

## Spis treści

1.	Cel i wymagania.....	1
2.	DZE.....	3
3.	Model relacyjny .....	4
4.	Normalizacja:.....	4
5.	Zasad poprawności danych: .....	4
6.	Definicja schematu bazy danych – utworzenie bazy danych: .....	5
7.	8. Definicja niedeklaratywnych mechanizmów sprawdzania poprawności danych .....	13
9.	Kod wspomagający aplikację użytkową .....	16
10	Wprowadzenie przykładowych danych.....	18

## 1. Cel i wymagania

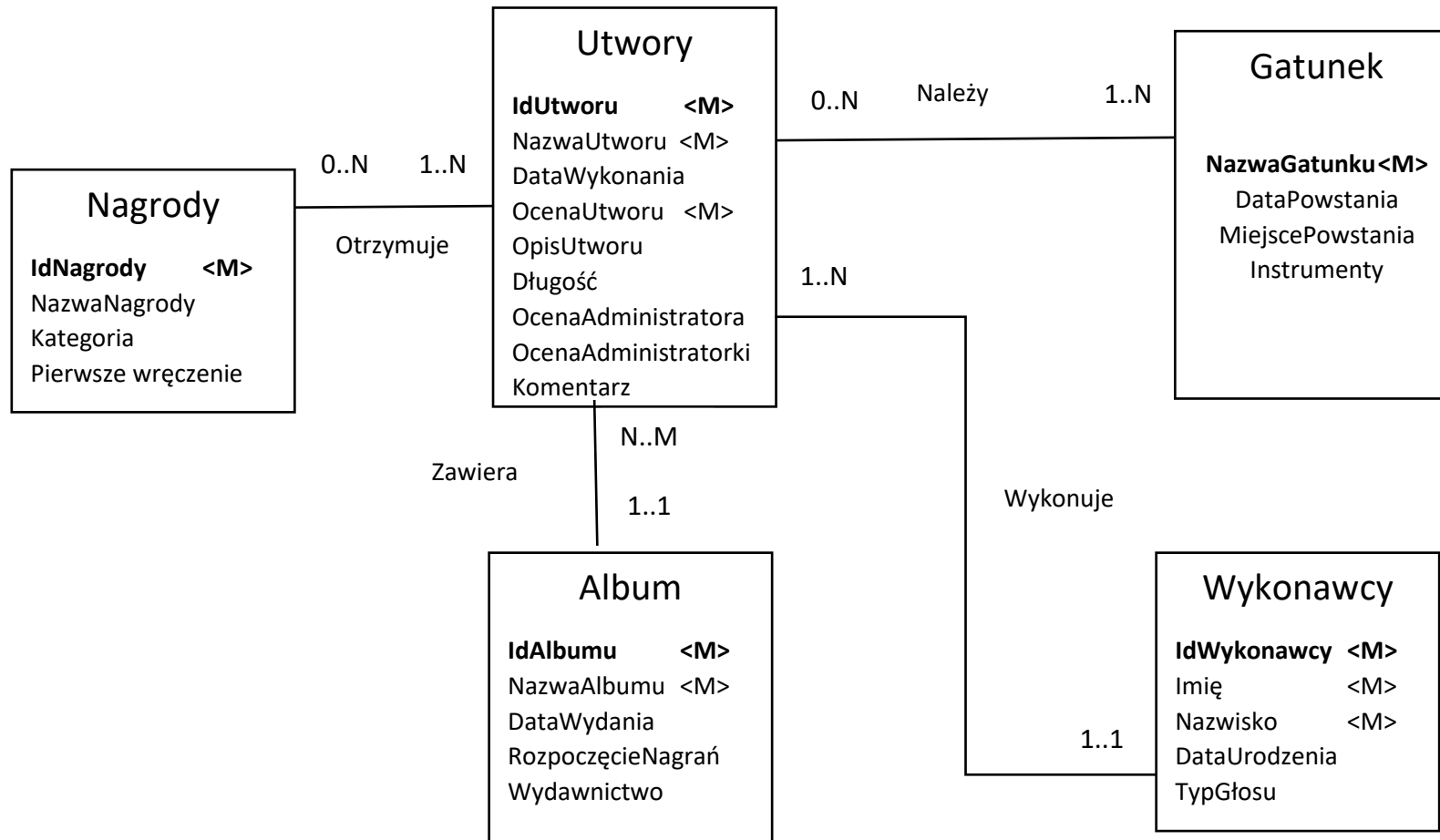
Celem systemu jest przechowywanie podstawowych informacji dotyczących przebojów muzycznych w sposób, który umożliwia intuicyjną modyfikację i intuicyjny przegląd danych.  
Wymagania:

Lp	Opis wymagania	Waga (0-1)	Źródło	Miara	Uwagi
1.	Możliwość dodawania nowych utworów do bazy i możliwość edycji informacji na ich temat.	1	Internet, albumy muzyczne		
2.	Przy wprowadzaniu nowej informacji do bazy każdy wymagany atrybut musi zostać uzupełniony	1	Internet, albumy muzyczne		
3.	Możliwość dodanie/ edycji wykonawcy wraz z informacjami o nim do bazy.	0.75	Internet, biografie		
4.	Możliwość dodania/ edycji albumu wraz z informacjami o nim do bazy.	0.25	Internet, albumy muzyczne		
5.	Możliwość dodania/ edycji gatunku i informacji o nim do bazy.	0.75	Internet, albumy muzyczne, encyklopedie		
6.	Możliwość dodania/ edycji nagrody i informacji na jej temat do bazy	0.25	Internet		
7.	Użytkownik: Administrator – dostęp do wszystkich części bazy, pełne prawo edytowania, tworzenia usuwania rekordów, wystawiania	1			

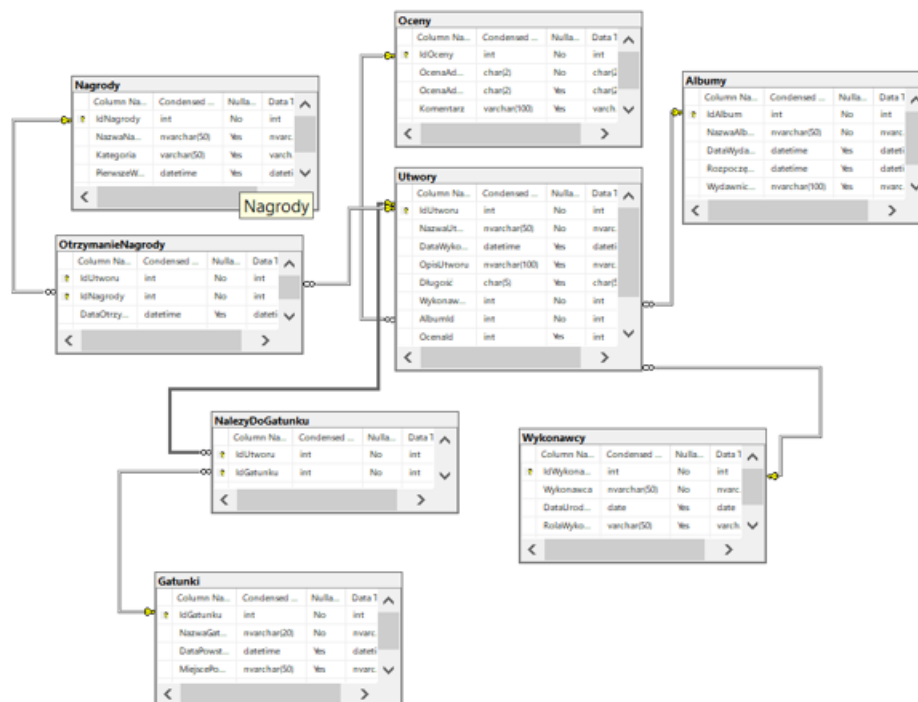
	ocenAdministratora, brak uprawnień do wystawiania ocenAdministratorki.				
8.	Użytkownik: Gość – dostęp do wszystkich części bazy, bez prawa edycji, tworzeni i usuwania rekordów, brak uprawnień do wystawiania ocen.	1			
9.	Użytkownik: Administratorka – dostęp do wszystkich części bazy, pełne prawo edycji, tworzenia, usuwania rekordów, brak uprawnień do wystawiania ocenyAdministratora, uprawnienie do wystawienia ocenyAdministratorki.	1			
10.	Możliwość dodanie/ edycji ocen wraz z danymi odnośnie werdyktu.	0.25	Administrator, administratorka		
11	Daty wprowadzane do Bazy muszą mieć sens logiczny – jeżeli jakaś rzecz nie mogła się wydarzyć przed dniem dzisiejszy powinna możliwość dodania takiej daty zostać zablokowana	0.1			

## 2. DZE

\*<M> - oznacza atrybut obowiązkowy



### 3. Model relacyjny



### 4. Normalizacja:

1NF – mówi o atomowości danych, każde pole ma przechowywać jedną informację. Wydaje mi się iż moja baza danych posiada atomowe wartości – nie widzę danych, które wymagałyby rozbicia na mniejsze. Każda encja posiada też swój klucz główny, który jednoznacznie identyfikuje wartości w tabeli.

2NF – W niej każda tabela powinna przechowywać dane dotyczące tylko konkretnej klasy obiektów. W pierwszych wersjach Nie istniała tabela oceny – ocena znajdowała się w tabeli utwory. Postanowiłem, że wyciągnę tę daną do osobnej tabeli. Rozwahałem tu usunięcie kolumny komentarze, z tabeli Oceny jako że byłaby to redundancja związana z kolumną Opis z klasy utwory. Uznałem jednak, że komentarz jest opinią, a opis jest faktem, czyli tyczy się tak naprawdę innych rzeczy.

3NF - zakłada że każdy niekluczowy argument jest bezpośrednio zależny tylko od klucza głównego, a nie od innej kolumny. Nie widzę w mojej Bazie takiego przypadku – więc zakładałam iż moja baza jest w 3-ciej postaci normalnej.

### 5. Zasad poprawności danych:

Data powstania gatunku nie powinna być późniejsza niż obecna data, podobnie jak data wydania albumu, data wykonania (tabela utworu), data urodzenia/ powstania (w tabeli wykonawcy), pierwsze wręczenie (tabela nagrody), dataOtrzymania (encja otrzymanie nagrody).

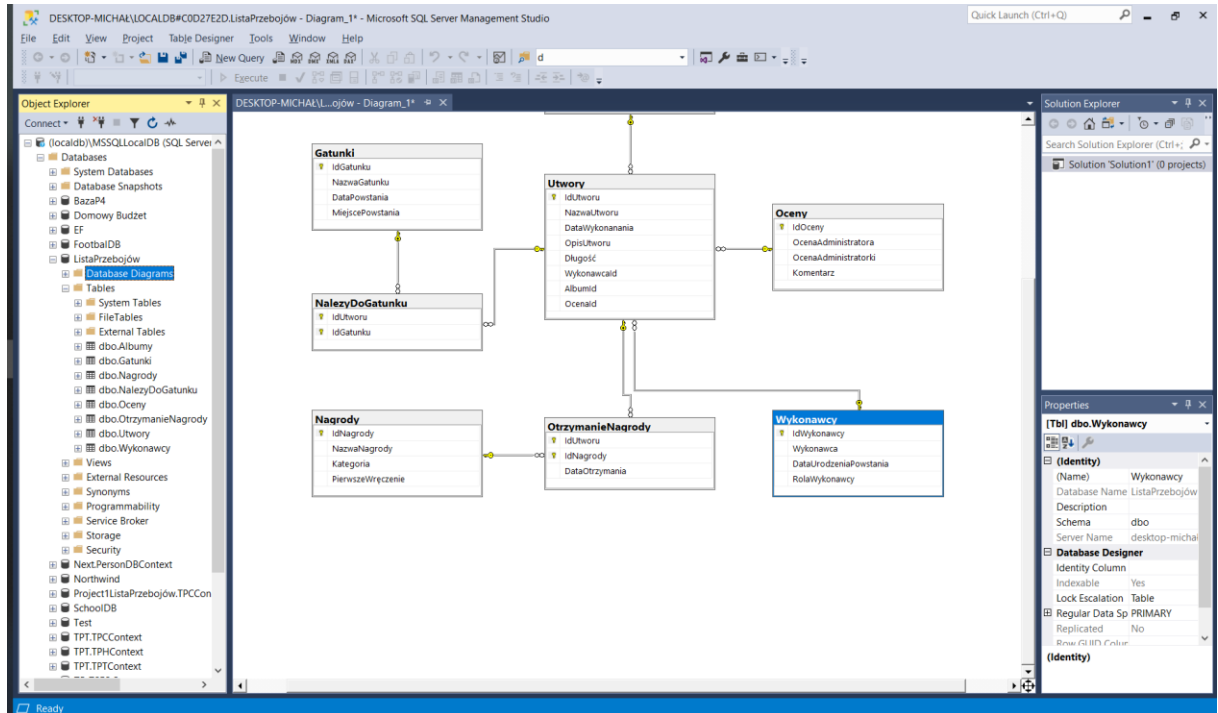
Data wydania albumu musi być późniejsza niż data rozpoczęcia nagrań.

Przy dodawaniu utworu do bazy jego nazwa, nie może być polem pustym. Tak samo do każdego utworu powinien być przypisany album, wykonawca, gatunki (-ek).

Analogiczna zasada dla innych tabel – dodając nowy album do bazy, należy podać jego nazwę (pole to nie może pozostać puste), wprowadzając nowy gatunek należy pamiętać o podaniu nazwy gatunku, dodając wykonawcę także wymagana jest nazwa wykonawcy/ zespołu. Przy dodawaniu utworu należy pamiętać o wpisaniu w tabeli Oceny wartości dla ocena administratora.

## 6. Definicja schematu bazy danych – utworzenie bazy danych:

Skrypt sqlowi znajduje się w pliku – zadanie6.sql, a także poniżej:



➔ [NACIŚNIJ TUTAJ ABY PRZESKOCZYĆ SKRYPT](#) ⬅

```
USE [master]
GO
/***** Object: Database [ListaPrzebojow]    Script Date: 23.05.2020 17:59:36
*****/
CREATE DATABASE [ListaPrzebojow]
    CONTAINMENT = NONE
    ON PRIMARY
    ( NAME = N'ListaPrzebojow', FILENAME = N'C:\Users\games\ListaPrzebojow.mdf' ,
      SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )
    LOG ON
    ( NAME = N'ListaPrzebojow_log', FILENAME =
      N'C:\Users\games\ListaPrzebojow_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB ,
      FILEGROWTH = 65536KB )
GO
ALTER DATABASE [ListaPrzebojow] SET COMPATIBILITY_LEVEL = 130
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [ListaPrzebojow].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [ListaPrzebojow] SET ANSI_NULL_DEFAULT OFF
GO
```

```

ALTER DATABASE [ListaPrzebojów] SET ANSI_NULLS OFF
GO
ALTER DATABASE [ListaPrzebojów] SET ANSI_PADDING OFF
GO
ALTER DATABASE [ListaPrzebojów] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [ListaPrzebojów] SET ARITHABORT OFF
GO
ALTER DATABASE [ListaPrzebojów] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [ListaPrzebojów] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [ListaPrzebojów] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [ListaPrzebojów] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [ListaPrzebojów] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [ListaPrzebojów] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [ListaPrzebojów] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [ListaPrzebojów] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [ListaPrzebojów] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [ListaPrzebojów] SET  DISABLE_BROKER
GO
ALTER DATABASE [ListaPrzebojów] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [ListaPrzebojów] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [ListaPrzebojów] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [ListaPrzebojów] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [ListaPrzebojów] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [ListaPrzebojów] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [ListaPrzebojów] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [ListaPrzebojów] SET RECOVERY SIMPLE
GO
ALTER DATABASE [ListaPrzebojów] SET  MULTI_USER
GO
ALTER DATABASE [ListaPrzebojów] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [ListaPrzebojów] SET DB_CHAINING OFF
GO
ALTER DATABASE [ListaPrzebojów] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [ListaPrzebojów] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
ALTER DATABASE [ListaPrzebojów] SET DELAYED_DURABILITY = DISABLED
GO
ALTER DATABASE [ListaPrzebojów] SET QUERY_STORE = OFF
GO

```

```

USE [ListaPrzebojów]
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
USE [ListaPrzebojów]
GO
/***** Object:  User [Administratorka]    Script Date: 23.05.2020 17:59:36
*****/
CREATE USER [Administratorka] FOR LOGIN [ListaPrzebojowKonto2] WITH
DEFAULT_SCHEMA=[dbo]
GO
/***** Object:  User [Administrator]    Script Date: 23.05.2020 17:59:36
*****/
CREATE USER [Administrator] FOR LOGIN [ListaPrzebojowKonto] WITH
DEFAULT_SCHEMA=[dbo]
GO
/***** Object:  Table [dbo].[Oceny]    Script Date: 23.05.2020 17:59:36
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Oceny](
    [IdOceny] [int] NOT NULL,
    [OcenaAdministratora] [char](2) NOT NULL,
    [OcenaAdministratorki] [char](2) NULL,
    [Komentarz] [varchar](100) NULL,
    CONSTRAINT [PK_Oceny] PRIMARY KEY CLUSTERED
(
    [IdOceny] 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
/***** Object:  Table [dbo].[Utwory]    Script Date: 23.05.2020 17:59:36
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Utwory](
    [IdUtworu] [int] NOT NULL,
    [NazwaUtworu] [nvarchar](50) NOT NULL,
    [DataWykonania] [datetime] NULL,
    [OpisUtworu] [nvarchar](100) NULL,
    [Długość] [char](5) NULL,
    [WykonawcaId] [int] NOT NULL,
    [AlbumId] [int] NOT NULL,
    [OcenaId] [int] NULL,
    CONSTRAINT [PK_Utwory] PRIMARY KEY CLUSTERED
(
    [IdUtworu] 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
/***** Object: UserDefinedFunction
[dbo].[f_NajlepszeUtworyOcenaAdministratoraRowna10]    Script Date: 23.05.2020
17:59:36 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[f_NajlepszeUtworyOcenaAdministratoraRowna10] ()
RETURNS TABLE
AS
RETURN
(
SELECT * FROM Utwory INNER JOIN Oceny on
oceny.IdOceny=Utwory.OcenaId
where OcenaAdministratora=10
);

GO
/***** Object: UserDefinedFunction [dbo].[f_OcenaAdministratoraWiekszaNiz5]
Script Date: 23.05.2020 17:59:36 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[f_OcenaAdministratoraWiekszaNiz5] ()
RETURNS TABLE
AS
RETURN
(
SELECT * FROM Utwory INNER JOIN Oceny on
oceny.IdOceny=Utwory.OcenaId
where OcenaAdministratora>5
);

GO
/***** Object: Table [dbo].[Gatunki]    Script Date: 23.05.2020 17:59:36
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Gatunki](
    [IdGatunku] [int] NOT NULL,
    [NazwaGatunku] [nvarchar](20) NOT NULL,
    [DataPowstania] [datetime] NULL,
    [MiejscePowstania] [nvarchar](50) NULL,
    CONSTRAINT [PK_Gatunki] PRIMARY KEY CLUSTERED
(
    [IdGatunku] 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

```



```

/***** Object: Table [dbo].[OtrzymanieNagrody]    Script Date: 23.05.2020
17:59:36 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[OtrzymanieNagrody](
    [IdUtworu] [int] NOT NULL,
    [IdNagrody] [int] NOT NULL,
    [DataOtrzymania] [datetime] NULL,
    CONSTRAINT [PK_OtrzymanieNagrody] PRIMARY KEY CLUSTERED
(
    [IdUtworu] ASC,
    [IdNagrody] 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
/***** Object: Table [dbo].[NalezyDoGatunku]    Script Date: 23.05.2020
17:59:36 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[NalezyDoGatunku](
    [IdUtworu] [int] NOT NULL,
    [IdGatunku] [int] NOT NULL,
    CONSTRAINT [PK_NalezyDoGatunku] PRIMARY KEY CLUSTERED
(
    [IdUtworu] ASC,
    [IdGatunku] 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
/***** Object: Table [dbo].[Wykonawcy]    Script Date: 23.05.2020 17:59:36
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Wykonawcy](
    [IdWykonawcy] [int] NOT NULL,
    [Wykonawca] [nvarchar](50) NOT NULL,
    [DataUrodzeniaPowstania] [date] NULL,
    [RolaWykonawcy] [varchar](50) NULL,
    CONSTRAINT [PK_Wykonawcy] PRIMARY KEY CLUSTERED
(
    [IdWykonawcy] 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
/***** Object: Table [dbo].[Nagrody]    Script Date: 23.05.2020 17:59:36
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
CREATE TABLE [dbo].[Nagrody](
    [IdNagrody] [int] NOT NULL,
    [NazwaNagrody] [nvarchar](50) NULL,
    [Kategoria] [varchar](50) NULL,
    [PierwszeWręczenie] [datetime] NULL,
    CONSTRAINT [PK_Nagrody] PRIMARY KEY CLUSTERED
(
    [IdNagrody] 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
/***** Object: Table [dbo].[Albumy]    Script Date: 23.05.2020 17:59:36
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Albumy](
    [IdAlbum] [int] NOT NULL,
    [NazwaAlbumu] [nvarchar](50) NOT NULL,
    [DataWydania] [datetime] NULL,
    [RozpoczęcieNagrań] [datetime] NULL,
    [Wydawnictwo] [nvarchar](100) NULL,
    CONSTRAINT [PK_Albumy] PRIMARY KEY CLUSTERED
(
    [IdAlbum] 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
/***** Object: UserDefinedFunction [dbo].[f_WszystkieDane]    Script Date:
23.05.2020 17:59:36 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[f_WszystkieDane] ()
RETURNS TABLE
AS
RETURN
(
    SELECT
u.IdUtworu,u.AlbumId,u.WykonawcaId,u.OcenaId,ndo.IdGatunku,ng.IdNagrody,
u.NazwaUtworu,u.DataWykonania,u.Długość,u.OpisUtworu,
o.OcenaAdministradora,o.OcenaAdministratorki,o.Komentarz,
a.NazwaAlbumu,a.RozpoczęcieNagrań,a.DataWydania,a.Wydawnictwo,
w.Wykonawca,w.DataUrodzeniaPowstania,w.RolaWykonawcy,
g.NazwaGatunku,g.DataPowstania,g.MiejscePowstania,
n.NazwaNagrody,n.PierwszeWręczenie,n.Kategoria
FROM Utwory u JOIN oceny o ON u.IdUtworu=o.IdOceny
JOIN Albumy a ON a.IdAlbum=u.AlbumId
JOIN Wykonawcy w ON w.IdWykonawcy=u.WykonawcaId
JOIN NależyDoGatunku ndo ON ndo.IdUtworu=u.IdUtworu
JOIN Gatunki g ON g.IdGatunku=ndo.IdGatunku
JOIN OtrzymanieNagrody ng ON ng.IdUtworu=u.IdUtworu
JOIN Nagrody n ON n.IdNagrody=ng.IdNagrody

```

```

        );
GO
/***** Object: UserDefinedFunction [dbo].[f_UtworyZNagrodami]    Script Date:
23.05.2020 17:59:36 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[f_UtworyZNagrodami] ()
RETURNS TABLE
AS
RETURN
(

SELECT
u.IdUtworu,u.NazwaUtworu,u.DataWykonania,u.Długość,u.OpisUtworu,n.NazwaNagrod
y,onag.DataOtrzymania FROM Utwory u INNER JOIN OtrzymanieNagrody onag on
onag.IdUtworu=u.IdUtworu
INNER JOIN Nagrody n on n.IdNagrody=onag.IdNagrody
where onag.IdUtworu IS NOT NULL
        );
GO
/***** Object: UserDefinedFunction [dbo].[f_WykonawcyIIchAlbumy]    Script
Date: 23.05.2020 17:59:36 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[f_WykonawcyIIchAlbumy] ()
RETURNS TABLE
AS
RETURN
(
SELECT w.Wykonawca, a.NazwaAlbumu, a.RozpoczęcieNagrań, a.DataWydania FROM
Wykonawcy w inner join Utwory u on u.WykonawcaId=w.IdWykonawcy
INNER JOIN Albumy a on a.IdAlbum=u.AlbumId
        );
GO
/***** Object: UserDefinedFunction [dbo].[f_wykonawcyIIchGatunkui]    Script
Date: 23.05.2020 17:59:36 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[f_wykonawcyIIchGatunkui] ()
RETURNS TABLE
AS
RETURN
(
SELECT w.Wykonawca, g.NazwaGatunku, g.MiejscePowstania, g.DataPowstania FROM
Wykonawcy w inner join Utwory u on u.WykonawcaId=w.IdWykonawcy
INNER JOIN NalezyDoGatunku a on a.IdUtworu=u.IdUtworu
INNER JOIN Gatunki g on g.IdGatunku=a.IdGatunku
        );
GO

```

```

ALTER TABLE [dbo].[NalezyDoGatunku] WITH CHECK ADD CONSTRAINT
[FK_NalezyDoGatunku_Gatunki] FOREIGN KEY([IdGatunku])
REFERENCES [dbo].[Gatunki] ([IdGatunku])
GO
ALTER TABLE [dbo].[NalezyDoGatunku] CHECK CONSTRAINT
[FK_NalezyDoGatunku_Gatunki]
GO
ALTER TABLE [dbo].[NalezyDoGatunku] WITH CHECK ADD CONSTRAINT
[FK_NalezyDoGatunku_Utwory] FOREIGN KEY([IdUtworu])
REFERENCES [dbo].[Utwory] ([IdUtworu])
GO
ALTER TABLE [dbo].[NalezyDoGatunku] CHECK CONSTRAINT
[FK_NalezyDoGatunku_Utwory]
GO
ALTER TABLE [dbo].[OtrzymanieNagrody] WITH CHECK ADD CONSTRAINT
[FK_OtrzymanieNagrody_Nagrody] FOREIGN KEY([IdNagrody])
REFERENCES [dbo].[Nagrody] ([IdNagrody])
GO
ALTER TABLE [dbo].[OtrzymanieNagrody] CHECK CONSTRAINT
[FK_OtrzymanieNagrody_Nagrody]
GO
ALTER TABLE [dbo].[OtrzymanieNagrody] WITH CHECK ADD CONSTRAINT
[FK_OtrzymanieNagrody_Utwory] FOREIGN KEY([IdUtworu])
REFERENCES [dbo].[Utwory] ([IdUtworu])
GO
ALTER TABLE [dbo].[OtrzymanieNagrody] CHECK CONSTRAINT
[FK_OtrzymanieNagrody_Utwory]
GO
ALTER TABLE [dbo].[Utwory] WITH CHECK ADD CONSTRAINT [FK_Utwory_Albumy]
FOREIGN KEY([AlbumId])
REFERENCES [dbo].[Albumy] ([IdAlbum])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Utwory] CHECK CONSTRAINT [FK_Utwory_Albumy]
GO
ALTER TABLE [dbo].[Utwory] WITH CHECK ADD CONSTRAINT [FK_Utwory_Oceny]
FOREIGN KEY([OcenaId])
REFERENCES [dbo].[Oceny] ([IdOceny])
GO
ALTER TABLE [dbo].[Utwory] CHECK CONSTRAINT [FK_Utwory_Oceny]
GO
ALTER TABLE [dbo].[Utwory] WITH CHECK ADD CONSTRAINT [FK_Utwory_Wykonawcy]
FOREIGN KEY([WykonawcaId])
REFERENCES [dbo].[Wykonawcy] ([IdWykonawcy])
GO
ALTER TABLE [dbo].[Utwory] CHECK CONSTRAINT [FK_Utwory_Wykonawcy]
GO
USE [master]
GO
ALTER DATABASE [ListaPrzebojów] SET READ_WRITE
GO

```

Plik z zadaniem można znaleźć także na repozytorium:

<https://github.com/Jarverr/ProjektBazyDanych>

## 7. 8. Definicja niedeklaratywnych mechanizmów sprawdzania poprawności danych

Wyzwalacz mający za cel sprawdzać, czy aby data urodzenia nie jest późniejsza niż data dzisiejszego dnia:

```
USE [ListaPrzebojów]
GO
/***** Object: Trigger [dbo].[tooLateDataErrorWykonawcy]    Script Date: 23.05.2020
16:05:33 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[tooLateDataErrorWykonawcy] ON [dbo].[Wykonawcy]
FOR INSERT
NOT FOR REPLICATION
AS
IF(SELECT COUNT(*) FROM inserted)>=1
BEGIN
    IF((SELECT DataUrodzeniaPowstania from inserted)>GETDATE())
    BEGIN
        RAISERROR('Data urodzenia/ powstania zespołu nie może być późniejsza niż
data dzisiejsza',16,1)
        ROLLBACK TRANSACTION
    END
END
```

Kolejny trigger opisujący podobnie jak poprzedni co ma się wydarzyć gdy data wydania utworu będzie późniejsza od daty dzisiejszej:

```
USE [ListaPrzebojów]
GO
/***** Object: Trigger [dbo].[DateIsNotCorrect]    Script Date: 23.05.2020 16:05:08
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[DateIsNotCorrect] ON [dbo].[Utworky]
FOR INSERT
NOT FOR REPLICATION
AS
IF (SELECT COUNT(*) FROM inserted)=1
BEGIN
    IF(Select YEAR(i.DataWykonanania)from inserted i)>=YEAR(GETDATE())
    BEGIN
        RAISERROR('Ta data wykonania jeszcze nie została osiągnięta przez
czas',16,1)
        ROLLBACK TRANSACTION
    END
END
```

Analogicznie dla daty w tabeli nagrody

```
USE [ListaPrzebojów]
GO
/***** Object: Trigger [dbo].[tooLateDataErrorNagrody]    Script Date: 23.05.2020
16:02:51 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[tooLateDataErrorNagrody] ON [dbo].[Nagrody]
FOR INSERT
NOT FOR REPLICATION
AS
IF(SELECT COUNT(*) FROM inserted)>=1
BEGIN
    IF((SELECT PierwszeWręczenie from inserted)>GETDATE())
    BEGIN
        RAISERROR('Data pierwszego wręczenia nie może być późniejsza niż data
dzisiejsza',16,1)
        ROLLBACK TRANSACTION
    END
END
END
```

Podoba zasada odnośnie dat w tabeli gatunki:

```
USE [ListaPrzebojów]
GO
/***** Object: Trigger [dbo].[tooLateDataErrorGatunki]    Script Date: 23.05.2020
16:03:06 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[tooLateDataErrorGatunki] ON [dbo].[Gatunki]
FOR INSERT
NOT FOR REPLICATION
AS
IF(SELECT COUNT(*) FROM inserted)>=1
BEGIN
    IF((SELECT DataPowstania from inserted)>GETDATE())
    BEGIN
        RAISERROR('Data powstania nie może być późniejsza niż data
dzisiejsza',16,1)
        ROLLBACK TRANSACTION
    END
END
END
```

Dla albumów musi sprawdzić dwie daty:

```
USE [ListaPrzebojów]
GO
/***** Object: Trigger [dbo].[tooLateDataErrorAlbums]    Script Date: 23.05.2020
16:04:03 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[tooLateDataErrorAlbums] ON [dbo].[Albumy]
FOR INSERT
NOT FOR REPLICATION
AS
IF(SELECT COUNT(*) FROM inserted)>=1
    BEGIN
        IF((SELECT DataWydania from inserted)>GETDATE())
        BEGIN
            RAISERROR('Data wydania nie może być późniejsza niż data
dzisiejsza',16,1)
            ROLLBACK TRANSACTION
        END
        IF((SELECT RozpoczęcieNagrań from inserted)>GETDATE())
        BEGIN
            RAISERROR('Data rozpoczęcia nagrań nie może być późniejsza niż data
dzisiejsza',16,1)
            ROLLBACK TRANSACTION
        END
    END
```

Wartości będące zakazanymi jako puste, zostały zdefiniowane w czasie tworzenia bazy danych – poprzez dodania do tworzonej kolumny dopisku NOT NULL (podobnie jak albumy każda inna tabela dostała podobną właściwość dla odpowiednich kolumn).

```
CREATE TABLE [dbo].[Albumy](
    [IdAlbum] [int] NOT NULL,
    [NazwaAlbumu] [nvarchar](50) NOT NULL,
    [DataWydania] [datetime] NULL,
    [RozpoczęcieNagrań] [datetime] NULL,
    [Wydawnictwo] [nvarchar](100) NULL,
    CONSTRAINT [PK_Albumy] PRIMARY KEY CLUSTERED
(
    [IdAlbum] 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
```

W pliku zadanie7.sql (oraz poniżej) znajduje się kod dodatkowy, który prezentuje co byłoby wymagane w razie potrzeby modyfikacji własności null na not null dla tabel:

```
ALTER TABLE dbo.Utwory ALTER COLUMN [NazwaUtworu] [nvarchar] NOT NULL
GO
ALTER TABLE dbo.Gatunki ALTER COLUMN [NazwaGatunku] [nvarchar] NOT NULL
GO
ALTER TABLE dbo.Albumy ALTER COLUMN [NazwaAlbumu] [nvarchar] NOT NULL
GO
ALTER TABLE dbo.Wykonawcy ALTER COLUMN Wykonawca [nvarchar] NOT NULL
GO
```

## 9. Kod wspomagający aplikację użytkową

Poniżej kod, który ma wspomóc tworzenie aplikacji użytkowej (kod znajduje się także w pliku zadanie9.sql)

Funkcja która wyświetla utwory o ocenie większej niż 5:

```
DROP FUNCTION IF EXISTS dbo.f_OcenaAdministradoraWiekszaNiz5
GO
CREATE FUNCTION f_OcenaAdministradoraWiekszaNiz5 ()
RETURNS TABLE
AS
RETURN
(
    SELECT * FROM Utwory INNER JOIN Oceny on
    oceny.IdOceny=Utwory.OcenaId
    where OcenaAdministradora>5
);

GO
```

Funkcja wyświetlająca oceny na poziomie 10:

```
DROP FUNCTION IF EXISTS dbo.f_NajlepszeUtworyOcenaAdministradoraRowna10
GO
CREATE FUNCTION f_NajlepszeKawałki ()
RETURNS TABLE
AS
RETURN
(
    SELECT * FROM Utwory INNER JOIN Oceny on
    oceny.IdOceny=Utwory.OcenaId
    where OcenaAdministradora=10
);

GO
```

Funkcja wyświetlająca wszystkie dane z bazy danych:

```
DROP FUNCTION IF EXISTS dbo.f_WszystkieDane
GO
CREATE FUNCTION f_WszystkieDane ()
RETURNS TABLE
AS
RETURN
(
    SELECT
    u.IdUtworu,u.AlbumId,u.WykonawcaId,u.OcenaId,ndo.IdGatunku,ng.IdNagrody,
    u.NazwaUtworu,u.DataWykonania,u.Długość,u.OpisUtworu,
    o.OcenaAdministradora,o.OcenaAdministratorki,o.Komentarz,
    a.NazwaAlbumu,a.RozpoczęcieNagrań,a.DataWydania,a.Wydawnictwo,
    w.Wykonawca,w.DataUrodzeniaPowstania,w.RolaWykonawcy,
    g.NazwaGatunku,g.DataPowstania,g.MiejscePowstania,
    n.NazwaNagrody,n.PierwszeWręczenie,n.Kategoria
    FROM Utwory u JOIN oceny o ON u.IdUtworu=o.IdOceny
    JOIN Albumy a ON a.IdAlbum=u.AlbumId
    JOIN Wykonawcy w ON w.IdWykonawcy=u.WykonawcaId
    JOIN NależyDoGatunku ndo ON ndo.IdUtworu=u.IdUtworu
    JOIN Gatunki g ON g.IdGatunku=ndo.IdGatunku
    JOIN OtrzymanieNagrody ng ON ng.IdUtworu=u.IdUtworu
    JOIN Nagrody n ON n.IdNagrody=ng.IdNagrody
);
```



```
);  
GO
```

Funkcja wyświetlająca utwory, które otrzymały jakąkolwiek nagrodę:

```
DROP FUNCTION IF EXISTS dbo.f_UtworyZNagrodami  
GO  
CREATE FUNCTION f_UtworyZNagrodami ()  
RETURNS TABLE  
AS  
RETURN  
(  
  
SELECT  
u.IdUtworu,u.NazwaUtworu,u.DataWykonania,u.Długość,u.OpisUtworu,n.NazwaNagrody,onag.  
DataOtrzymania FROM Utwory u INNER JOIN OtrzymanieNagrody onag on  
onag.IdUtworu=u.IdUtworu  
INNER JOIN Nagrody n on n.IdNagrody=onag.IdNagrody  
where onag.IdUtworu IS NOT NULL  
);  
GO
```

Funkcja wyświetlająca wykonawcę i jego albumy:

```
DROP FUNCTION IF EXISTS dbo.f_WykonawcyIIchAlbumy  
GO  
CREATE FUNCTION f_WykonawcyIIchAlbumy ()  
RETURNS TABLE  
AS  
RETURN  
(  
SELECT w.Wykonawca, a.NazwaAlbumu, a.RozpoczęcieNagrań, a.DataWydania FROM Wykonawcy w  
inner join Utwory u on u.WykonawcaId=w.IdWykonawcy  
INNER JOIN Albumy a on a.IdAlbum=u.AlbumId  
  
);  
GO
```

Funkcja wyświetlająca gatunki, w których tworzył/ tworzyli wykonawcy:

```
DROP FUNCTION IF EXISTS dbo.f_wykonawcyIIchGatunki  
GO  
CREATE FUNCTION f_wykonawcyIIchGatunki ()  
RETURNS TABLE  
AS  
RETURN  
(  
SELECT w.Wykonawca, g.NazwaGatunku, g.MiejscePowstania, g.DataPowstania FROM Wykonawcy  
w inner join Utwory u on u.WykonawcaId=w.IdWykonawcy  
INNER JOIN NależyDoGatunku a on a.IdUtworu=u.IdUtworu  
INNER JOIN Gatunki g on g.IdGatunku=a.IdGatunku  
  
);  
GO
```

## 10 Wprowadzenie przykładowych danych

Dane wprowadzić można poprzez polecenie BULK INSERT np.

```
19 BULK INSERT Albumy
20 FROM 'D:\Test.txt'
21 WITH (
22     FIELDTERMINATOR=';',
23     ROWTERMINATOR='\n'
24     --CODEPAGE='ACP'
25 )
26
27 SELECT * FROM Albumy
28 DELETE FROM Albumy where IdAlbum=1 OR IdAlbum=2
```

98 %

Results Messages

	IdAlbum	NazwaAlbu...	DataWydania	RozpoczęcieNagrań	Wydawnictwo
1	1	The Art of War	2008-05-30 00:00:00.0...	2007-08-01 00:00:00.0...	Sabatón and Tommy & Peter T ingtr...
2	2	Tutu	1986-09-01 00:00:00.0...	NULL	Marcus MillerTommy LiPuma

Pliki zawierające przykładowe dane znajdują pod nazwami Albumy.txt, Wykonawcy.txt etc. Przy dodawaniu ustawione są też dla utworu odpowiednie id albumu, id wykonawcy itd. Dla tego utwory zalecane aby dodać na końcu by referencja mogła powstać.

Dodatkowo przygotowany jest kod wstawiający w pliku zadanie10.sql:

```
USE ListaPrzebojów
GO
--GATUNKI
INSERT INTO Gatunki VALUES (1, 'Metal', '19650101', 'USA/UK'), (2, 'Jazz', '19100101', 'Nowy
Orlean');

--ALBUMY
INSERT INTO Albumy VALUES (1, 'Tha art of War', '20080530', '20070801', 'Black Lodge
Records'), (2, 'Tutu', '19860901', Null, 'Marcus MillerTommy LiPuma');

--NAGRODY
INSERT INTO Nagrody VALUES(1, 'Grammis Award for Hard Rock/Metal of the
Year', NULL, '19900221'), (2, 'Nagroda Grammy', 'Najlepsza solowa improwizacja
jazzowa', '19590101');

--WYKONAWCY
INSERT INTO Wykonawcy VALUES (1, 'Sabaton', '19990101', 'Zespół'), (2, 'Miles
Davis', '19260526', 'trebacz, lider, kompozytor')

--OCENY
INSERT INTO Oceny VALUES (1, 9, 3, 'Dobry kawalek o mestwie i wojnie'), (2, 9, 7, 'Stare
przeboje');

--UTWORY
INSERT INTO Utwory VALUES (
1, '40:1', '20080520', NULL, '4:11', 1, 1, 1), (2, 'Tutu', '19860901', 'Stary jazzowy kawalek, z
trabka', '5:18', 2, 2, 2);
```

```
INSERT INTO NalezyDoGatunku VALUES (1,1),(2,2)
INSERT INTO OtrzymanieNagrody VALUES (1,1),(2,2)
```

Pliki znaleźć można pod linkiem : <https://github.com/Jarverr/ProjektBazyDanych> (repozytorium).