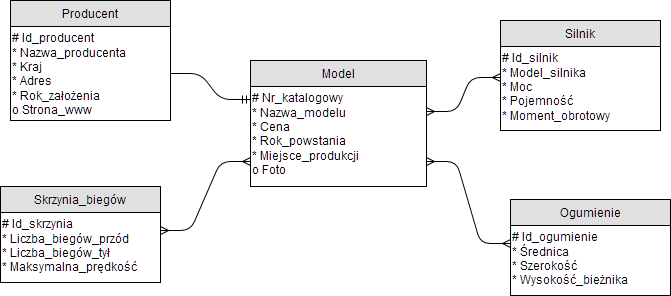
1. **Treść zadania**

Baza danych zrealizowana w projekcie przedstawia katalog ciągników rolniczych. Są one uszeregowane w zależności od modelu ciągnika i znajdują się w tabeli głównej Model. Zawiera ona atrybuty: Nr\_katalogowy (klucz podstawowy), Nazwa\_modelu, Cena, Rok\_powstania, Miejsce produkcji i Foto. Z tabelą Model związane są tabele o następujących atrybutach:

* Producent: Id\_producent (klucz podstawowy), Nazwa\_producenta, Kraj, Adres, Rok\_założenia, Strona\_www,
* Silnik: Id\_silnik (klucz podstawowy), Model\_silnika, Moc, Pojemność, Moment\_obrotowy,
* Skrzynia\_biegów: Id\_skrzynia (klucz podstawowy), Liczba\_biegów\_przód, Liczba\_biegów\_tył, Maksymalna\_prędkość,
* Ogumienie: Id\_ogumienie, Średnica, Szerokość, Wysokość\_bieżnika.

1. **Diagram E/R bazy danych w notacji Barkiera (wykonany ręcznie w edytorze tekstowym). Do diagramu należy dodać także krótkie uzasadnienie**

****

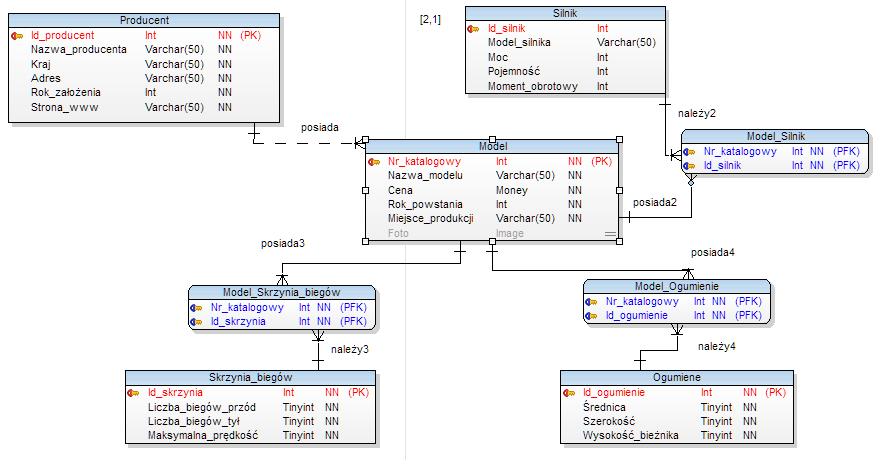
Rys. 1. Diagram E/R bazy danych Ciągniki rolnicze w notacji Barkera

Do utworzenia diagramu bazy danych Ciągniki rolnicze wykorzystano następujące założenia:

* Każdy model musi należeć do konkretnego producenta, producent posiada wiele modeli. Nie istnieją producenci, którzy nie posiadają modeli (związek jeden do wielu, 1:M).
* Ten sam model może posiadać różne rodzaje silników, ten sam silnik może być zamontowany w różnych modelach (związek wiele do wielu, M:N). Każdy model posiada jeden silnik, a każdy silnik należy do jakiegoś modelu.
* Ten sam model może posiadać różne rodzaje skrzyni biegów, ta sama skrzynia biegów może być zamontowana w różnych modelach (związek wiele do wielu, M:N). Każdy model posiada jedną skrzynię biegów, a każdy skrzynia biegów należy do jakiegoś modelu.
* Ten sam model może posiadać różne rodzaje ogumienia, to samo ogumienie może być zamontowane w różnych modelach (związek wiele do wielu, M:N). Każdy model posiada jedno ogumienie, a każdy ogumienie należy do jakiegoś modelu.

1. **Diagram E/R bazy danych wykonany za pomocą jednego z narzędzi CASE. Każdy atrybut encji powinien być przedstawiony ze swoim typem danych lub mają być przedstawione okna właściwości tych atrybutów**

W bazie danych Ciągniki rolnicze występują 3 wiązki wiele do wielu. Dlatego w celu prawidłowego przedstawienia relacji w diagramie niezbędne było zastosowanie tabel pośrednich, które zawierają klucze podstawowe z tabeli nadrzędnych, które są obcymi w tabelach pośrednich. Do przedstawienia związku pomiędzy tabelą Model a tabelą Silnik utworzono tabelę pośrednią Model\_Silnik. Do przedstawienia związku pomiędzy tabelą Model a tabelą Skrzynia\_biegów utworzono tabelę pośrednią Model\_Skrzynia\_biegów. Do przedstawienia związku pomiędzy tabelą Model a tabelą Ogumienie utworzono tabelę pośrednią Model\_Ogumienie.

Ze względu na mnogość wystąpień relacji M:N, jako narzędzie CASE wykorzystano oprogramowanie Toad Data Modeler 6.3. Umożliwia ono łatwe określanie tego typu związków poprzez automatyczne wstawianie tabel pośrednich. Ponadto w tym środowisku można w łatwy sposób edytować rodzaje związków oraz ich opcjonalność lub obowiązkowość. 

Rys. 2. Diagram E/R bazy danych Ciągniki rolnicze w notacji Barkera wykonany za pomocą narzędzia CASE Toad Data Modeler 6.3

**4. Skrypt Transact SQL bazy danych**

Za pomocą Toad Data Modeler wygenerowano następujący skrypt SQL:

/\*

Created: 2017-12-15

Modified: 2018-01-10

Model: Microsoft SQL Server 2014

Database: MS SQL Server 2014

\*/

-- Create tables section -------------------------------------------------

-- Table Producent

CREATE TABLE [Producent]

(

[Id\_producent] Int IDENTITY NOT NULL,

[Nazwa\_producenta] Varchar(50) NOT NULL,

[Kraj] Varchar(50) NOT NULL,

[Adres] Varchar(50) NOT NULL,

[Rok\_założenia] Int NOT NULL,

[Strona\_www] Varchar(50) NOT NULL

)

go

-- Add keys for table Producent

ALTER TABLE [Producent] ADD CONSTRAINT [Key1] PRIMARY KEY ([Id\_producent])

go

-- Table Model

CREATE TABLE [Model]

(

[Nr\_katalogowy] Int IDENTITY NOT NULL,

[Nazwa\_modelu] Varchar(50) NOT NULL,

[Cena] Money NOT NULL,

[Rok\_powstania] Int NOT NULL,

[Miejsce\_produkcji] Varchar(50) NOT NULL,

[Foto] Image NULL,

[Id\_producent] Int NOT NULL

)

go

-- Create indexes for table Model

CREATE INDEX [IX\_Relationship1] ON [Model] ([Id\_producent])

go

-- Add keys for table Model

ALTER TABLE [Model] ADD CONSTRAINT [Key2] PRIMARY KEY ([Nr\_katalogowy])

go

-- Table Silnik

CREATE TABLE [Silnik]

(

[Id\_silnik] Int IDENTITY NOT NULL,

[Model\_silnika] Varchar(50) NOT NULL,

[Moc] Int NOT NULL,

[Pojemność] Int NOT NULL,

[Moment\_obrotowy] Int NOT NULL

)

go

-- Add keys for table Silnik

ALTER TABLE [Silnik] ADD CONSTRAINT [Key3] PRIMARY KEY ([Id\_silnik])

go

-- Table Skrzynia\_biegów

CREATE TABLE [Skrzynia\_biegów]

(

[Id\_skrzynia] Int IDENTITY NOT NULL,

[Liczba\_biegów\_przód] Tinyint NOT NULL,

[Liczba\_biegów\_tył] Tinyint NOT NULL,

[Maksymalna\_prędkość] Tinyint NOT NULL

)

go

-- Add keys for table Skrzynia\_biegów

ALTER TABLE [Skrzynia\_biegów] ADD CONSTRAINT [Key4] PRIMARY KEY ([Id\_skrzynia])

go

-- Table Ogumiene

CREATE TABLE [Ogumiene]

(

[Id\_ogumienie] Int IDENTITY NOT NULL,

[Średnica] Tinyint NOT NULL,

[Szerokość] Tinyint NOT NULL,

[Wysokość\_bieżnika] Tinyint NOT NULL

)

go

-- Add keys for table Ogumiene

ALTER TABLE [Ogumiene] ADD CONSTRAINT [Key5] PRIMARY KEY ([Id\_ogumienie])

go

-- Table Model\_Silnik

CREATE TABLE [Model\_Silnik]

(

[Nr\_katalogowy] Int NOT NULL,

[Id\_silnik] Int NOT NULL

)

go

-- Add keys for table Model\_Silnik

ALTER TABLE [Model\_Silnik] ADD CONSTRAINT [Key7] PRIMARY KEY ([Nr\_katalogowy],[Id\_silnik])

go

-- Table Model\_Ogumienie

CREATE TABLE [Model\_Ogumienie]

(

[Nr\_katalogowy] Int NOT NULL,

[Id\_ogumienie] Int NOT NULL

)

go

-- Add keys for table Model\_Ogumienie

ALTER TABLE [Model\_Ogumienie] ADD CONSTRAINT [Key8] PRIMARY KEY ([Nr\_katalogowy],[Id\_ogumienie])

go

-- Table Model\_Skrzynia\_biegów

CREATE TABLE [Model\_Skrzynia\_biegów]

(

[Nr\_katalogowy] Int NOT NULL,

[Id\_skrzynia] Int NOT NULL

)

go

-- Add keys for table Model\_Skrzynia\_biegów

ALTER TABLE [Model\_Skrzynia\_biegów] ADD CONSTRAINT [Key9] PRIMARY KEY ([Nr\_katalogowy],[Id\_skrzynia])

go

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

ALTER TABLE [Model\_Silnik] ADD CONSTRAINT [posiada2] FOREIGN KEY ([Nr\_katalogowy]) REFERENCES [Model] ([Nr\_katalogowy]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [Model\_Silnik] ADD CONSTRAINT [należy2] FOREIGN KEY ([Id\_silnik]) REFERENCES [Silnik] ([Id\_silnik]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [Model\_Ogumienie] ADD CONSTRAINT [posiada4] FOREIGN KEY ([Nr\_katalogowy]) REFERENCES [Model] ([Nr\_katalogowy]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [Model\_Ogumienie] ADD CONSTRAINT [należy4] FOREIGN KEY ([Id\_ogumienie]) REFERENCES [Ogumiene] ([Id\_ogumienie]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [Model\_Skrzynia\_biegów] ADD CONSTRAINT [posiada3] FOREIGN KEY ([Nr\_katalogowy]) REFERENCES [Model] ([Nr\_katalogowy]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [Model\_Skrzynia\_biegów] ADD CONSTRAINT [należy3] FOREIGN KEY ([Id\_skrzynia]) REFERENCES [Skrzynia\_biegów] ([Id\_skrzynia]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [Model] ADD CONSTRAINT [posiada] FOREIGN KEY ([Id\_producent]) REFERENCES [Producent] ([Id\_producent]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

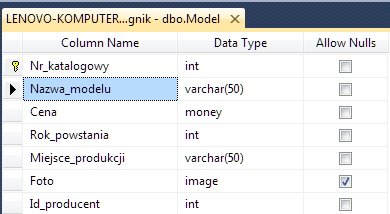
**5. Widok bazy danych w oknie programu MS SQL Management Studio, w tym zdjęcia tabel w widoku projektu oraz widoku danych. (Należy wprowadzić do tabel przykładowe dane). Dla każdej tabeli także należy podać, jakie ma wartości domyślne, indeksy, klucze obce**

Po utworzeniu pustej bazy danych Projekt\_ciągniki wygenerowano za pomocą skryptu polecenia Query zawartość bazy. Wykorzystano do tego powyższy skrypt, który zmodyfikowano dodając na początku nazwę bazy danych poleceniami:

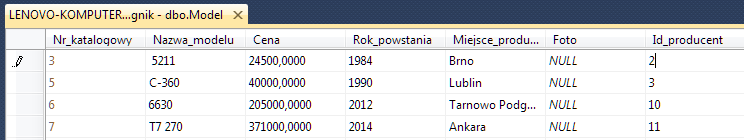
USE Projekt\_ciagnik

GO

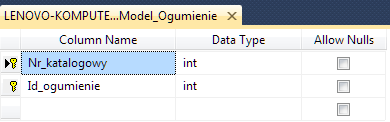
Do tabel wprowadzono dane i tabele przedstawiono w widoku projektu oraz widoku danych.



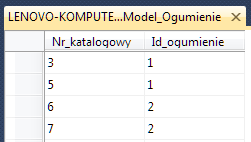
Rys. 3. Tabela Model- widok projektu



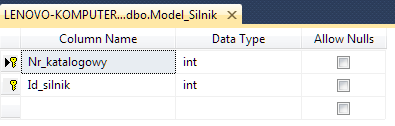
Rys. 4. Tabela Model- widok danych



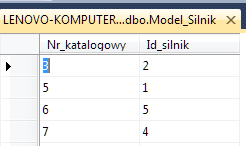
Rys. 5. Tabela Model\_Ogumienie- widok projektu



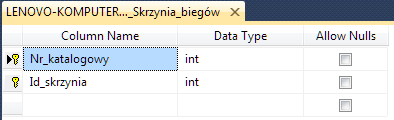
Rys. 6. Tabela Model\_Ogumienie- widok danych



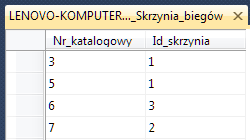
Rys. 7. Tabela Model\_Silnik- widok projektu



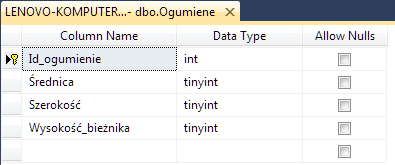
Rys. 8. Tabela Model\_Silnik- widok danych



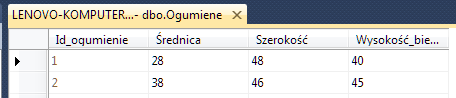
Rys. 9. Tabela Model\_Skrzynia\_biegów- widok projektu



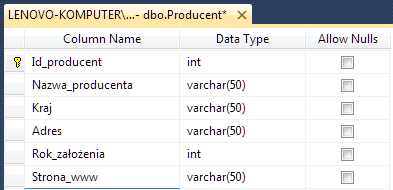
Rys. 10. Tabela Model\_Skrzynia\_biegów- widok danych



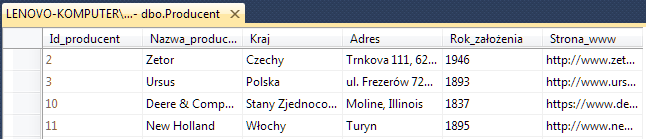
Rys. 11. Tabela Ogumienie- widok projektu



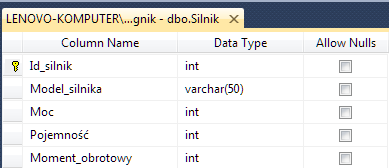
Rys. 12. Tabela Ogumienie- widok danych



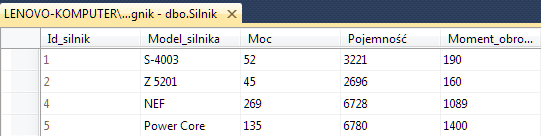
Rys. 13. Tabela Producent- widok projektu



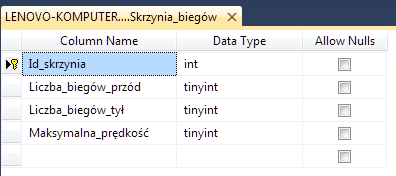
Rys. 14. Tabela Producent- widok danych



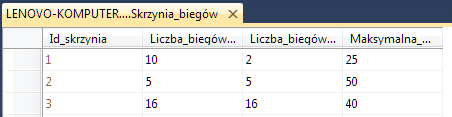
Rys. 15. Tabela Silnik- widok projektu



Rys. 16. Tabela Silnik- widok danych



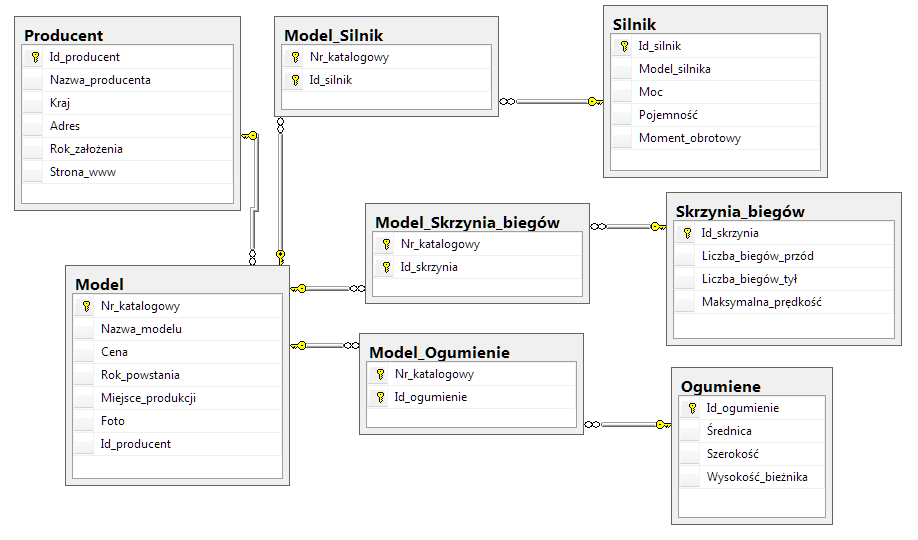
Rys. 17. Tabela Skrzynia\_biegów- widok projektu



Rys. 18. Tabela Skrzynia\_biegów- widok danych

**6. Diagram bazy danych wykonany w programie MS SQL Management Studio**

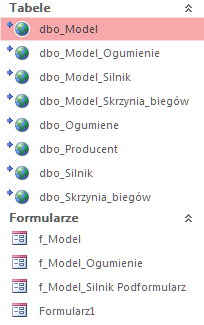
Diagram baz danych został utworzony automatycznie po uruchomieniu skryptu i dodaniu tabel do diagramu. Relacje są zgodne z założeniami projektowymi i diagramem utworzonym w narzędziu CASE.



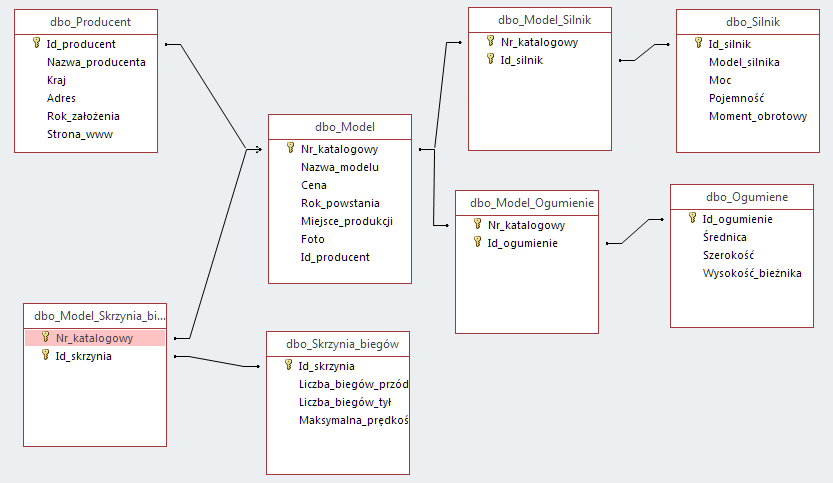
Rys. 19. Diagram bazy danych Ciągniki rolnicze wykonany w programie MS SQL Management Studio

**7. Zdjęcia formularzy aplikacji klienckiej pozwalającej co najmniej na przegląd danych oraz ich wyszukiwanie według różnych kryteriów**

Aby utworzyć aplikację klienta bazy danych należało utworzyć połączenie pomiędzy Microsoft Access 2016 i SQL Server 2014. W tym celu w Accessie utworzono pustą bazę danych i połączono ją z bazą danych na SQL Serverze korzystając z protokołu ODBC. Następnie zaimportowano do niej dane z SQL Servera oraz utworzono diagram relacji baz danych.



Rys. 20. Widok tabeli i formularzy w Access 2016

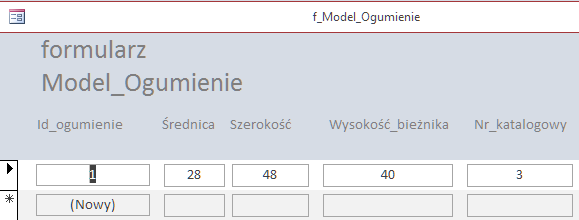


Rys. 21. Diagram bazy danych Ciągniki rolnicze wykonany w programie Access 2016

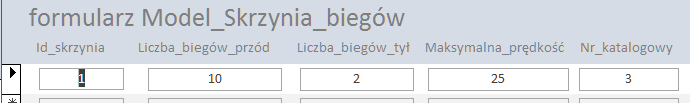
Następnie za pomocą kreatora formularzy utworzono formularz aplikacji klienckiej. W formularzu głównym f\_Model wyświetlono formularz umożliwiający przegląd danych z tabeli Model oraz podformularz pozwalający na wyświetlenie danych z tabeli Silnik. Do formularza dodano opcję wyszukiwania po polu Nazwa\_modelu oraz trzy przyciski przenoszące użytkownika do formularzy z tabeli Ogumienie, Skrzynia\_biegów i Producent.



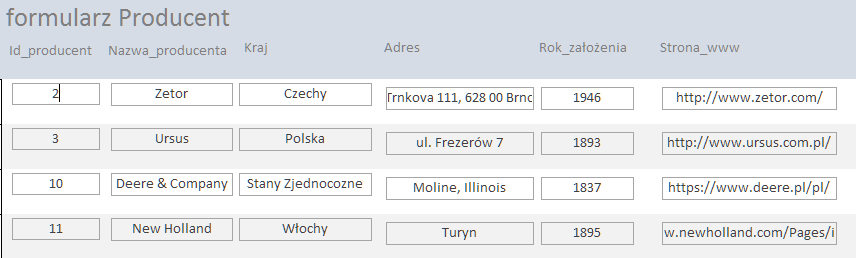
Rys. 22. Formularz Model z podformularzem Silnik



Rys. 23. Formularz Model\_Ogumienie



Rys. 24. Formularz Model\_Skrzynia\_Biegów



Rys. 25. Formularz Producent

**8. Wnioski**

Udało się poprawnie zrealizować zadanie projektowe. Utworzono diagram bazy danych E/R w notacji Barkera. Następnie wykonano bazę danych za pomocą narzędzia CASE w programie Toad Data Modeler. W tym programie wygenerowano również skrypt Transact SQL bazy danych. Następnie przy wykorzystaniu skryptu utworzono bazę danych w MS SQL Management Studio. Tutaj również wprowadzono do tabeli prawidłowe przykładowe dane. Na koniec nawiązano połączenie pomiędzy MS SQL Management Studio a Microsoft Access gdzie zostały utworzone formularze aplikacji klienckiej.

Podczas projektowania bazy danych należy pamiętać aby relacje pomiędzy tabelami odpowiadały rzeczywistości. W przeciwnym razie dane zawarte w tabelach mogą być ze sobą niespójne. Należy również dobrać odpowiedni format w zależności od rodzaju wartości jaką ma reprezentować dane pole. Poprawne utworzenie diagramu E/R pozwala na zaobserwowanie zależności pomiędzy tabelami, a następnie utworzenie aplikacji w oprogramowaniu na podstawie takiego modelu.