danych
  - 5.4 Projekt transakcji i weryfikacja ich wykonalności

## 1. Założenia/ opis działalności biznesowej

Realizowany projekt dotyczy przedsiębiorstwa z branży suplementacyjnej. Przedsiębiorstwo to zajmuje się sprzedażą odżywek oraz suplementów. Siedziba sklepu mieści się w Warszawie, jednak sprzedaż odbywa się głównie przez internet. Jej oferta obejmuje różnorodne produkty krajowych jak i zagranicznych producentów. Na koszt produktu składa się łączny koszt zakupu towaru od producenta plus 23- procentowy podatek VAT oraz marża. Pracownicy opłacani są według stawek godzinowych w ramach umowy o pracę, otrzymują premię za nadgodziny. W swojej codziennej działalności przedsiębiorstwo to skupia się na promocji swojej oferty w postaci dostępnych dla klienta suplementów. Pracownicy pracują na dwie zmiany (6-14 oraz 14-22) według określonego grafika. Przy zakupie może być wystawiana faktura. Klient ma możliwość dokonywania transakcji na bazie gotówki lub karty kredytowej.

## 2. Definicja systemu

### 2.1 Zidentyfikowane perspektywy użytkowników:

- **Administrator**, mogący edytować strukturę bazy danych oraz mający dostęp do wszystkich danych; dostęp do perspektywy mają właściciele sklepu
- **Pracownik** – widzi swoje dane osobowe, historie wynagrodzenia oraz zadania wynikające z aktualnej pracy. Może podejrzeć także dane każdego klienta
- **Klient** - Ma wgląd we własne dane (gdy ma aktywne konto internetowe, w przeciwnym wypadku, nie ma dostępu bezpośredniego), posiada możliwość sprawdzenia historii zakupów

1. Podgląd danych personalnych pracowników **A**,
2. Podgląd danych placówek **A, P, K**
3. Modyfikacja/dodawanie/ usuwanie danych personalnych pracowników **A**,
4. Podgląd danych personalnych klientów **A, P**,
5. Modyfikacja/dodawanie/ usuwanie danych personalnych klientów **A, P**
6. Podgląd wynagrodzenia każdego z pracowników **A**,
7. Dodawanie/usuwanie/ modyfikacja wynagrodzenia każdego z **A**,
8. Modyfikacja/dodawanie/ usuwanie danych o placówkach oraz firmie **A**,
9. Modyfikacja/dodawanie / usuwanie danych produktów oraz ich cen **A, P**

10. Podgląd danych produktów **A, P, K**

11. Uprawnienia do modyfikacji struktury bazy danych **A,**

12. Modyfikacja/ dodawanie/usuwanie i podgląd zleceń **A,**

13. Modyfikacja/ dodawanie/usuwanie i podgląd danych o dostawcach **A, P**

### 3. Model konceptualny

3.1 Definicja zbiorów encji określonych w projekcie (decyzje projektowe)

#### Legenda:

PUI – Czy to pole jest kluczem głównym?

Attribute Name – nazwa atrybutu

Domain – określa domenę, jeśli występuje

Data Type – typ danych

<sup>1)</sup>Mandatory – Czy pole jest obowiązkowe?

<sup>2)</sup>Default - Czy pole jest domyślnie określone?

<sup>3)</sup>Rule – Czy w tym polu obowiązuje zasada?

Description – Opis atrybutu

**Sklep** - istnieje tylko jedna encja banku, a wszystkie atrybuty przedstawiono poniżej.

PUI	Attribute Name	Domain	Data Type	M <sup>1)</sup>	D <sup>2)</sup>	R <sup>3)</sup>	Description
YES	idSklepu		Integer	YES	NO	NO	Unikalny numer sklepu
NO	Nazwa		VarChar(50)	YES	NO	NO	Nazwa biura
NO	Data_zalozenia		Date	YES	NO	NO	Data założenia sklepu
NO	Wlasiciel		VarChar(30)	YES	NO	NO	Imie nazwisko i opis właściciela
NO	Powierzchnia		Number	NO	NO	NO	Powierzchnia sklepu w metrach kwadratowych

**Placówka** - istnieje wiele obiektów tej encji. Atrybuty w tabelce opisują poszczególne cechy tej encji

PUI	Attribute Name	Domain	Data Type	M <sup>1)</sup>	D <sup>2)</sup>	R <sup>3)</sup>	Description
YES	idPlacowki		Integer	YES	NO	NO	Unikalny numer placowki
NO	Adres		VarChar(50)	YES	NO	NO	Adres placowki
NO	data_zalozenia_placowki		Date	YES	NO	NO	Data zalozzenia placowki

**Dostawca** - istnieje wiele obiektów tej encji. Atrybuty w tabelce opisują poszczególne cechy tej encji.

PUI	Attribute Name	Domain	Data Type	M <sup>1)</sup>	D <sup>2)</sup>	R <sup>3)</sup>	Description
YES	idDostawcy		Integer	YES	NO	NO	Unikalny nr dostawcy
NO	Imie		VarChar(30)	YES	NO	NO	Imie dostawcy
NO	Nazwisko		VarChar(30)	YES	NO	NO	Nazwisko dostawcy
NO	Nazwa_firmy		VarChar(50)	YES	NO	NO	Nazwa firmy dostarczającej towar

**Klient** - istnieje wiele obiektów tej encji. Atrybuty w tabelce opisują poszczególne cechy tej encji.

PUI	Attribute Name	Domain	Data Type	M <sup>1)</sup>	D <sup>2)</sup>	R <sup>3)</sup>	Description
YES	idKlienta		Integer	YES	NO	NO	Unikalny numer klienta
NO	Imie		VarChar(30)	YES	NO	NO	Imie klienta
NO	Nazwisko		VarChar(30)	YES	NO	NO	Nazwisko klienta
NO	Plec	Plec	Bigint	NO	NO	NO	Plec klienta
NO	Data_urodzenia		Date	NO	NO	NO	Data urodzenia
NO	Nr_dokumentu		VarChar(15)	NO	NO	NO	Data urodzenia klienta
NO	Nr_telefonu		VarChar(12)	YES	NO	NO	Numer telefonu klienta
NO	e_mail		VarChar(40)	YES	NO	NO	Adres email klienta

**Produkt** - istnieje wiele obiektów tej encji. Atrybuty w tabelce opisują poszczególne cechy tej encji.

PUI	Attribute Name	Domain	Data Type	M <sup>1)</sup>	D <sup>2)</sup>	R <sup>3)</sup>	Description
YES	idProduktu		Integer	YES	NO	NO	Unikalny numer id
NO	Rodzaj_produk <u>t</u> u	Rodzaj_produk <u>t</u> u	VarChar(50)	YES	NO	NO	Definiuje jaki rodzaj reprezentaentuje produkt
NO	Cena		VarChar(5)	YES	NO	NO	Cena produktu
NO	Data_waznosci		Date	YES	NO	NO	Data ważności produktu

**Stanowisko** - istnieje wiele obiektów tej encji. Atrybuty w tabelce opisują poszczególne cechy tej encji.

PUI	Attribute Name	Domain	Data Type	M <sup>1)</sup>	D <sup>2)</sup>	R <sup>3)</sup>	Description
YES	idStanowiska		Integer	YES	NO	NO	Unikalny numer stanowiska
NO	Nazwa		VarChar(50)	YES	NO	NO	Nazwa zajmowanego przez pracownika stanowiska
NO	Opis		VarChar(500)	YES	NO	NO	Opis stanowiska

**Pracownik** - istnieje wiele obiektów tej encji. Atrybuty w tabelce opisują poszczególne cechy tej encji.

PUI	Attribute Name	Domain	Data Type	M <sup>1)</sup>	D <sup>2)</sup>	R <sup>3)</sup>	Description
YES	idPracownika		Integer	YES	NO	NO	Unikalny numer pracownika
NO	Imie		VarChar(30)	YES	NO	NO	Imię pracownika
NO	Nazwisko		VarChar(50)	YES	NO	NO	Nazwisko pracownika
NO	Data_zatrudnienia		Date	YES	NO	NO	Data zatrudnienia pracownika
NO	PESEL		VarChar(11)	NO	NO	NO	Unikalny numer pesel pracownika
NO	Plec	Plec	Bigint	YES	NO	NO	Płeć pracownika

**Wynagrodzenie** - istnieje wiele obiektów tej encji. Atrybuty w tabelce opisują poszczególne cechy tej encji.

PUI	Attribute Name	Domain	Data Type	M <sup>1)</sup>	D <sup>2)</sup>	R <sup>3)</sup>	Description
YES	idWynagrodzenia		Integer	YES	NO	NO	Unikalny numer wynagrodzenia
NO	Data		Date	YES	NO	NO	data wydania wynagrodzenia
NO	Kwota_podstawy		Float(6)	YES	NO	NO	Kwota podstawy wynagrodzenia
NO	Kwota_dodatku		Float(6)	NO	NO	NO	Kwota dodatku do wynagrodzenia

**Grafik** - istnieje wiele obiektów tej encji. Atrybuty w tabelce opisują poszczególne cechy tej encji.

PUI	Attribute Name	Domain	Data Type	M <sup>1)</sup>	D <sup>2)</sup>	R <sup>3)</sup>	Description
YES	idGrafiku		Integer	YES	NO	NO	Unikalny numer grafiku
NO	Data		Date	YES	NO	NO	Data obowiązywania grafiku
NO	Godzina_od		Character(5)	YES	NO	NO	Godzina rozpoczęcia pracy przez pracownika
NO	Godzina_do		Character(5)	YES	NO	NO	Godzina zakończenia pracy przez pracownika

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

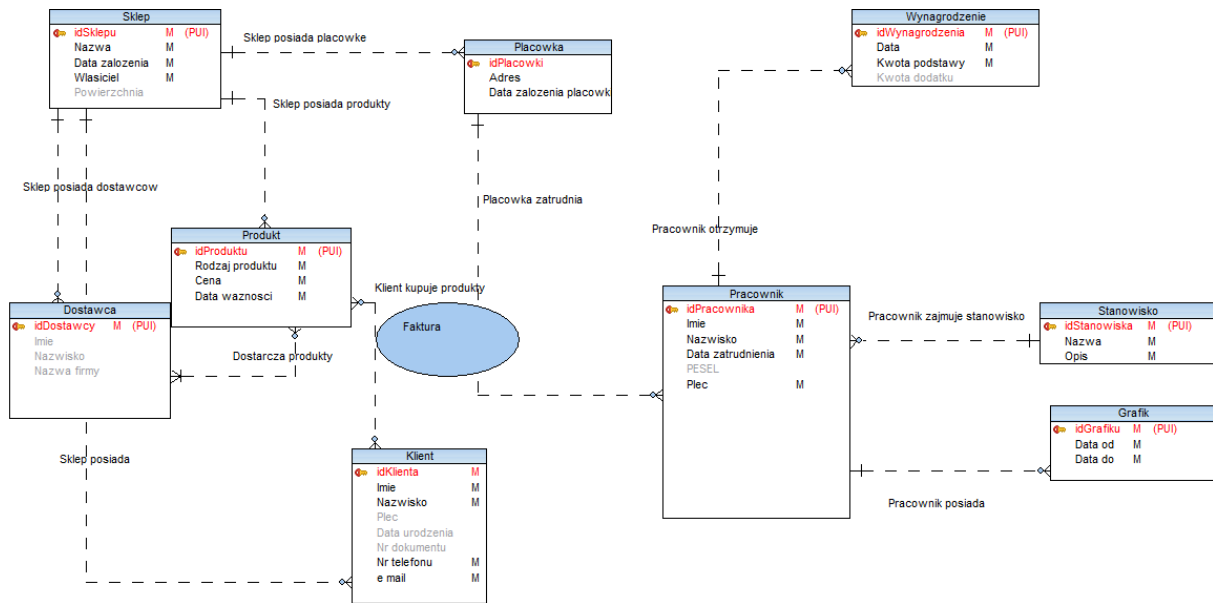
Większość związków jest typu 1:n jednak istnieją również związki wielu do wielu, ponieważ na przykład Klient może zakupić wiele produktów oraz jeden Produkt może być zakupiony przez wielu klientów. Widzimy tutaj konieczność stworzenia encji łączącej obie te tabele(prawdopodobnie będzie to Transakcja).

Relationship Name	Relationship Type	Relationship Between	Cardinality
Dostarcza_produkty	Non-Identifying	Dostawca - Produkt	1..n - 0..m
Klient_kupuje_produkty	Non-Identifying	Produkt - Klient	0..n - 0..m
Placowka_zatrudnia	Non-Identifying	Placowka - Pracownik	1..1 - 0..m
Pracownik_posiada	Non-Identifying	Pracownik - Grafik	1..1 - 0..m
Pracownik_otrzymuje	Non-Identifying	Pracownik - Wynagrodzenie	1..1 - 0..m
Pracownik_zajmuje_stanowisko	Non-Identifying	Stanowisko - Pracownik	1..1 - 0..m
Sklep_posiada	Non-Identifying	Sklep - Klient	1..1 - 0..m
Sklep_posiada_dostawcow	Non-Identifying	Sklep - Dostawca	1..1 - 0..m
Sklep_posiada_placowke	Non-Identifying	Sklep - Placowka	1..1 - 0..m
Sklep_posiada_produkty	Non-Identifying	Sklep - Produkt	1..1 - 0..m

### 3.3 Klucze kandydujące i główne (decyzje projektowe)

Entity Name	Primary key	Candidate key
Dostawca	idDostawcy	-
Grafik	idGrafiku	-
Klient	idKlienta	Imie, Nazwisko; Numer dokumentu
Placowka	idPlacowki	-
Pracownik	idPracownika	Imie, Nazwisko; PESEL
Produkt	idProduktu	-
Sklep	idSklepu	Nazwa
Stanowisko	idStanowiska	Nazwa
Wynagrodzenie	idWynagrodzenia	-

### 3.4 Schemat ER na poziomie konceptualnym



### 3.5 Pułapki szczelinowe i wachlarzowe

**Pułapka szczelinowa** występuje gdy model sugeruje istnienie związku pomiędzy zbiorami encji (klasami), ale nie istnieje ścieżka łącząca pewne wystąpienia tych encji (obiektów); pułapka ta może wystąpić, gdy w modelu znajduje się co najmniej jeden związek o minimalnej krotności zero, który jest elementem ścieżki pomiędzy powiązаныmi encjami (klasami).

W projekcie występuje ryzyko wystąpienia błędów wynikających z pułapek szczelinowych.

Pracownik jest zatrudniony w placówce, a nie w sklepie. Model sugeruje, że pracownik mógłby być zatrudniony w sklepie. Niektóre operacje na bazie mogą być błędnie wykonane.

**Pułapka wachlarzowa** występuje w sytuacji, gdy model przedstawia związek pomiędzy pewnymi zbiorami encji (klasami), ale wynikające z tego ścieżki pomiędzy wystąpieniami encji (obiektami) nie są jednoznaczne; pułapka taka może wystąpić, gdy co najmniej dwa związki typu 1:\* wychodzą z tej samej encji (klasy).

W moim projekcie nie stwierdziłem występowania pułapek wachlarzowych.



## 4 Model logiczny

### 4.1 Charakterystyka modelu relacyjnego

W celu otrzymania modelu logicznego należało usunąć między innymi związki wiele do wielu. Nazwy encji zostały zmienione na liczbę mnogą, aby zachowały unikatowość

na poziomie modelu konceptualnego i umożliwiły odróżnienie relacji od encji. Każda z encji zawierała już klucz główny i różne atrybuty (niegłówne). Nazwy atrybutów nie potrzebowały wielu zmian gdyż już na poziomie konceptualnym zostały zaprojektowane jako pola segmentowe. Te atrybuty, które nie zostały wykluczone, zostały przekształcone na nowe encje.

### 4.2 Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

Usunięcie pól wielowartościowych:

Atrybut adres został zastąpiony odrębną encją.

Usunięcie związki wiele do wielu:

Związek m:n pomiędzy Klientem i produktem został zastąpiony poprzez dodanie relacji pośredniej w postaci Transakcji.

### 4.3 Proces normalizacji

Relacja jest w pierwszej postaci normalnej, jeśli:

- Każda wartość atrybutu w każdej krotce tej relacji jest wartością elementarną, czyli nierozkładalną
- Nie ma powtarzających się grup

Przykład: W celu normalizacji pola segmentowe adres oraz właściciel zostały przeniesione do osobnej tabeli.

Relacja jest w drugiej postaci normalnej, jeśli:

- Jest w 1PN
- Każdy atrybut tej relacji nie wchodzący w skład żadnego klucza potencjalnego jest w pełni funkcyjnie zależny od wszystkich kluczy potencjalnych relacji
- Każdy atrybut nie wchodzący w skład klucza zależy od klucza a nie od jego części

Wszystkie klucze potencjalne w mojej bazie są kluczami prostymi.

Relacja jest w trzeciej postaci normalnej, jeśli:

- Jest w 2PN
- Nie jest relacją przechodnią (transytywną)

Relacja nie była by w trzeciej postaci normalnej, jeśli np. w tabeli **Transakcje** do atrybutów: **Cena netto** i **Cena brutto** dodał atrybut **Tara**, ponieważ tarę można określić na podstawie ceny netto i brutto.

#### 4.4 Więzy integralności wynikające z modelu relacyjnego

W bazie na tym etapie nie występują pola segmentowe. Wszystkie klucze są unikalne (UNIQUE). Większość pól jest obowiązkowa (NOT NULL).

#### 4.5 Proces denormalizacji

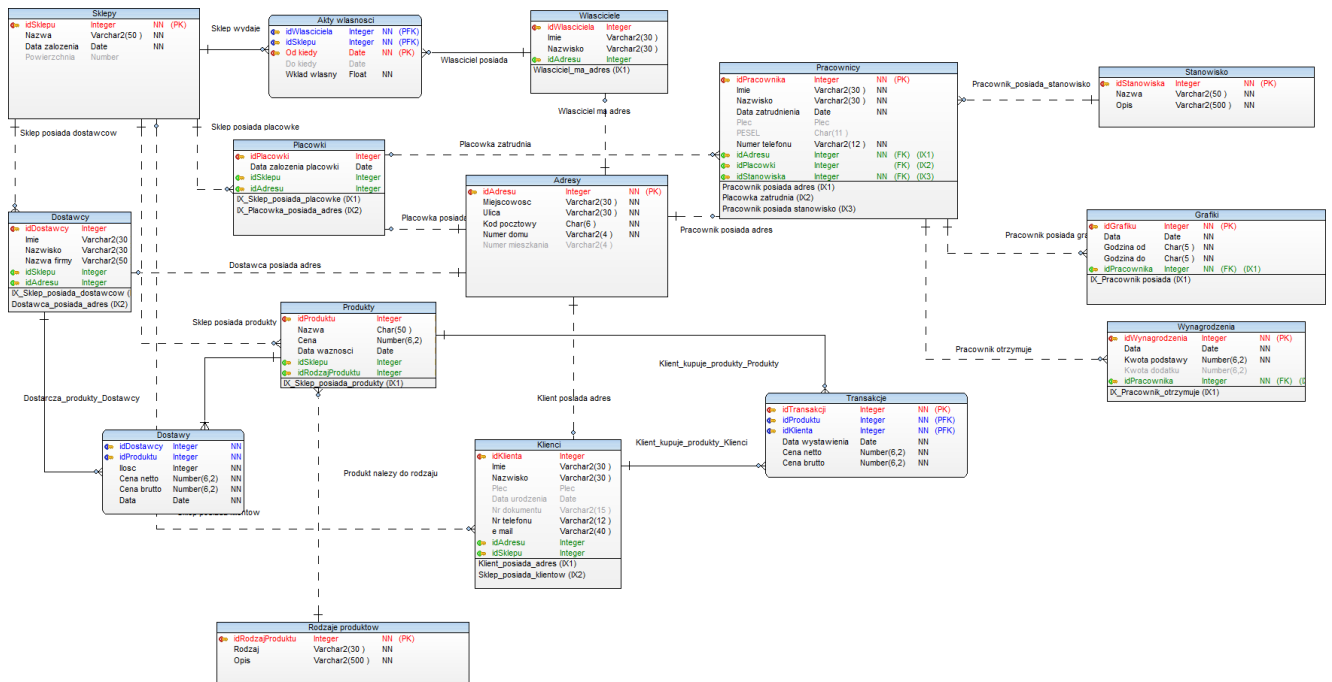
**Denormalizacja bazy** jest to wprowadzenie kontrolowanej nadmierności do **bazy danych** w celu przyspieszenia wykonywania na niej operacji (np. obsługiwanie zapytań); dzięki **denormalizacji bazy** unika się kosztownych operacji połączeń tabel.

Zaletą: szybki dostęp do informacji, bez konieczności wykonywania np 4 joinów.

Wadą: redundacja danych oraz utrudnione utrzymywanie spójności.

W moim projekcie nie stwierdziłem możliwości denormalizacji bazy danych, która w znaczący sposób przyspieszyłaby do niej dostęp.

## 4.6 Schemat ER na poziomie modelu logicznego



## 5. Faza fizyczna

### 5.1 Skrypt SQL tworzący bazę danych oraz dobór indeksów

-- Table Sklepy

```
CREATE TABLE Sklepy(
  idSklepu Integer NOT NULL,
  Nazwa Varchar2(50) NOT NULL,
  Data_zalozenia Date NOT NULL,
  Powierzchnia Number
)
```

-- Add keys for table Sklepy

```
ALTER TABLE Sklepy ADD CONSTRAINT sklep_PK PRIMARY KEY (idSklepu)
```

-- Table Placowki

```
CREATE TABLE Placowki(
  idPlacowki Integer NOT NULL,
  data_zalozenia_placowki Date NOT NULL,
  idSklepu Integer NOT NULL,
  idAdresu Integer NOT NULL
)
```

-- Create indexes for table Placowki

```
CREATE INDEX IX_Sklep_posiada_placowke ON Placowki (idSklepu)
CREATE INDEX IX_Placowka_posiada_adres ON Placowki (idAdresu)
```

-- Add keys for table Placowki

```
ALTER TABLE Placowki ADD CONSTRAINT Placowki_PK PRIMARY KEY (idPlacowki)
-- Table Pracownicy
```

```
CREATE TABLE Pracownicy(
  idPracownika Integer NOT NULL,
  Imie Varchar2(30 ) NOT NULL,
  Nazwisko Varchar2(30 ) NOT NULL,
  Data_zatrudnienia Date NOT NULL,
  Plec Char(2 )
    CHECK (Plec IN ('M','K')),
  PESEL Char(11 ),
  Numer_telefonu Varchar2(12 ) NOT NULL,
  idAdresu Integer NOT NULL,
  idPlacowki Integer,
  idStanowiska Integer NOT NULL
)
```

-- Create indexes for table Pracownicy

```
CREATE INDEX Pracownik_posiada_adres ON Pracownicy (idAdresu)
CREATE INDEX Placowka_zatrudnia ON Pracownicy (idPlacowki)
CREATE INDEX Pracownik_posiada_stanowisko ON Pracownicy (idStanowiska)
```

-- Add keys for table Pracownicy

```
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownicy_PK PRIMARY KEY (idPracownika)
```

-- Table Produkty

```
CREATE TABLE Produkty(
  idProduktu Integer NOT NULL,
  Nazwa Char(50 ) NOT NULL,
  Cena Number(6,2) NOT NULL,
  Data_waznosci Date NOT NULL,
  idSklepu Integer NOT NULL,
  idRodzajProduktu Integer NOT NULL
)
```

-- Create indexes for table Produkty

```
CREATE INDEX IX_Sklep_posiada_produkty ON Produkty (idSklepu)
CREATE INDEX Produkt_nalezy_do_rodzaju ON Produkty (idRodzajProduktu)
```

-- Add keys for table Produkty

```
ALTER TABLE Produkty ADD CONSTRAINT Produkty_PK PRIMARY KEY (idProduktu)
```

-- Table and Columns comments section

COMMENT ON COLUMN Produkty.Nazwa IS 'Nazwa produktu'

-- Table Klienci

```
CREATE TABLE Klienci(  
  idKlienta Integer NOT NULL,  
  Imie Varchar2(30) NOT NULL,  
  Nazwisko Varchar2(30) NOT NULL,  
  Plec Char(2)  
    CHECK (Plec IN ('M','K')),  
  Data_urodzenia Date,  
  Nr_dokumentu Varchar2(15),  
  Nr_telefonu Varchar2(12) NOT NULL,  
  e_mail Varchar2(40) NOT NULL,  
  idAdresu Integer NOT NULL,  
  idSklepu Integer  
)
```

-- Create indexes for table Klienci

```
CREATE INDEX Klient_posiada_adres ON Klienci (idAdresu)  
CREATE INDEX Sklep_posiada_klientow ON Klienci (idSklepu)
```

-- Add keys for table Klienci

```
ALTER TABLE Klienci ADD CONSTRAINT Klienci_PK PRIMARY KEY (idKlienta)
```

-- Table Dostawcy

```
CREATE TABLE Dostawcy(  
  idDostawcy Integer NOT NULL,  
  Imie Varchar2(30) NOT NULL,  
  Nazwisko Varchar2(30) NOT NULL,  
  Nazwa_firmy Varchar2(50) NOT NULL,  
  idSklepu Integer NOT NULL,  
  idAdresu Integer NOT NULL  
)
```

-- Create indexes for table Dostawcy

```
CREATE INDEX IX_Sklep_posiada_dostawcow ON Dostawcy (idSklepu)  
CREATE INDEX Dostawca_posiada_adres ON Dostawcy (idAdresu)
```

-- Add keys for table Dostawcy

```
ALTER TABLE Dostawcy ADD CONSTRAINT Dostawcy_PK PRIMARY KEY (idDostawcy)
```

-- Table Wynagrodzenia

```
CREATE TABLE Wynagrodzenia(  
  idWynagrodzenia Integer NOT NULL,  
  Data Date NOT NULL,
```

```
Kwota_podstawy Float NOT NULL,  
Kwota_dodatku Float,  
idPracownika Integer NOT NULL  
)
```

```
-- Create indexes for table Wynagrodzenia
```

```
CREATE INDEX IX_Pracownik_otrzymuje ON Wynagrodzenia (idPracownika)
```

```
-- Add keys for table Wynagrodzenia
```

```
ALTER TABLE Wynagrodzenia ADD CONSTRAINT Wynagrodzenia_PK PRIMARY KEY  
(idWynagrodzenia)
```

```
-- Table Grafiki
```

```
CREATE TABLE Grafiki(  
idGrafiku Integer NOT NULL,  
Data Date NOT NULL,  
Godzina_od Char(5) NOT NULL,  
Godzina_do Char(5) NOT NULL,  
idPracownika Integer NOT NULL  
)
```

```
-- Create indexes for table Grafiki
```

```
CREATE INDEX IX_Pracownik_posiada ON Grafiki (idPracownika)
```

```
-- Add keys for table Grafiki
```

```
ALTER TABLE Grafiki ADD CONSTRAINT Grafiki_PK PRIMARY KEY (idGrafiku)
```

```
-- Table and Columns comments section
```

```
COMMENT ON COLUMN Grafiki.Data IS 'Data grafiku'
```

```
-- Table Adresy
```

```
CREATE TABLE Adresy(  
idAdresu Integer NOT NULL,  
Miejscowosc Varchar2(30) NOT NULL,  
Ulica Varchar2(30) NOT NULL,  
Kod_pocztowy Char(6) NOT NULL,  
Numer_domu Varchar2(4) NOT NULL,  
Numer_mieszkania Varchar2(4)  
)
```

```
-- Add keys for table Adresy
```

```
ALTER TABLE Adresy ADD CONSTRAINT Adres_PK PRIMARY KEY (idAdresu)
```

```
-- Table and Columns comments section
```

```
COMMENT ON COLUMN Adresy.Numer_mieszkania IS 'Numer mieszkania'
```

-- Table Dostawy

```
CREATE TABLE Dostawy(  
  idDostawcy Integer NOT NULL,  
  idProduktu Integer NOT NULL,  
  Ilosc Integer NOT NULL,  
  Cena_netto Float NOT NULL,  
  Cena_brutto Float NOT NULL,  
  Data Date NOT NULL  
)
```

-- Table and Columns comments section

```
COMMENT ON COLUMN Dostawy.Data IS 'Data dostawy'
```

-- Table Transakcje

```
CREATE TABLE Transakcje(  
  idTransakcji Integer NOT NULL,  
  idProduktu Integer NOT NULL,  
  idKlienta Integer NOT NULL,  
  Data_wystawienia Date NOT NULL,  
  Cena_netto Number(6,2) NOT NULL,  
  Cena_brutto Number(6,2) NOT NULL  
)
```

-- Table Wlasciciele

```
CREATE TABLE Wlasciciele(  
  idWlasciciela Integer NOT NULL,  
  Imie Varchar2(30) NOT NULL,  
  Nazwisko Varchar2(30) NOT NULL,  
  idAdresu Integer NOT NULL  
)
```

-- Create indexes for table Wlasciciele

```
CREATE INDEX Wlasciciel_ma_adres ON Wlasciciele (idAdresu)
```

-- Add keys for table Wlasciciele

```
ALTER TABLE Wlasciciele ADD CONSTRAINT PK_Wlasciciele PRIMARY KEY (idWlasciciela)
```

-- Table Akty\_wlasnosci

```
CREATE TABLE Akty_wlasnosci(  
  idWlasciciela Integer NOT NULL,  
  idSklepu Integer NOT NULL,  
  Od_kiedy Date NOT NULL,  
  Do_kiedy Date,  
  Wklad_wlasny Float NOT NULL  
)
```

-- Add keys for table Akty\_wlasnosci

```
ALTER TABLE Akty_wlasnosci ADD CONSTRAINT PK_Akty_wlasnosci PRIMARY KEY
(idWlasciciela,idSklepu,Od_kiedy)
```

-- Table Rodzaje\_produkow

```
CREATE TABLE Rodzaje_produkow(
  idRodzajProduktu Integer NOT NULL,
  Rodzaj Varchar2(30 ) NOT NULL,
  Opis Varchar2(500 ) NOT NULL
)
```

-- Add keys for table Rodzaje\_produkow

```
ALTER TABLE Rodzaje_produkow ADD CONSTRAINT PK_Rodzaje_produkow PRIMARY KEY
(idRodzajProduktu)
```

-- Table Stanowisko

```
CREATE TABLE Stanowisko(
  idStanowiska Integer NOT NULL,
  Nazwa Varchar2(50 ) NOT NULL,
  Opis Varchar2(500 ) NOT NULL
)
```

-- Add keys for table Stanowisko

```
ALTER TABLE Stanowisko ADD CONSTRAINT PK_Stalowisko PRIMARY KEY (idStanowiska)
```

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

```
ALTER TABLE Placowki ADD CONSTRAINT Sklep_posiada_placowke FOREIGN KEY (idSklepu)
REFERENCES Sklepy (idSklepu)
```

```
ALTER TABLE Dostawcy ADD CONSTRAINT Sklep_posiada_dostawcow FOREIGN KEY (idSklepu)
REFERENCES Sklepy (idSklepu)
```

```
ALTER TABLE Produkty ADD CONSTRAINT Sklep_posiada_produkty FOREIGN KEY (idSklepu)
REFERENCES Sklepy (idSklepu)
```

```
ALTER TABLE Wynagrodzenia ADD CONSTRAINT Pracownik_otrzymuje FOREIGN KEY
(idPracownika) REFERENCES Pracownicy (idPracownika)
```

```
ALTER TABLE Grafiki ADD CONSTRAINT Pracownik_posiada_grafik FOREIGN KEY (idPracownika)
REFERENCES Pracownicy (idPracownika)
```

```
ALTER TABLE Placowki ADD CONSTRAINT Placowka_posiada_adres FOREIGN KEY (idAdresu)
REFERENCES Adresy (idAdresu)
```

```
ALTER TABLE Dostawcy ADD CONSTRAINT Dostawca_posiada_adres FOREIGN KEY (idAdresu)
REFERENCES Adresy (idAdresu)
```

```
ALTER TABLE Klienci ADD CONSTRAINT Klient_posiada_adres FOREIGN KEY (idAdresu)
REFERENCES Adresy (idAdresu)
```

```
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_posiada_adres FOREIGN KEY
(idAdresu) REFERENCES Adresy (idAdresu)
```



```

ALTER TABLE Akty_wlasnosci ADD CONSTRAINT Wlasciciel_posiada FOREIGN KEY
(idWlasciciela) REFERENCES Wlasciciele (idWlasciciela)
ALTER TABLE Akty_wlasnosci ADD CONSTRAINT Sklep_wydaje FOREIGN KEY (idSklepu)
REFERENCES Sklepy (idSklepu)
ALTER TABLE Klienci ADD CONSTRAINT Sklep_posiada_klientow FOREIGN KEY (idSklepu)
REFERENCES Sklepy (idSklepu)
ALTER TABLE Wlasciciele ADD CONSTRAINT Wlasciciel_ma_adres FOREIGN KEY (idAdresu)
REFERENCES Adresy (idAdresu)
ALTER TABLE Pracownicy ADD CONSTRAINT Placowka_zatrudnia FOREIGN KEY (idPlacowki)
REFERENCES Placowki (idPlacowki)
ALTER TABLE Produkty ADD CONSTRAINT Produkt_nalezy_do_rodzaju FOREIGN KEY
(idRodzajProduktu) REFERENCES Rodzaje_produkow (idRodzajProduktu)
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_posiada_stanowisko FOREIGN KEY
(idStanowiska) REFERENCES Stanowisko (idStanowiska)

```

## 5.2 Skrypt uzupełniający bazę danych

```

INSERT INTO Sklepy VALUES(1,'Sklep z odzywkami',TO_DATE('2003/07/09','yyyy/mm/dd'),null);

```

```

INSERT INTO Adresy VALUES(1, 'Warszawa', 'Pulawska', '11-111', '51', null);
INSERT INTO Adresy VALUES(2, 'Radom', 'Krakowska', '22-221', '11', '3');
INSERT INTO Adresy VALUES(3, 'Warszawa', 'Mazowiecka', '11-221', '51', '21');
INSERT INTO Adresy VALUES(4, 'Warszawa', 'Braci Wagow', '55-111', '12', null);
INSERT INTO Adresy VALUES(5, 'Krakow', 'Kaczynskiego', '05-121', '15', null);
INSERT INTO Adresy VALUES(6, 'Lublin', 'Pulawska', '65-911', '151', null);
INSERT INTO Adresy VALUES(7, 'Konin', 'Jana Pawla II', '75-191', '199', null);
INSERT INTO Adresy VALUES(8, 'Warszawa', 'Stryjenskich', '99-221', '510', null);
INSERT INTO Adresy VALUES(9, 'Warszawa', 'Nowoursynowska', '21-765', '451', '921');
INSERT INTO Adresy VALUES(10, 'Gdynia', 'Marina', '01-765', '241', '1');

```

```

INSERT INTO Wlasciciele VALUES(1,'Filip','Kulon',3);

```

```

INSERT INTO Akty_wlasnosci VALUES(1,1,TO_DATE('2003/07/09','yyyy/mm/dd'),null,100000);

```

```

INSERT INTO Dostawcy VALUES(1, 'Jan', 'Matejko', 'KFD', 1, 2);
INSERT INTO Dostawcy VALUES(2, 'Wojciech', 'Kossak', 'Trec', 1, 10);

```

```

INSERT INTO Placowki VALUES(1,TO_DATE('2003/07/09','yyyy/mm/dd'),1,1);

```

```

INSERT INTO Stanowiska VALUES(1, 'Kasjer', 'Osoba na stanowisku kasjera zajmuje sie
sprzedarza produktow, posiada bezposreni kontakt z kolientem w placowce sklepu');
INSERT INTO Pracownicy
VALUES(1,'Michal','Owczarzak',TO_DATE('2004/07/09','yyyy/mm/dd'),'M','72012493385','486584
24782',4,1,1);
INSERT INTO Grafiki VALUES(1,TO_DATE('2018/12/17','yyyy/mm/dd'),'08.00', '16.00', 1);

```

```

INSERT INTO Wynagrodzenia VALUES(1,TO_DATE('2004/08/09', 'yyyy/mm/dd'),'3000','200',1);

INSERT INTO Klienci VALUES(1,'Krystian','Kapala','M',null,null,'987654321','krystian2@o2.pl',5,1);
INSERT INTO Klienci VALUES(2,'Jan','Nowak','M',null,null,'127654321','jannowak@o2.pl',6,1);
INSERT INTO Klienci VALUES(3,'Zbigniew','Kloc','M',null,null,'557654321','zbychuk69@o2.pl',7,1);

INSERT INTO Rodzaje_produkow VALUES(1,'Bialko','Odzywki z wysoka zawartoscia bialka w skladzie');
INSERT INTO Rodzaje_produkow VALUES(2,'Kreatyna','Odzywki z wysoka zawartoscia kreatyny w skladzie');
INSERT INTO Rodzaje_produkow VALUES(3,'Witaminy','Odzywki z wysoka zawartoscia witamin w skladzie');

INSERT INTO Produkty VALUES(1,'Dobre Whey','59,99',TO_DATE('2019/07/09', 'yyyy/mm/dd'),1,1);
INSERT INTO Produkty VALUES(2,'Dobra kreatyna','39,99',TO_DATE('2019/03/19', 'yyyy/mm/dd'),1,2);
INSERT INTO Produkty VALUES(3,'Trec Whey','89,99',TO_DATE('2019/04/29', 'yyyy/mm/dd'),1,1);

INSERT INTO Dostawy VALUES(1,1,1,'10,00','12,30',TO_DATE('2016/07/09', 'yyyy/mm/dd'));
INSERT INTO Dostawy VALUES(2,2,1,'100,00','123,00',TO_DATE('2016/07/04', 'yyyy/mm/dd'));
INSERT INTO Dostawy VALUES(3,3,1,'20,00','24,60',TO_DATE('2016/07/06', 'yyyy/mm/dd'));

```

### 5.3 Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

Podgląd pracowników którzy zajmują stanowisko Kasjera:

```

SELECT p.idPracownika, p.imie, p.nazwisko FROM Pracownicy p
JOIN Stanowiska s ON s.idStanowiska = p.idStanowiska
WHERE s.Nazwa = "Kasjer";

```

Podgląd wszystkich klientów o nazwisku Nowak:

```

SELECT * FROM Klienci k WHERE k.Nazwisko = "Nowak";

```

Podgląd pracowników pracujących 17 grudnia 2018 roku:

```

SELECT p.idPracownika, p.imie, p.nazwisko FROM Pracownicy p
JOIN Grafiki g ON p.idPracownika = g.idPracownika
WHERE g.Data= TO_DATE('2018/12/17', 'yyyy/mm/dd');

```

#### 5.4 Projekt transakcji i weryfikacja ich wykonalności

Transakcja	Potrzebne zasoby	Czy wykonalne?
Podgląd danych personalnych pracowników	Pracownicy, Adresy	<b>TAK</b>
Podgląd danych placówek	Placówki, Adresy	<b>TAK</b>
Modyfikacja/dodawanie/ usuwanie danych personalnych pracowników	Pracownicy, Adresy, Stanowiska	<b>TAK</b>
Podgląd danych personalnych klientów	Klienci, Adresy	<b>TAK</b>
Modyfikacja/dodawanie/ usuwanie danych personalnych klientów	Klienci, Adresy	<b>TAK</b>
Podgląd wynagrodzenia każdego z pracowników	Pracownicy, Wynagrodzenia	<b>TAK</b>
Dodawanie/usuwanie/ modyfikacja wynagrodzenia każdego z pracowników	Wynagrodzenia, Pracownicy	<b>TAK</b>
Modyfikacja/dodawanie/ usuwanie danych o placówkach oraz firmie	Placówki	<b>TAK</b>
Modyfikacja/dodawanie / usuwanie danych produktów oraz ich cen	Produkty, Dostawcy, Dostawy	<b>TAK</b>
Podgląd danych produktów	Produkty	<b>TAK</b>
Modyfikacja/ dodawanie/usuwanie i podgląd danych o dostawcach	Dostawcy	<b>TAK</b>