

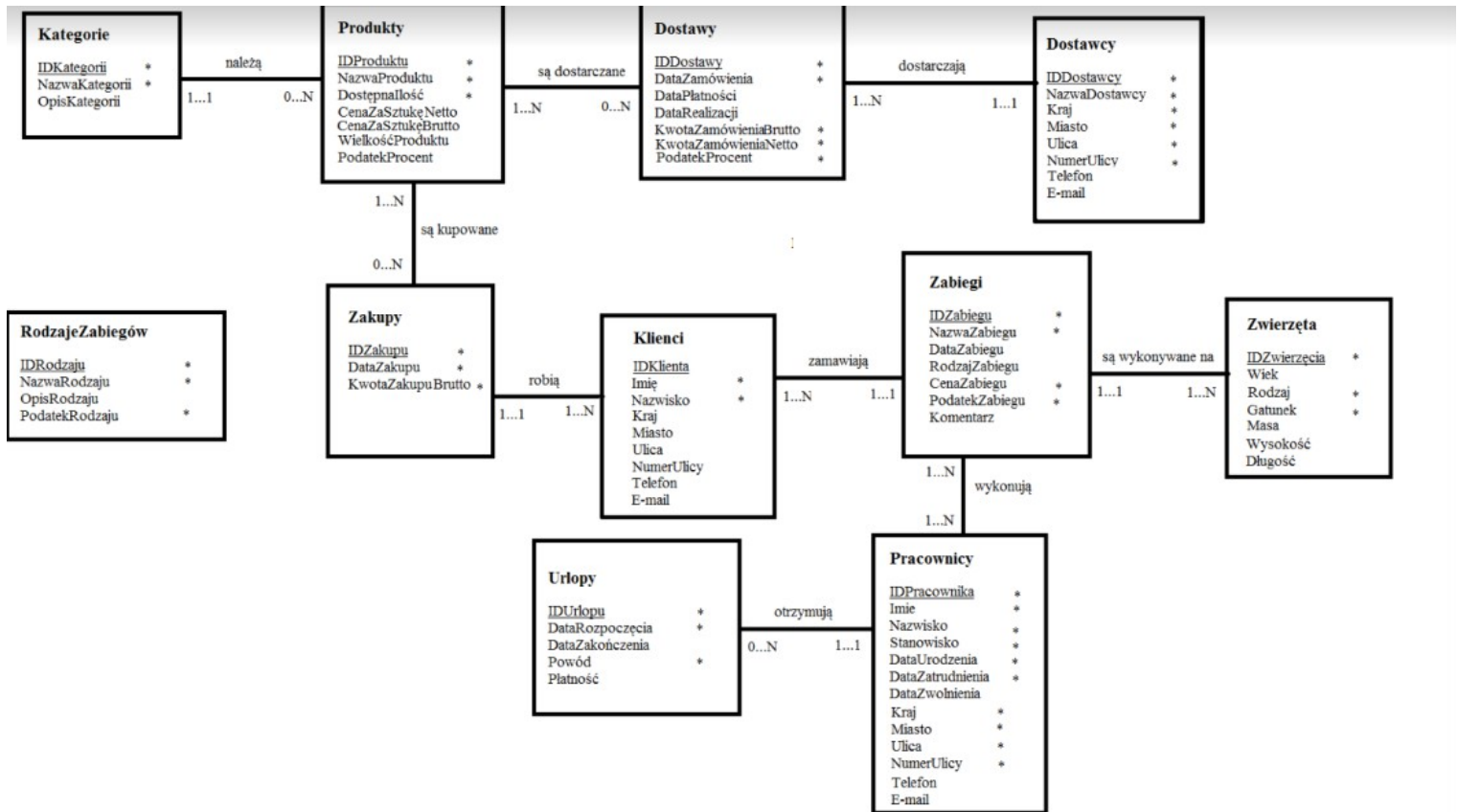
Magiera Szymon grupa 1B
Baza danych – mała przychodnia weterynaryjna
(45)
Zadanie 14 – projekt

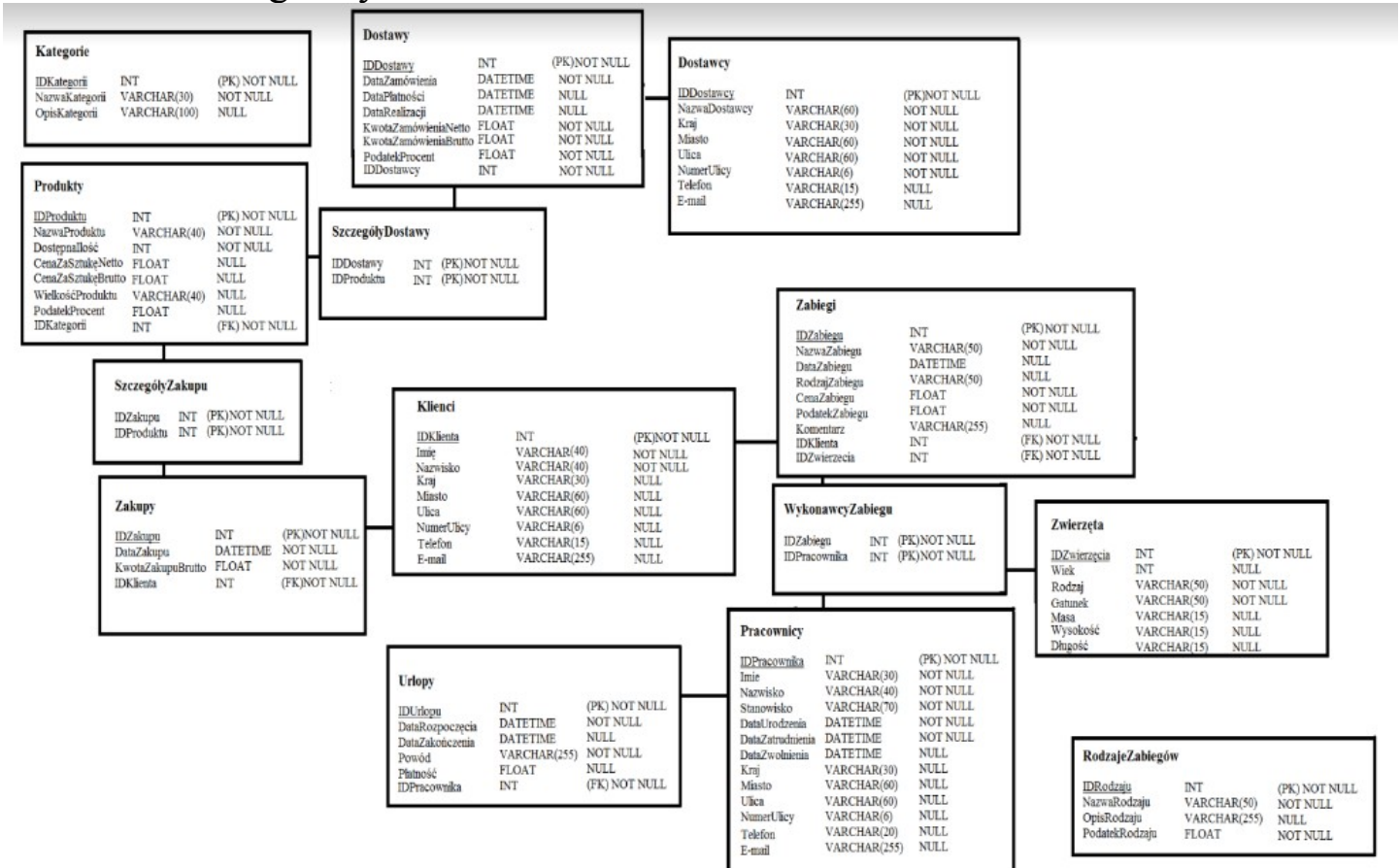
1.Cel projektu:

2.Wymagania projektu:

Możliwość dodania danych klientów takich jak imię I nazwisko, dane kontaktowe	10	
Możliwość dodania oferty produktów dostępnych w placówce do bazy danych	10	
Obsługa danych pracowników pracujących w placówce	10	Wiek pracowników zatrudnionych min. 18 lat
Możliwość dodania I edycji danych na temat leczonych zwierząt, ich zabiegów I zakupionych przez właścicieli produktów	10	Uwzględnienie podatków na poszczególne produkty
Dodawanie I edycja danych dotyczących dostaw I powiązanych z nimi dostawców	8	
Dodawanie I edycja danych na temat urlopów branych przez pracowników	10	
Zarządzanie danymi dotyczącymi zabiegów przeprowadzonych w placówce	7	Uwzględnienie podatków na poszczególne zabiegi
Przechowywanie danych opisujących zakupy dokonane przez klientów	7	
Dane wprowadzane do bazy danych muszą być odpowiedniego, wcześniej ustalonego typu	5	
Daty muszą posiadać logiczny sens	4	

2. DZE:





4. Normalizacja:

W modelu logicznym powyżej wszystkie tabele są w 3 postaci normalnej, z wyjątkiem tabel Dostawy I Produkty, w których ceny brutto są zależne od ceny netto I podatku, więc atrybuty CenaZaSztukęBrutto z tabeli Produkty I KwotaZamówieniaBrutto z tabeli Dostawy można usunąć.

```

graph TD
    Kategorie --> Produkty
    Produkty --> SzczegółyZakupu
    SzczegółyZakupu --> Zakupy
    Zakupy --> Klienci
    Klienci --> Dostawy
    Dostawy --> SzczegółyDostawy
    SzczegółyDostawy --> Zabiegi
    Zabiegi --> WykonawcyZabiegu
    WykonawcyZabiegu --> Pracownicy
    Pracownicy --> Zwierzęta
    Zwierzęta --> RodzajeZabiegów
  
```

Kategorie

IDKategori	INT	(PK) NOT NULL
NazwaKategori	VARCHAR(30)	NOT NULL
OpisKategori	VARCHAR(100)	NULL

Produkty

IDProduktu	INT	(PK) NOT NULL
NazwaProduktu	VARCHAR(40)	NOT NULL
Dostępnalność	INT	NOT NULL
CenaZamówieniaNetto	FLOAT	NULL
WielkośćProduktu	VARCHAR(40)	NULL
PodatekPercent	FLOAT	NULL
IDKategori	INT	(FK) NOT NULL

SzczegółyZakupu

IDZakupu	INT	(PK) NOT NULL
IDProduktu	INT	(FK) NOT NULL

Zakupy

IDZakupu	INT	(PK) NOT NULL
DataZakupu	DATETIME	NOT NULL
KwotaZakupuBrutto	FLOAT	NOT NULL
IDKlienta	INT	(FK) NOT NULL

Dostawy

IDDostawy	INT	(PK) NOT NULL
DataZamówienia	DATETIME	NOT NULL
DataPlatności	DATETIME	NULL
DataRealizacji	DATETIME	NULL
KwotaZamówieniaNetto	FLOAT	NOT NULL
PodatekPercent	FLOAT	NOT NULL
IDDostawcy	INT	NOT NULL

SzczegółyDostawy

IDDostawy	INT	(PK) NOT NULL
IDProduktu	INT	(FK) NOT NULL

Klienci

IDKlienta	INT	(PK) NOT NULL
Imię	VARCHAR(40)	NOT NULL
Nazwisko	VARCHAR(40)	NOT NULL
Kraj	VARCHAR(30)	NULL
Miasto	VARCHAR(60)	NULL
Ulica	VARCHAR(60)	NULL
NumerUlicy	VARCHAR(6)	NULL
Telefon	VARCHAR(15)	NULL
E-mail	VARCHAR(255)	NULL

Zabiegi

IDZabiegu	INT	(PK) NOT NULL
NazwaZabiegu	VARCHAR(50)	NOT NULL
DataZabiegu	DATETIME	NULL
RodzajZabiegu	VARCHAR(50)	NULL
CenaZabiegu	FLOAT	NOT NULL
PodatekZabiegu	FLOAT	NOT NULL
Komentarz	VARCHAR(255)	NULL
IDKlienta	INT	(FK) NOT NULL
IDZwierzęcia	INT	(FK) NOT NULL

WykonawcyZabiegu

IDZabiegu	INT	(PK) NOT NULL
IDPracownika	INT	(FK) NOT NULL

Pracownicy

IDPracownika	INT	(PK) NOT NULL
Imię	VARCHAR(30)	NOT NULL
Nazwisko	VARCHAR(40)	NOT NULL
Stanowisko	VARCHAR(70)	NOT NULL
DataUrodzenia	DATETIME	NOT NULL
DataZatrudnienia	DATETIME	NOT NULL
DataZwolnienia	DATETIME	NULL
Kraj	VARCHAR(30)	NULL
Miasto	VARCHAR(60)	NULL
Ulica	VARCHAR(60)	NULL
NumerUlicy	VARCHAR(6)	NULL
Telefon	VARCHAR(20)	NULL
E-mail	VARCHAR(255)	NULL

Zwierzęta

IDZwierzęcia	INT	(PK) NOT NULL
Wiek	INT	NULL
Rodzaj	VARCHAR(50)	NOT NULL
Gatunek	VARCHAR(50)	NOT NULL
Masa	VARCHAR(15)	NULL
Wysokość	VARCHAR(15)	NULL
Długość	VARCHAR(15)	NULL

RodzajeZabiegów

IDRodzaju	INT	(PK) NOT NULL
NazwaRodzaju	VARCHAR(50)	NOT NULL
OpisRodzaju	VARCHAR(255)	NULL
PodatekRodzaju	FLOAT	NOT NULL

IDKategorii – INT, NOT NULL

NazwaKategorii – VARCHAR(30), NOT NULL

OpisKategorii – VARCHAR(100)

IDProduktu – INT, NOT NULL

NazwaProduktu – VARCHAR(40), NOT NULL

Dostępna ilość – INT, NOT NULL, @x >= 0

CenaZaSztukęNetto – FLOAT, @x >= 0

WielkośćProduktu – VARCHAR(40)

PodatekProcent – FLOAT, @x>=0

IDKategorii – INT, NOT NULL

IDDostawy – INT, NOT NULL

DataZamówienia – DATETIME, @x <= NOW()

DataPłatności – DATETIME, @x <= NOW(), @x >= DataZamówienia

DataRealizacji – DATETIME, @x <= NOW(), @x >= DataZamówienia, @x >= DataPłatności

KwotaZamówieniaNetto – FLOAT, NOT NULL, @x >= 0

PodatekProcent – FLOAT, NOT NULL, @x >= 0

IDDostawcy – INT, NOT NULL

Dostawcy

IDDostawcy – INT, NOT NULL
NazwaDostawcy – VARCHAR(60), NOT NULL
Kraj – VARCHAR(30), NOT NULL
Miasto – VARCHAR(60), NOT NULL
Ulica – VARCHAR(60), NOT NULL
NumerUlicy – VARCHAR(6), NOT NULL
Telefon – VARCHAR(15)
E-mail – VARCHAR(255)

SzczegółyDostawy

IDDostawcy – INT, NOT NULL
IDProduktu – INT, NOT NULL

Zabiegi

IDZabiegu – INT, NOT NULL
NazwaZabiegu – VARCHAR(50), NOT NULL
DataZabiegu – DATETIME, @x <= NOW()
RodzajZabiegu – VARCHAR(50), NOT NULL
CenaZabiegu – FLOAT, NOT NULL, @x >= 0
PodatekZabiegu – FLOAT, NOT NULL, @x >= 0
Komentarz – VARCHAR(255)
IDKlienta – INT, NOT NULL
IDZwierzecia – INT, NOT NULL

SzczegółyZakupu

IDZakupu – INT, NOT NULL
IDProduktu – INT, NOT NULL

Klienci

IDKlienta – INT, NOT NULL
Imię – VARCHAR(40), NOT NULL
Nazwisko – VARCHAR(40), NOT NULL
Kraj – VARCHAR(30)
Miasto – VARCHAR(60)
Ulica – VARCHAR(60)
NumerUlicy – VARCHAR(6)
Telefon – VARCHAR(15)
E-mail – VARCHAR(255)

WykonawcyZabiegu

IDZabiegu – INT, NOT NULL
IDPracownika – INT, NOT NULL

Zakupy

IDZakupu – INT, NOT NULL
DataZakupu – DATETIME, NOT NULL, @x <= NOW()
KwotaZakupuBrutto – FLOAT, NOT NULL, @x >= 0
IDKlienta – INT, NOT NULL

Zwierzęta

IDZwierzęcia – INT, NOT NULL

Wiek – INT

Rodzaj – VARCHAR(50), NOT NULL

Gatunek – VARCHAR(50), NOT NULL

Masa – VARCHAR(15), @x > 0

Wysokość – VARCHAR(15), @x > 0

Długość – VARCHAR(15), @x > 0

Urlopy

IDUrlopu – INT, NOT NULL

DataRozpoczęcia – DATETIME, NOT NULL, @x <= NOW()

DataZakończenia – DATETIME, @x <= NOW(), @x >= DataRozpoczęcia

Powód – VARCHAR(255), NOT NULL

Płatność – FLOAT

IDPracownika – INT, NOT NULL

Pracownicy

IDPracownika – INT, NOT NULL

Imię – VARCHAR(30), NOT NULL

Nazwisko – VARCHAR(40), NOT NULL

Stanowisko – VARCHAR(70), NOT NULL

DataUrodzenia – DATETIME, NOT NULL, @x <= NOW()

DataZatrudnienia – DATETIME, NOT NULL, @x <= NOW(), @x >= DataUrodzenia

DataZwolnienia – DATETIME, @x <= NOW(), @x >= DataUrodzenia, @x >= DataZatrudnienia

Kraj – VARCHAR(30)

Miasto – VARCHAR(60)

Ulica – VARCHAR(60)

NumerUlicy – VARCHAR(6)

Telefon – VARCHAR(20)

E-mail – VARCHAR(255)

RodzajeZabiegów

IDRodzaju – INT, NOT NULL

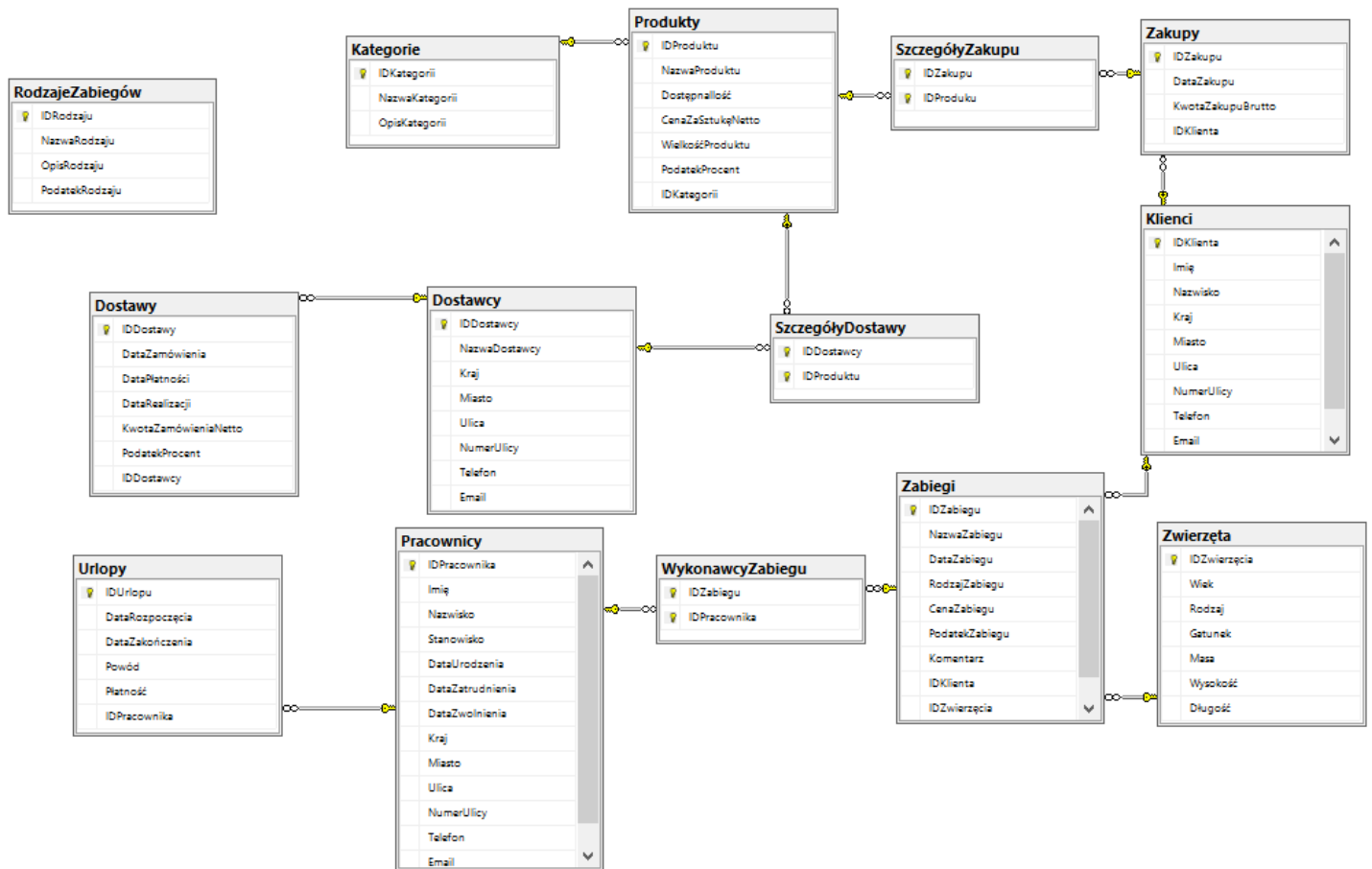
NazwaRodzaju – VARCHAR(50), NOT NULL

OpisRodzaju – VARCHAR(255)

PodatekRodzaju – FLOAT, NOT NULL, @x >= 0

6. Definicja schematu bazy danych, utworzenie bazy danych

Diagram:



Kod sql:

```
USE [master]
GO

/***** Object: Database [Weterynaryjna Klinika]    Script Date: 25/05/2020 02:33:03 *****/
CREATE DATABASE [Weterynaryjna Klinika]
CONTAINMENT = NONE
ON PRIMARY
( NAME = N'Weterynaryjna Klinika', FILENAME = N'C:\Users\Intel\Weterynaryjna Klinika.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )
LOG ON
( NAME = N'Weterynaryjna Klinika_log', FILENAME = N'C:\Users\Intel\Weterynaryjna Klinika_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )
GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [Weterynaryjna Klinika].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO

ALTER DATABASE [Weterynaryjna Klinika] SET ANSI_NULL_DEFAULT OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET ANSI_NULLS OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET ANSI_PADDING OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET ANSI_WARNINGS OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET ARITHABORT OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET AUTO_CLOSE OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET AUTO_SHRINK OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET AUTO_UPDATE_STATISTICS ON
GO
```

```

ALTER DATABASE [Weterynaryjna Klinika] SET CURSOR_CLOSE_ON_COMMIT OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET CURSOR_DEFAULT GLOBAL
GO

ALTER DATABASE [Weterynaryjna Klinika] SET CONCAT_NULL_YIELDS_NULL OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET NUMERIC_ROUNDABORT OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET QUOTED_IDENTIFIER OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET RECURSIVE_TRIGGERS OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET DISABLE_BROKER
GO

ALTER DATABASE [Weterynaryjna Klinika] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET DATE_CORRELATION_OPTIMIZATION OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET TRUSTWORTHY OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET PARAMETERIZATION SIMPLE
GO

ALTER DATABASE [Weterynaryjna Klinika] SET READ_COMMITTED_SNAPSHOT OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET HONOR_BROKER_PRIORITY OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET RECOVERY SIMPLE
GO

ALTER DATABASE [Weterynaryjna Klinika] SET MULTI_USER
GO

ALTER DATABASE [Weterynaryjna Klinika] SET PAGE_VERIFY CHECKSUM
GO

ALTER DATABASE [Weterynaryjna Klinika] SET DB_CHAINING OFF
GO

ALTER DATABASE [Weterynaryjna Klinika] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO

ALTER DATABASE [Weterynaryjna Klinika] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO

ALTER DATABASE [Weterynaryjna Klinika] SET DELAYED_DURABILITY = DISABLED
GO

ALTER DATABASE [Weterynaryjna Klinika] SET QUERY_STORE = OFF
GO

USE [Weterynaryjna Klinika]
GO

ALTER DATABASE SCOPED CONFIGURATION SET LEGACY_CARDINALITY_ESTIMATION = OFF;
GO

ALTER DATABASE SCOPED CONFIGURATION SET MAXDOP = 0;
GO

ALTER DATABASE SCOPED CONFIGURATION SET PARAMETER_SNIFFING = ON;
GO

ALTER DATABASE SCOPED CONFIGURATION SET QUERY_OPTIMIZER_HOTFIXES = OFF;
GO

```



```
ALTER DATABASE [Weterynaryjna Klinika] SET READ_WRITE
GO
```

```
USE [Weterynaryjna Klinika]
GO
```

```
/***** Object: Table [dbo].[Dostawcy]    Script Date: 25/05/2020 02:34:47 *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[Dostawcy](
    [IDDostawcy] [int] NOT NULL,
    [NazwaDostawcy] [varchar](60) NOT NULL,
    [Kraj] [varchar](30) NOT NULL,
    [Miasto] [varchar](60) NOT NULL,
    [Ulica] [varchar](60) NOT NULL,
    [NumerUlicy] [varchar](6) NOT NULL,
    [Telefon] [varchar](15) NULL,
    [Email] [varchar](255) NULL,
    CONSTRAINT [PK_Dostawcy] PRIMARY KEY CLUSTERED
(
    [IDDostawcy] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
USE [Weterynaryjna Klinika]
GO
```

```
/***** Object: Table [dbo].[Dostawy]    Script Date: 25/05/2020 02:35:08 *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[Dostawy](
    [IDDostawy] [int] NOT NULL,
    [DataZamówienia] [datetime] NULL,
    [DataPłatności] [datetime] NULL,
    [DataRealizacji] [datetime] NULL,
    [KwotaZamówieniaNetto] [float] NOT NULL,
    [PodatekProcent] [float] NOT NULL,
    [IDDostawcy] [int] NOT NULL,
    CONSTRAINT [PK_Dostawy] PRIMARY KEY CLUSTERED
(
    [IDDostawy] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[Dostawy] WITH CHECK ADD CONSTRAINT [FK_Dostawy_Dostawcy1] FOREIGN KEY([IDDostawcy])
REFERENCES [dbo].[Dostawcy] ([IDDostawcy])
GO
```

```
ALTER TABLE [dbo].[Dostawy] CHECK CONSTRAINT [FK_Dostawy_Dostawcy1]
GO
```

```
USE [Weterynaryjna Klinika]
GO
```

```
/***** Object: Table [dbo].[Kategorie]    Script Date: 25/05/2020 02:35:24 *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[Kategorie](
    [IDKategorii] [int] NOT NULL,
    [NazwaKategorii] [varchar](30) NOT NULL,
    [OpisKategorii] [varchar](100) NULL,
    CONSTRAINT [PK_Kategorie] PRIMARY KEY CLUSTERED
(
    [IDKategorii] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```

USE [Weterynaryjna Klinika]
GO

/***** Object: Table [dbo].[Klienci]    Script Date: 25/05/2020 02:35:41 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Klienci](
    [IDKlienta] [int] NOT NULL,
    [Imię] [varchar](40) NOT NULL,
    [Nazwisko] [varchar](40) NOT NULL,
    [Kraj] [varchar](30) NULL,
    [Miasto] [varchar](60) NULL,
    [Ulica] [varchar](60) NULL,
    [NumerUlicy] [varchar](6) NULL,
    [Telefon] [varchar](15) NULL,
    [Email] [varchar](255) NULL,
    CONSTRAINT [PK_Klienci] PRIMARY KEY CLUSTERED
(
    [IDKlienta] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

USE [Weterynaryjna Klinika]
GO

/***** Object: Table [dbo].[Pracownicy]    Script Date: 25/05/2020 02:35:55 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Pracownicy](
    [IDPracownika] [int] NOT NULL,
    [Imię] [varchar](30) NOT NULL,
    [Nazwisko] [varchar](40) NOT NULL,
    [Stanowisko] [varchar](70) NOT NULL,
    [DataUrodzenia] [datetime] NOT NULL,
    [DataZatrudnienia] [datetime] NOT NULL,
    [DataZwolnienia] [datetime] NULL,
    [Kraj] [varchar](30) NULL,
    [Miasto] [varchar](60) NULL,
    [Ulica] [varchar](60) NULL,
    [NumerUlicy] [varchar](6) NULL,
    [Telefon] [varchar](20) NULL,
    [Email] [varchar](255) NULL,
    CONSTRAINT [PK_Pracownicy] PRIMARY KEY CLUSTERED
(
    [IDPracownika] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

USE [Weterynaryjna Klinika]
GO

/***** Object: Table [dbo].[Produkty]    Script Date: 25/05/2020 02:36:12 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Produkty](
    [IDProduktu] [int] NOT NULL,
    [NazwaProduktu] [varchar](40) NOT NULL,
    [DostępnaIlość] [int] NOT NULL,
    [CenaZaSztukęNetto] [float] NULL,
    [WielkośćProduktu] [varchar](40) NULL,
    [PodatekProcent] [float] NULL,
    [IDKategorii] [int] NOT NULL,
    CONSTRAINT [PK_Produkty] PRIMARY KEY CLUSTERED
(
    [IDProduktu] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Produkty] WITH CHECK ADD CONSTRAINT [FK_Produkty_Kategorie] FOREIGN KEY([IDKategorii])
REFERENCES [dbo].[Kategorie] ([IDKategorii])
GO

ALTER TABLE [dbo].[Produkty] CHECK CONSTRAINT [FK_Produkty_Kategorie]
GO

```

```
USE [Weterynaryjna Klinika]
GO

/***** Object: Table [dbo].[RodzajeZabiegów]    Script Date: 25/05/2020 02:36:31 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[RodzajeZabiegów](
    [IDRodzaju] [int] NOT NULL,
    [NazwaRodzaju] [varchar](50) NOT NULL,
    [OpisRodzaju] [varchar](255) NULL,
    [PodatekRodzaju] [float] NOT NULL,
    CONSTRAINT [PK_RodzajeZabiegów] PRIMARY KEY CLUSTERED
(
    [IDRodzaju] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
USE [Weterynaryjna Klinika]
GO

/***** Object: Table [dbo].[SzczegółyDostawy]    Script Date: 25/05/2020 02:36:45 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[SzczegółyDostawy](
    [IDDostawy] [int] NOT NULL,
    [IDProduktu] [int] NOT NULL,
    CONSTRAINT [PK_SzczegółyDostawy] PRIMARY KEY CLUSTERED
(
    [IDDostawy] ASC,
    [IDProduktu] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[SzczegółyDostawy] WITH CHECK ADD CONSTRAINT [FK_SzczegółyDostawy_Dostawy] FOREIGN KEY([IDDostawy])
REFERENCES [dbo].[Dostawy] ([IDDostawy])
GO

ALTER TABLE [dbo].[SzczegółyDostawy] CHECK CONSTRAINT [FK_SzczegółyDostawy_Dostawy]
GO

ALTER TABLE [dbo].[SzczegółyDostawy] WITH CHECK ADD CONSTRAINT [FK_SzczegółyDostawy_Produkty] FOREIGN KEY([IDProduktu])
REFERENCES [dbo].[Produkty] ([IDProduktu])
GO
```

```
ALTER TABLE [dbo].[SzczegółyDostawy] CHECK CONSTRAINT [FK_SzczegółyDostawy_Produkty]
GO

USE [Weterynaryjna Klinika]
GO

/***** Object: Table [dbo].[SzczegółyZakupu]    Script Date: 25/05/2020 02:37:04 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[SzczegółyZakupu](
    [IDZakupu] [int] NOT NULL,
    [IDProduktu] [int] NOT NULL,
    CONSTRAINT [PK_SzczegółyZakupu] PRIMARY KEY CLUSTERED
(
    [IDZakupu] ASC,
    [IDProduktu] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[SzczegółyZakupu] WITH CHECK ADD CONSTRAINT [FK_SzczegółyZakupu_Produkty1] FOREIGN KEY([IDProduktu])
REFERENCES [dbo].[Produkty] ([IDProduktu])
GO

ALTER TABLE [dbo].[SzczegółyZakupu] CHECK CONSTRAINT [FK_SzczegółyZakupu_Produkty1]
GO

ALTER TABLE [dbo].[SzczegółyZakupu] WITH CHECK ADD CONSTRAINT [FK_SzczegółyZakupu_Zakupy] FOREIGN KEY([IDZakupu])
REFERENCES [dbo].[Zakupy] ([IDZakupu])
GO

ALTER TABLE [dbo].[SzczegółyZakupu] CHECK CONSTRAINT [FK_SzczegółyZakupu_Zakupy]
GO
```

```

USE [Weterynaryjna Klinika]
GO

/***** Object: Table [dbo].[Urlopy]    Script Date: 25/05/2020 02:37:22 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Urlopy](
    [IDUrlopu] [int] NOT NULL,
    [DataRozpoczęcia] [datetime] NOT NULL,
    [DataZakończenia] [datetime] NULL,
    [Powód] [varchar](255) NOT NULL,
    [Płatność] [float] NULL,
    [IDPracownika] [int] NOT NULL,
    CONSTRAINT [PK_Urlopy] PRIMARY KEY CLUSTERED
(
    [IDUrlopu] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Urlopy] WITH CHECK ADD CONSTRAINT [FK_Urlopy_Pracownicy] FOREIGN KEY([IDPracownika])
REFERENCES [dbo].[Pracownicy] ([IDPracownika])
GO

ALTER TABLE [dbo].[Urlopy] CHECK CONSTRAINT [FK_Urlopy_Pracownicy]
GO


USE [Weterynaryjna Klinika]
GO

/***** Object: Table [dbo].[WykonawcyZabiegu]    Script Date: 25/05/2020 02:37:36 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[WykonawcyZabiegu](
    [IDZabiegu] [int] NOT NULL,
    [IDPracownika] [int] NOT NULL,
    CONSTRAINT [PK_WykonawcyZabiegu] PRIMARY KEY CLUSTERED
(
    [IDZabiegu] ASC,
    [IDPracownika] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[WykonawcyZabiegu] WITH CHECK ADD CONSTRAINT [FK_WykonawcyZabiegu_Pracownicy] FOREIGN KEY([IDPracownika])
REFERENCES [dbo].[Pracownicy] ([IDPracownika])
GO

ALTER TABLE [dbo].[WykonawcyZabiegu] CHECK CONSTRAINT [FK_WykonawcyZabiegu_Pracownicy]
GO

ALTER TABLE [dbo].[WykonawcyZabiegu] WITH CHECK ADD CONSTRAINT [FK_WykonawcyZabiegu_Zabiegi] FOREIGN KEY([IDZabiegu])
REFERENCES [dbo].[Zabiegi] ([IDZabiegu])
GO

ALTER TABLE [dbo].[WykonawcyZabiegu] CHECK CONSTRAINT [FK_WykonawcyZabiegu_Zabiegi]
GO


USE [Weterynaryjna Klinika]
GO

/***** Object: Table [dbo].[Zwierzęta]    Script Date: 25/05/2020 02:37:54 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Zwierzęta](
    [IDZwierzęcia] [int] NOT NULL,
    [Wiek] [int] NULL,
    [Rodzaj] [varchar](50) NOT NULL,
    [Gatunek] [varchar](50) NOT NULL,
    [Masa] [varchar](15) NULL,
    [Wysokość] [varchar](15) NULL,
    [Długość] [varchar](15) NULL,
    CONSTRAINT [PK_Zwierzęta] PRIMARY KEY CLUSTERED
(
    [IDZwierzęcia] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

USE [Weterynaryjna Klinika]
GO

/***** Object: Table [dbo].[Zabiegi]    Script Date: 25/05/2020 02:38:08 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

)CREATE TABLE [dbo].[Zabiegi](
    [IDZabiegu] [int] NOT NULL,
    [NazwaZabiegu] [varchar](50) NOT NULL,
    [DataZabiegu] [datetime] NULL,
    [RodzajZabiegu] [varchar](50) NOT NULL,
    [CenaZabiegu] [float] NOT NULL,
    [PodatekZabiegu] [float] NOT NULL,
    [Komentarz] [varchar](255) NULL,
    [IDKlienta] [int] NOT NULL,
    [IDZwierzęcia] [int] NOT NULL,
    CONSTRAINT [PK_Zabiegi] PRIMARY KEY CLUSTERED
(
    [IDZabiegu] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

)ALTER TABLE [dbo].[Zabiegi] WITH CHECK ADD CONSTRAINT [FK_Zabiegi_Klienci] FOREIGN KEY([IDKlienta])
REFERENCES [dbo].[Klienci] ([IDKlienta])
GO

ALTER TABLE [dbo].[Zabiegi] CHECK CONSTRAINT [FK_Zabiegi_Klienci]
GO

)ALTER TABLE [dbo].[Zabiegi] WITH CHECK ADD CONSTRAINT [FK_Zabiegi_Zwierzęta] FOREIGN KEY([IDZwierzęcia])
REFERENCES [dbo].[Zwierzęta] ([IDZwierzęcia])
GO

ALTER TABLE [dbo].[Zabiegi] CHECK CONSTRAINT [FK_Zabiegi_Zwierzęta]
GO

```

```

USE [Weterynaryjna Klinika]
GO

/***** Object: Table [dbo].[Zakupy]    Script Date: 25/05/2020 02:38:24 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Zakupy](
    [IDZakupu] [int] NOT NULL,
    [DataZakupu] [datetime] NOT NULL,
    [KwotaZakupuBrutto] [float] NOT NULL,
    [IDKlienta] [int] NOT NULL,
    CONSTRAINT [PK_Zakupy] PRIMARY KEY CLUSTERED
(
    [IDZakupu] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Zakupy] WITH CHECK ADD CONSTRAINT [FK_Zakupy_Klienci] FOREIGN KEY([IDKlienta])
REFERENCES [dbo].[Klienci] ([IDKlienta])
GO

ALTER TABLE [dbo].[Zakupy] CHECK CONSTRAINT [FK_Zakupy_Klienci]
GO

```

7 + 8. Definicja I implementacja niedeklaratywnych mechanizmów sprawdzania poprawności danych:

Wszystkie kolumny, w których znajdują się daty I wartości przedstawiające pieniądze muszą zostać sprawdzone czy mają sens.

Sprawdzenie wartości numerycznych dotyczących zwierzęcia w tabeli Zwierzęta (muszą być większe niż 0):

```
USE [Weterynaryjna Klinika]
GO

/***** Object: Trigger [dbo].[sprawdzenieParamZwierzat]    Script Date: 25/05/2020 03:07:46 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TRIGGER [dbo].[sprawdzenieParamZwierzat] ON [dbo].[Zwierzęta]
FOR INSERT
NOT FOR REPLICATION
AS
IF (SELECT COUNT(*) FROM inserted)>=1
BEGIN
    IF((SELECT Masa from inserted) <= 0)
    BEGIN RAISERROR('Masa zwierzęcia nie może być mniejsza lub równa 0',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT Wysokość from inserted) <= 0)
    BEGIN RAISERROR('Wysokość zwierzęcia nie może być mniejsza lub równa 0',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT Długość from inserted) <= 0)
    BEGIN RAISERROR('Długość nie może mieć wartości mniejszej lub równa 0',16,1)
    ROLLBACK TRANSACTION
    END
END
GO

ALTER TABLE [dbo].[Zwierzęta] ENABLE TRIGGER [sprawdzenieParamZwierzat]
GO
```

Sprawdzenie wartości numerycznych i dat w tabeli Dostawy:

```
USE [Weterynaryjna Klinika]
GO

/***** Object: Trigger [dbo].[sprawdzenieDatDostawy]    Script Date: 25/05/2020 03:08:53 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TRIGGER [dbo].[sprawdzenieDatDostawy] ON [dbo].[Dostawy]
FOR INSERT
NOT FOR REPLICATION
AS
IF (SELECT COUNT(*) FROM inserted)>=1
BEGIN
    IF((SELECT DataZamowienia from inserted) > GETDATE())
    BEGIN RAISERROR('Data zamowienia nie może być późniejsza niż dzień dzisiejszy',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT DataPłatności from inserted) > GETDATE() AND (SELECT DataPłatności from inserted) > (SELECT DataZamowienia from inserted))
    BEGIN RAISERROR('Data płatności nie może być późniejsza niż dzień dzisiejszy i data zamowienia',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT DataRealizacji from inserted) > GETDATE() AND (SELECT DataRealizacji from inserted) > (SELECT DataPłatności from inserted)
    AND (SELECT DataPłatności from inserted) IS NOT NULL)
    BEGIN RAISERROR('Data realizacji nie może być późniejsza niż dzień dzisiejszy i data płatności',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT KwotaZamowieniaNetto from inserted) < 0)
    BEGIN RAISERROR('Kwota zamówienia nie może być mniejsza niż 0',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT PodatekProcent from inserted) < 0)
    BEGIN RAISERROR('Podatek nie może mieć wartości mniejszej niż 0',16,1)
    ROLLBACK TRANSACTION
    END
END
END
GO

ALTER TABLE [dbo].[Dostawy] ENABLE TRIGGER [sprawdzenieDatDostawy]
```

Sprawdzenie dat w tabeli Pracownicy:

```
USE [Weterynaryjna Klinika]
GO

/***** Object: Trigger [dbo].[sprawdzenieDatPracownikow]    Script Date: 25/05/2020 03:10:09 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TRIGGER [dbo].[sprawdzenieDatPracownikow] ON [dbo].[Pracownicy]
FOR INSERT
NOT FOR REPLICATION
AS
IF (SELECT COUNT(*) FROM inserted)>=1
BEGIN
    IF((SELECT DataUrodzenia from inserted) > GETDATE() )
    BEGIN RAISERROR('Data urodzenia nie może być późniejsza niż dzień dzisiejszy',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT DataZatrudnienia from inserted) > GETDATE() AND DATEDIFF(YY,(SELECT DataUrodzenia from inserted),(SELECT DataZatrudnienia from inserted)) < 18)
    BEGIN RAISERROR('Data zatrudnienia nie może być późniejsza niż dzień dzisiejszy i data urodzenia musi być przynajmniej 18 lat wcześniejsza niż data zatrudnienia',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT DataZwolnienia from inserted) > GETDATE() AND (SELECT DataZatrudnienia from inserted) < (SELECT DataZwolnienia from inserted))
    BEGIN RAISERROR('Data zwolnienia nie może być późniejsza niż dzień dzisiejszy i data zatrudnienia nie może być później niż data zwolnienia',16,1)
    ROLLBACK TRANSACTION
    END
END
END
GO

ALTER TABLE [dbo].[Pracownicy] ENABLE TRIGGER [sprawdzenieDatPracownikow]
```

Sprawdzenie wartości numerycznych w tabeli Produkty:

```
USE [Weterynaryjna Klinika]
GO

/***** Object: Trigger [dbo].[sprawdzenieParamProduktow]    Script Date: 25/05/2020 03:10:43 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TRIGGER [dbo].[sprawdzenieParamProduktow] ON [dbo].[Produkty]
FOR INSERT
NOT FOR REPLICATION
AS
IF (SELECT COUNT(*) FROM inserted)>=1
BEGIN
    IF((SELECT DostepnaIlość from inserted) < 0)
    BEGIN RAISERROR('Dostępna ilość nie może być mniejsza niż 0',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT CenaZaSztukeNetto from inserted) < 0)
    BEGIN RAISERROR('Cena za sztukę nie może być mniejsza niż 0',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT PodatekProcent from inserted) < 0)
    BEGIN RAISERROR('Podatek nie może być mniejszy niż 0',16,1)
    ROLLBACK TRANSACTION
    END
END
GO

ALTER TABLE [dbo].[Produkty] ENABLE TRIGGER [sprawdzenieParamProduktow]
GO
```

Sprawdzenie wartości numerycznych w tabeli RodzajeZabiegów:

```
USE [Weterynaryjna Klinika]
GO

/***** Object: Trigger [dbo].[sprawdzenieRodzajowZabiegow]    Script Date: 25/05/2020 03:11:23 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TRIGGER [dbo].[sprawdzenieRodzajowZabiegow] ON [dbo].[RodzajeZabiegów]
FOR INSERT
NOT FOR REPLICATION
AS
IF (SELECT COUNT(*) FROM inserted)>=1
BEGIN
    IF((SELECT PodatekRodzaju from inserted) < 0)
    BEGIN RAISERROR('Podatek nie może być mniejszy niż 0',16,1)
    ROLLBACK TRANSACTION
    END
END
GO

ALTER TABLE [dbo].[RodzajeZabiegów] ENABLE TRIGGER [sprawdzenieRodzajowZabiegow]
GO
```


Sprawdzenie wartości numerycznych i dat w tabeli Urlopy:

```
USE [Weterynaryjna Klinika]
GO

/***** Object: Trigger [dbo].[sprawdzenieParamUrlopu]    Script Date: 25/05/2020 03:12:15 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TRIGGER [dbo].[sprawdzenieParamUrlopu] ON [dbo].[Urlopy]
FOR INSERT
NOT FOR REPLICATION
AS
IF (SELECT COUNT(*) FROM inserted)>=1
BEGIN
    IF((SELECT DataRozpoczecia from inserted) > GETDATE())
    BEGIN RAISERROR('Data rozpoczęcia urlopu nie może być późniejsza niż dzień dzisiejszy',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT DataZakończenia from inserted) > GETDATE() AND (SELECT DataRozpoczecia from inserted) > (SELECT DataZakończenia from inserted))
    BEGIN RAISERROR('Data zakończenia nie może być późniejsza niż dzień dzisiejszy i da',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT Płatność from inserted) < 0)
    BEGIN RAISERROR('Płatność nie może mieć wartości mniejszej niż 0',16,1)
    ROLLBACK TRANSACTION
    END
END
GO

ALTER TABLE [dbo].[Urlopy] ENABLE TRIGGER [sprawdzenieParamUrlopu]
GO
```

Sprawdzenie wartości numerycznych i dat w tabeli Zabiegi:

```
USE [Weterynaryjna Klinika]
GO

/***** Object: Trigger [dbo].[sprawdzenieParamZabiegow]    Script Date: 25/05/2020 03:12:53 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TRIGGER [dbo].[sprawdzenieParamZabiegow] ON [dbo].[Zabiegi]
FOR INSERT
NOT FOR REPLICATION
AS
IF (SELECT COUNT(*) FROM inserted)>=1
BEGIN
    IF((SELECT DataZabiegu from inserted) > GETDATE())
    BEGIN RAISERROR('Data rozpoczęcia zabiegu nie może być późniejsza niż dzień dzisiejszy',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT CenaZabiegu from inserted) < 0)
    BEGIN RAISERROR('Cena zabiegu nie może mieć wartości mniejszej niż 0',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT PodatekZabiegu from inserted) < 0)
    BEGIN RAISERROR('Podatek zabiegu nie może mieć wartości mniejszej niż 0',16,1)
    ROLLBACK TRANSACTION
    END
END
GO

ALTER TABLE [dbo].[Zabiegi] ENABLE TRIGGER [sprawdzenieParamZabiegow]
GO
```

Sprawdzenie wartości numerycznych i dat w tabeli Zakupy:

```
USE [Weterynaryjna Klinika]
GO

/***** Object: Trigger [dbo].[sprawdzenieParamZakupu]    Script Date: 25/05/2020 03:13:19 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TRIGGER [dbo].[sprawdzenieParamZakupu] ON [dbo].[Zakupy]
FOR INSERT
NOT FOR REPLICATION
AS
IF (SELECT COUNT(*) FROM inserted)>=1
BEGIN
    IF((SELECT DataZakupu from inserted) > GETDATE())
    BEGIN RAISERROR('Data zakupu nie może być późniejsza niż dzień dzisiejszy',16,1)
    ROLLBACK TRANSACTION
    END
    IF((SELECT KwotaZakupuBrutto from inserted) < 0)
    BEGIN RAISERROR('Kwota zakupu nie może być mniejsza niż 0',16,1)
    ROLLBACK TRANSACTION
    END
END
GO

ALTER TABLE [dbo].[Zakupy] ENABLE TRIGGER [sprawdzenieParamZakupu]
GO
```

9. Implementacja kodu wspomagająca aplikację użytkową:

```
DROP FUNCTION IF EXISTS dbo.fn_ProduktyKtórychNieMa
GO
CREATE FUNCTION fn_ProduktyKtórychNieMa()
RETURNS TABLE
AS
RETURN
(
    SELECT * FROM Produkty
    WHERE DostępnaIlość = 0
);
GO
```

```
DROP FUNCTION IF EXISTS dbo.fn_ProduktyKtóreSa
GO
CREATE FUNCTION fn_ProduktyKtóreSa()
RETURNS TABLE
AS
RETURN
(
    SELECT * FROM Produkty
    WHERE DostępnaIlość != 0
);
GO
```

```
DROP FUNCTION IF EXISTS dbo.fn_PracownicyNaUrlopie
GO
CREATE FUNCTION fn_PracownicyNaUrlopie()
RETURNS TABLE
AS
RETURN
(
    SELECT Pracownicy.Imię, Pracownicy.Nazwisko FROM Pracownicy
    INNER JOIN Urlopy ON
    Pracownicy.IDPracownika = Urlopy.IDPracownika
    WHERE Urlopy.DataZakończenia IS NULL OR Urlopy.DataZakończenia > GETDATE()
);
GO
```

```
DROP FUNCTION IF EXISTS dbo.fn_DostawyOczekujaceBrakProduktu
GO
CREATE FUNCTION fn_DostawyOczekujaceBrakProduktu()
RETURNS TABLE
AS
RETURN
(
    SELECT Dostawy.IDDostawy, SzczegółyDostawy.IDProduktu FROM Dostawy
    INNER JOIN SzczegółyDostawy ON
    SzczegółyDostawy.IDDostawy = Dostawy.IDDostawy
    INNER JOIN Produkty ON
    Produkty.IDProduktu = SzczegółyDostawy.IDProduktu
    WHERE DostępnaIlość = 0
);
GO
```

10. Wprowadzenie przykładowych danych:

```
INSERT INTO Dostawcy VALUES(1,'Firma leków', 'Polska', 'Warszawa', 'Grochowa', '13', '665533542', 'losowyemail@dop.pl');
INSERT INTO Dostawcy VALUES(2,'Firma rzeczy', 'Polska', 'Warszawa', 'Warecka', '44', '454138735', 'losowyemail2@dop.pl');
INSERT INTO Dostawy VALUES(1,1999-06-06, 1999-06-06, 1999-06-06, 34.99, 0.5, 1);
INSERT INTO Kategorie VALUES(1,'Lek', 'Leki i preparaty lecznicze');
INSERT INTO Kategorie VALUES(2,'Artykuł pielęgnacyjny', 'Środki do pielęgnacji i środki czyszczące');
INSERT INTO Klienci VALUES(1,'Maciej', 'Grochowski', 'Polska', 'Warszawa', 'Grochowa', '55', '665533542', 'losowyemail4545@dop.pl');
INSERT INTO Pracownicy VALUES(1,'Artur', 'Żmijewski', 'Weterynarz', 1995-06-03, 2018-04-03, Null, 'Polska', 'Kraków', 'Krakowska', '13', Null, Null);
INSERT INTO Pracownicy VALUES(2,'Roman', 'Nowak', 'Weterynarz', 1986-04-03, 2018-04-03, Null, 'Polska', 'Kraków', 'Krakowska', '13', Null, Null);
INSERT INTO Produkty VALUES(1,'Rzecz1', 20, 25.99, '3 litry', 0.24, 1);
INSERT INTO Produkty VALUES(2,'Rzecz2', 0, 19.99, '20 tabletek', 0.08, 2);
INSERT INTO Zakupy VALUES(1,2019-06-04, 99.99, 1);
INSERT INTO Zakupy VALUES(2,2019-06-04, 46.09, 1);
INSERT INTO Zwierzeta VALUES(1,5, 'Pies', 'Duży', '14 kg', '3 m', '1 m');
INSERT INTO Zwierzeta VALUES(2,1, 'Kot', 'Mały', '1 kg', '2 m', '1 m');
```

Link do githuba:

https://github.com/sm2009pl/BDIIS14_Magiera_Szymon_1B