

# 1 Tytuł projektu

Ośrodek nauki jazdy dla kierowców

## 2 Opis problemu

Zgodnie z wymaganiami kierownika ośrodka nauki jazdy dla kierowców aplikacja ma za zadanie dostarczać informacji o:

- Kursantach (imię, nazwisko, data urodzenia, numer telefonu, numer dowodu osobistego, adres zamieszkania).
- Instruktorach (imię, nazwisko, data urodzenia, pesel, numer telefonu).
- Licencjach (data wydania, data ważności, kategoria).
- Pojazdach szkoleniowych (rodzaj (np. motocykl, samochód osobowy itd. ), marka, model, rocznik, numer rejestracyjny, data ważności przeglądu technicznego).
- Jazdach szkoleniowych (data, liczba godzin, opis (np. jazda po mieście, jazda po placu manewrowym itd.)).
- Egzaminach (data, rodzaj (np. praktyka / teoria), wynik (zaliczony / niezaliczony))
- Kursach (kategoria, liczba godzin szkolenia, minimalny wiek).

Baza danych stanowi podstawę do opracowania aplikacji ułatwiającej wprowadzanie danych oraz przeszukiwanie jej. Przykładowe kwerendy (zapytania) do bazy:

- Podaj numer telefonów instruktorów, którzy posiadają uprawnienia kategorii D+E dłużej niż 3 lata.
- Podaj numery rejestracyjne samochodów, których przegląd techniczny jest ważny do końca stycznia 2023 roku.
- Podaj imiona oraz nazwiska kursantów, którzy są zapisani na kurs prawa jazdy kategorii B oraz do ukończenia 18 lat zostało im mniej niż jeden miesiąc.
- Podaj liczbę podejść każdego z kursantów do egzaminu praktycznego kategorii B, którzy są w przedziale wiekowym od 18 do 22 lat.
- Podaj listę kursantów, którzy zdali egzamin praktyczny za pierwszym podejściem w bieżącym miesiącu.
- Podaj listę kursantów, którzy biorą udział w kursie kategorii B oraz mieszkają w Krakowie przy ulicy Floriańskiej.

## 3 Model danych

### 3.1 Słownik pojęć

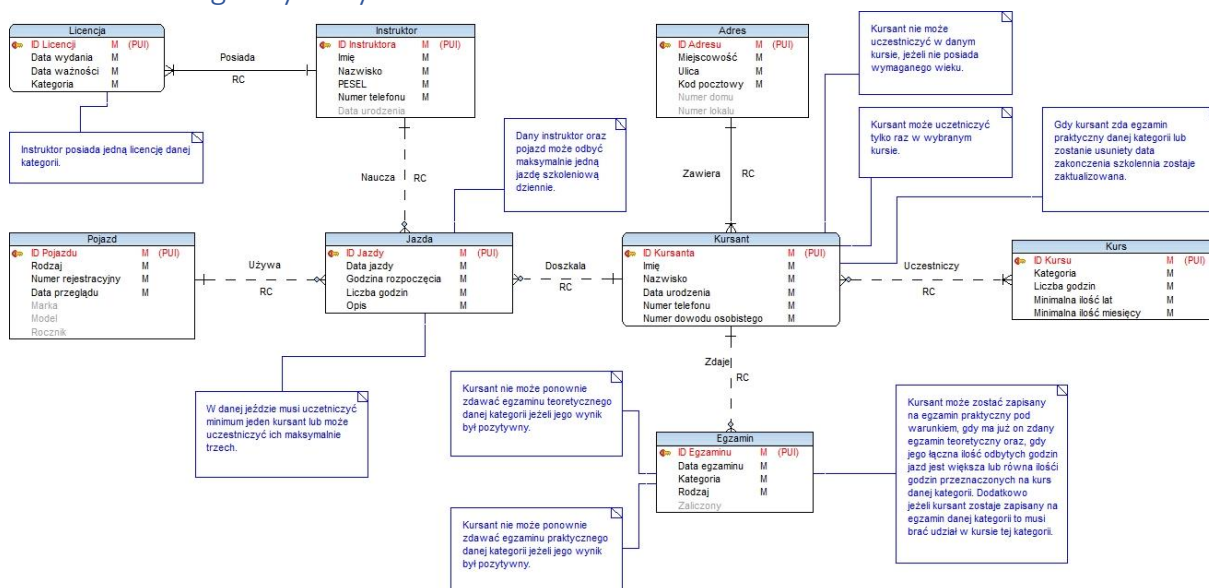
- **Instruktor** = osoba, która posiada uprawnienia oraz licencje do szkolenia kursantów.
- **Licencja** = dokument, upoważniający instruktora do prowadzenia zajęć szkoleniowych.
- **Kursant** = osoba, która uczestniczy w wybranym kursie w celu zdobycia prawa jazdy konkretnej kategorii.
- Kategorie prawa jazdy: **AM, A1, A2, A, B1, B, B+E, C, C1, C1+E, D, D1, D1+E, D+E, T**
- **Jazda szkoleniowa** = forma szkolenia kursantów, w której udział bierze instruktor oraz kursant. Dodatkowo wykorzystywany jest pojazd wybranej kategorii prawa jazdy.
- **Pojazd szkoleniowy** = środek transportu przeznaczony do nauki jazdy przez kursanta oraz instruktora.

### 3.2 Fakty (reguły biznesowe)

Szkolenie kursanta odbywa się w momencie wypełnienia przez niego formularza zgłoszeniowego i wysłania go do kierownika ośrodka szkolenia. W formularzu zgłoszeniowym kursant wybiera kategorie prawa jazdy i uzupełnia wszystkie niezbędne dane.

1. Instruktor musi posiadać co najmniej jedną licencję.
2. Licencja musi posiadać dokładnie jednego instruktora.
3. Instruktor może nauczać jazd wiele razy.
4. Jazda musi posiadać dokładnie jednego instruktora.
5. Kursant może doszkalać swoją jazdę wiele razy.
6. Jazda musi posiadać co najmniej jednego kursanta.
7. Kursant może zdawać egzamin wiele razy.
8. Egzamin musi być zdawany dokładnie przez jednego kursanta.
9. Kursant musi uczestniczyć w co najmniej w jednym kursie.
10. W kursie może uczestniczyć wielu kursantów.
11. Adres musi zawierać co najmniej jednego kursanta.
12. Kursant musi zawierać dokładnie jeden adres.
13. Jazda musi używać dokładnie jednego pojazdu.
14. Pojazd może być używany wiele razy.

### 3.3 Model logiczny danych



#### 3.3.1 Opis encji i atrybutów oraz ich dziedzin

##### Kursant

- **Imię** - obowiązkowe, napis długości nie więcej niż 30 znaków
- **Nazwisko** - obowiązkowe, napis długości nie więcej niż 50 znaków
- **Data urodzenia** - obowiązkowe, data poprzedzająca aktualną
- **Numer telefonu** - obowiązkowe, napis o stałej długości 9 znaków zawierający same cyfry. Wartość unikalna
- **Numer dowodu osobistego** - obowiązkowe, napis o stałej długości 9 znaków. Pierwsze 3 znaki są literami, pozostałe znaki to cyfry. Wartość unikalna

## Adres

- **Miejscowość** – obowiązkowe, napis o długości nie więcej niż 50 znaków
- **Ulica** – obowiązkowe, napis o długości nie więcej niż 50 znaków
- **Kod pocztowy** – obowiązkowe, napis o stałej długości 6 znaków. Wszystkie znaki to cyfry, oprócz znaku „ - ”, który znajduje się na trzeciej pozycji.
- **Numer domu** – opcjonalne, liczba całkowita większa niż 0
- **Numer lokalu** – opcjonalne, liczba całkowita większa niż 0

## Instruktor

- **Imię** - obowiązkowe, napis o długości nie więcej niż 30 znaków
- **Nazwisko** – obowiązkowe, napis o długości nie więcej niż 50 znaków
- **PESEL** – obowiązkowe, napis o stałej długości 11 znaków. Wszystkie znaki są cyframi. Wartość unikalna
- **Numer telefonu** - obowiązkowe, napis o długości 9 znaków zawierający same cyfry. Wartość unikalna
- **Data urodzenia** – opcjonalne, data poprzedzająca aktualną

## Licencja

- **Data wydania** – obowiązkowe, data poprzedzająca aktualną
- **Data ważności** – obowiązkowe, data późniejsza niż aktualna
- **Kategoria** – obowiązkowe, napis o długości nie więcej niż 4 znaki zawierający nazwę kategorii prawa jazdy, np. A, B itd.

## Pojazd

- **Rodzaj** – obowiązkowe, napis o długości nie więcej niż 30 znaków zawierający nazwę rodzaju pojazdu, np. samochód osobowy, motocykl itd.
- **Numer rejestracyjny** – obowiązkowe, napis o stałej długości 7 znaków. Pierwsze 3 znaki to duże litery, pozostałe mogą być cyframi lub dużymi literami. Wartość unikalna
- **Data przeglądu** – obowiązkowe, data późniejsza niż aktualna
- **Marka** – opcjonalne, napis o długości nie więcej niż 20 znaków
- **Model** – opcjonalne, napis o długości nie więcej niż 20 znaków
- **Rocznik** – opcjonalne, liczba całkowita większa od zera ale mniejsza lub równa aktualnemu rokowi

## Jazda

- **Data jazdy** – obowiązkowe, zawiera datę późniejszą niż aktualna
- **Godzina rozpoczęcia** – obowiązkowe, zawiera godzinę
- **Liczba godzin** – opcjonalne, liczba całkowita większa od zera
- **Opis** – opcjonalne, napis o długości nie więcej niż 100 znaków

## Egzamin

- **Data egzaminu** – obowiązkowe, zawiera datę późniejszą niż aktualna oraz godzinę

- **Kategoria** - obowiązkowe, napis długości nie więcej niż 4 znaki zawierający nazwę kategorii prawa jazdy, np. A, B itd.
- **Rodzaj** – obowiązkowe, napis długości 1 znaku zawierający literę T lub P
- **Zaliczony** – opcjonalne, napis długości 1 znaku zawierający literę T lub F

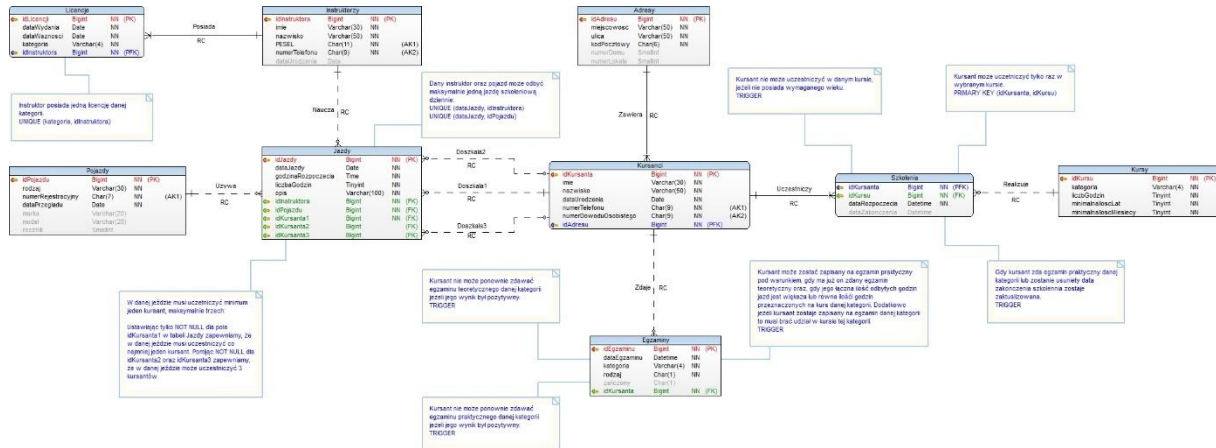
## Kurs

- **Kategoria** – obowiązkowe, napis długości nie więcej niż 4 znaki zawierający nazwę kategorii prawa jazdy, np. A, B itd.
- **Liczba godzin** – obowiązkowe, liczba całkowita większa od zera
- **Minimalna ilość lat** – obowiązkowe, liczba całkowita większa niż 0
- **Minimalna ilość miesięcy** – obowiązkowe, liczba całkowita większa lub równa 0

### 3.3.2 Weryfikacja reguł

- Fakty 1 oraz 2 zamodelowane są przez zależność „Posiada” między encjami „Licencja” i „Instruktor”:
  - ✓ Każdy instruktor musi posiadać co najmniej jedną licencję.
  - ✓ Każda licencja musi posiadać dokładnie jednego instruktora.
- Fakty 3 oraz 4 zamodelowane są przez zależność „Naucza” między encjami „Instruktor” i „Jazda”:
  - ✓ Każdy instruktor może nauczać jazd wiele razy.
  - ✓ Każda jazda musi posiadać dokładnie jednego instruktora.
- Fakty 5 oraz 6 zamodelowane są przez zależność „Doszkala” między encjami „Jazda” i „Kursant”:
  - ✓ Każdy kursant może doszkalać swoją jazdę wiele razy.
  - ✓ Każda jazda musi posiadać kursanta.
- Fakty 7 oraz 8 zamodelowane są przez zależność „Zdaje” między encjami „Kursant” i „Egzamin”:
  - ✓ Każdy kursant może zdawać egzamin wiele razy.
  - ✓ Każdy egzamin musi być zdawany przez dokładnie jednego kursanta.
- Fakty 9 oraz 10 zamodelowane są przez zależność „Uczestniczy” między encjami „Kursant” i „Kurs”:
  - ✓ Każdy kursant musi uczestniczyć w co najmniej jednym kursie.
  - ✓ Każdy kurs może być uczestniczony przez wielu kursantów.
- Fakty 11 oraz 12 zamodelowane są przez zależność „Zawiera” między encjami „Adres” i „Kursant”:
  - ✓ Każdy adres musi zawierać co najmniej jednego kursanta.
  - ✓ Każdy kursant musi zawierać dokładnie jeden adres.
- Fakty 13 oraz 14 zamodelowane są przez zależność „Używa” między encjami „Jazda” oraz „Pojazd”:
  - ✓ Każdy pojazd może być używany wiele razy przez jazdy.
  - ✓ Każda jazda musi używać dokładnie jednego pojazdu.

### 3.4 Model fizyczny danych

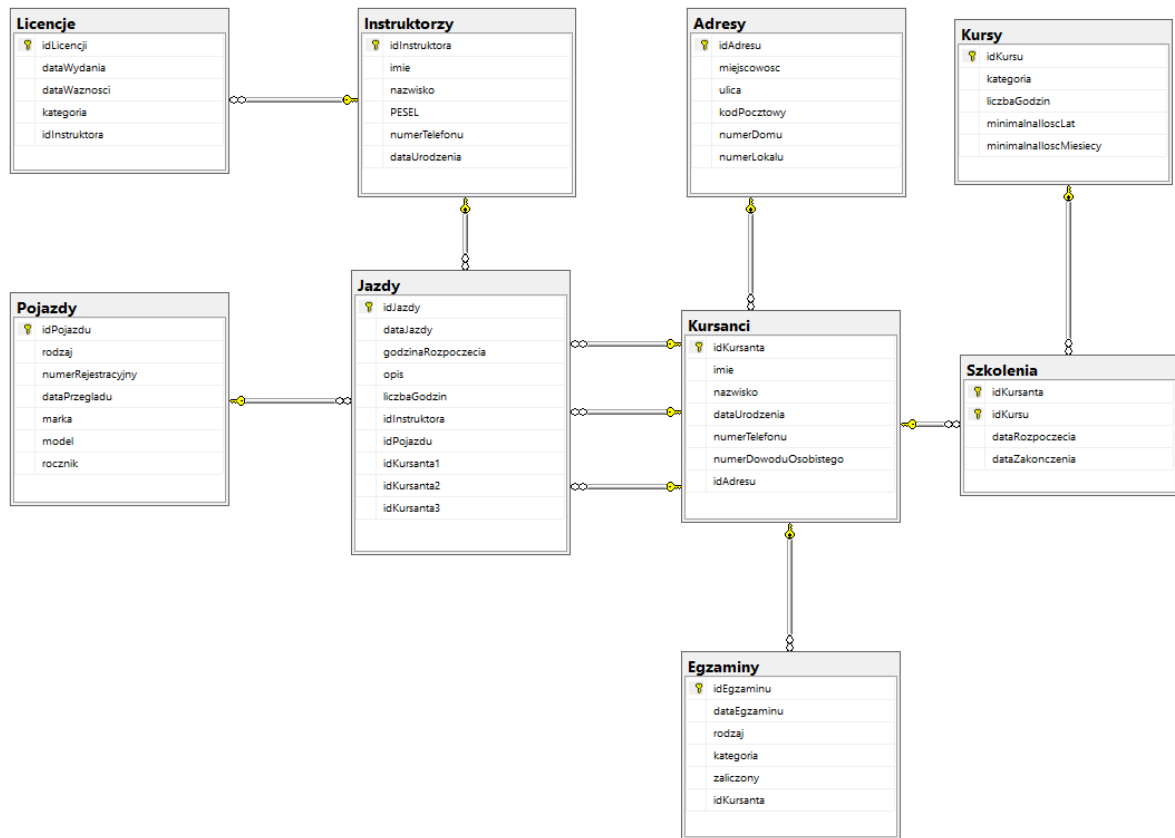


#### Uwagi:

- Wprowadzając Unique(kategoria, idInstruktora) w tabeli licencje zapewniamy instruktorowi posiadanie maksymalnie jednej licencji danej kategorii. Oznacza to, że instruktor nie może posiadać dwóch licencji danej kategorii np. B.
- W tabeli szkolenia nie ma potrzeby definiowania niezależnego klucza surogatowego idSzkolenia. Ustawiając klucz główny PRIMARY KEY(idKursanta, idKursu) zapewniamy kursantowi uczestniczenie w wybranym kursie tylko raz.
- Ustawiając w tabeli jazdy idKursanta1 na NOT NULL oraz pomijając NOT NULL dla idKursanta2 oraz idKursanta3 zapewniamy, że w danej jeździe może uczestniczyć maksymalnie trzech kursantów ale musi minimum jeden.
- Wprowadzając Unique(dataJazdy, idInstruktora) w tabeli jazdy zapewniamy instruktorowi odbycie maksymalnie jednej jazdy dziennie.
- Wprowadzając Unique(dataJazdy, idPojazdu) w tabeli jazdy zapewniamy użycie danego pojazdu maksymalnie raz dziennie.
- Wprowadzając CheckConstraint dla tabeli jazdy o treści: idKursanta1 != idKursanta2 AND idKursanta1 != idKursanta3 AND (idKursanta2 != idKursanta3 OR idKursanta2 IS NULL OR idKursanta3 IS NULL). Zapewniamy brak możliwości wprowadzenia identycznych kursantów podczas jazdy z instruktorem w danym dniu.

## 4 Realizacja bazy danych

### 4.1 Diagram ER realizacji bazy w MS SQL Server



### 4.2 Kod SQL tworzący bazę

-- Create tables section

-- Table dbo.Instruktorzy

```
CREATE TABLE [dbo].[Instruktorzy] (  
    [idInstruktora] INT IDENTITY(1, 1) PRIMARY KEY,  
    [imie] VARCHAR(30) NOT NULL,  
    [nazwisko] VARCHAR(50) NOT NULL,  
    [PESEL] CHAR(11) NOT NULL UNIQUE,  
    [numerTelefonu] CHAR(9) NOT NULL UNIQUE,  
    [dataUrodzenia] DATE,  
  
    CONSTRAINT [chkPeselInstruktora] CHECK(PESEL Like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
    CONSTRAINT [chkNumerTelefonuInstruktora] CHECK(numerTelefonu LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
    CONSTRAINT [chkDataUrodzeniaInstruktora] CHECK(dataUrodzenia < GETDATE())  
)  
go
```

-- Table dbo.Adresy

```
CREATE TABLE [Adresy] (  
    [idAdresu] INT IDENTITY(1, 1) PRIMARY KEY,  
    [miejscowosc] VARCHAR(50) NOT NULL,  
    [ulica] VARCHAR(50) NOT NULL,  
    [kodPocztowy] CHAR(6) NOT NULL,  
    [numerDomu] SMALLINT,  
    [numerLokalu] SMALLINT,
```

```

        CONSTRAINT [chkNumerZamieszkaniaAdresu] CHECK((numerDomu > 0 OR numerDomu IS
NULL) AND (numerLokalu > 0 OR numerLokalu IS NULL)),
        CONSTRAINT [chkKodPocztowyAdresu] CHECK(kodPocztowy LIKE '[0-9][0-9]-[0-9][0-
9][0-9]')
    )
go

-- Table dbo.Kursanci
CREATE TABLE [Kursanci] (
    [idKursanta] INT IDENTITY(1, 1) PRIMARY KEY,
    [imie] VARCHAR(30) NOT NULL,
    [nazwisko] VARCHAR(50) NOT NULL,
    [dataUrodzenia] DATE NOT NULL,
    [numerTelefonu] CHAR(9) NOT NULL UNIQUE,
    [numerDowoduOsobistego] CHAR(9) NOT NULL UNIQUE,
    [idAdresu] INT NOT NULL,

    CONSTRAINT [chkNumerTelefonuKursanta] CHECK(numerTelefonu LIKE '[0-9][0-9][0-
9][0-9][0-9][0-9][0-9][0-9]'),
    CONSTRAINT [chkNumerDowoduOsobistegoKursanta] CHECK(numerDowoduOsobistego LIKE
'[A-Z][A-Z][A-Z][0-9][0-9][0-9][0-9][0-9]'),
    CONSTRAINT [chDataUrodzeniaKursanta] CHECK(dataUrodzenia < GETDATE())
)
go

-- Table dbo.Egzaminy
CREATE TABLE [Egzaminy] (
    [idEgzaminu] INT IDENTITY(1, 1) PRIMARY KEY,
    [dataEgzaminu] DATETIME NOT NULL,
    [rodzaj] CHAR(1) NOT NULL,
    [kategoria] VARCHAR(4) NOT NULL,
    [zaliczony] CHAR(1),
    [idKursanta] INT NOT NULL,

    CONSTRAINT [chkDataEgzaminu] CHECK(dataEgzaminu > GETDATE()),
    CONSTRAINT [chkRodzajEgzaminu] CHECK(rodzaj IN ('T', 'P')),
    CONSTRAINT [chkZaliczenieEgzaminu] CHECK(zaliczony IS NULL OR zaliczony IN ('T',
'F')),
    CONSTRAINT [chkKategoriaEgzaminu] CHECK(kategoria IN('AM', 'A1', 'A2', 'A',
'B1', 'B', 'B+E', 'C', 'C1', 'C1+E', 'D', 'D1', 'D1+E', 'D+E', 'T'))
)
go

-- Table dbo.Kursy
CREATE TABLE [dbo].[Kursy] (
    [idKursu] INT IDENTITY(1, 1) PRIMARY KEY,
    [kategoria] VARCHAR(4) NOT NULL,
    [liczbaGodzin] TINYINT NOT NULL,
    [minimalnaIloscLat] TINYINT NOT NULL,
    [minimalnaIloscMiesiecy] TINYINT NOT NULL,

    CONSTRAINT [chkKategoriaKursu] CHECK(kategoria IN('AM', 'A1', 'A2', 'A', 'B1',
'B', 'B+E', 'C', 'C1', 'C1+E', 'D', 'D1', 'D1+E', 'D+E', 'T')),
    CONSTRAINT [chkMinimalnaIloscLatKursu] CHECK(minimalnaIloscLat > 0),
    CONSTRAINT [chkLiczbaGodzinKursu] CHECK(liczbaGodzin > 0)
)
go

-- Table dbo.Pojazdy
CREATE TABLE [dbo].[Pojazdy] (
    [idPojazdu] INT IDENTITY(1, 1) PRIMARY KEY,

```

```

[rodzaj] VARCHAR(30) NOT NULL,
[numerRejestracyjny] CHAR(7) UNIQUE NOT NULL,
[dataPrzeglądu] DATE NOT NULL,
[marka] VARCHAR(20),
[model] VARCHAR(20),
[rocznik] SMALLINT,

CONSTRAINT [chkRodzajPojazdu] CHECK(rodzaj IN ('samochód osobowy', 'autobus',
'motocykl', 'samochód ciężarowy')),
CONSTRAINT [chkNumerRejestracyjnyPojazdu] CHECK(numerRejestracyjny LIKE '[A-
Z][A-Z][A-Z][A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9]'),
CONSTRAINT [chkDataPrzegląduPojazdu] CHECK(dataPrzeglądu > GETDATE()),
CONSTRAINT [chkRocznikPojazdu] CHECK(rocznik IS NULL OR rocznik > 0)
)
go

-- Table dbo.Jazdy
CREATE TABLE [dbo].[Jazdy] (
[idJazdy] INT IDENTITY(1, 1) PRIMARY KEY,
[dataJazdy] DATE NOT NULL,
[godzinaRozpoczecia] TIME NOT NULL,
[opis] VARCHAR(100) NOT NULL,
[liczbaGodzin] TINYINT NOT NULL,
[idInstruktora] INT NOT NULL,
[idPojazdu] INT NOT NULL,
[idKursanta1] INT NOT NULL,
[idKursanta2] INT,
[idKursanta3] INT,

CONSTRAINT [unqJazdaInstruktora] UNIQUE(dataJazdy, idInstruktora),
CONSTRAINT [unqJazdaPojazdu] UNIQUE(dataJazdy, idPojazdu),
CONSTRAINT [chkDataJazdy] CHECK(dataJazdy >= GETDATE()),
CONSTRAINT [chkLiczbaGodzinJazdy] CHECK(liczbaGodzin > 0 OR liczbaGodzin IS
NULL),
CONSTRAINT [chkIdKursantaJazdy] CHECK(idKursanta1 != idKursanta2 AND idKursanta1
!= idKursanta3 AND (idKursanta2 != idKursanta3 OR idKursanta2 IS NULL OR idKursanta3
IS NULL))
)
go

-- Table dbo.Szkolenia
CREATE TABLE [dbo].[Szkolenia] (
[idKursanta] INT NOT NULL,
[idKursu] INT NOT NULL,
[dataRozpoczecia] DATE DEFAULT CONVERT(date, GETDATE()) NOT NULL,
[dataZakonczenia] DATE,

CONSTRAINT [PK_Szkolenia] PRIMARY KEY(idKursanta, idKursu),
CONSTRAINT [chkDataRozpoczeciaSzkolenia] CHECK(dataRozpoczecia >= CONVERT(date,
GETDATE())),
CONSTRAINT [chkDataZakonczeniaSzkolenia] CHECK(dataZakonczenia >=
dataRozpoczecia)
)
go

-- Table dbo.Licencje
CREATE TABLE [dbo].[Licencje] (
[idLicencji] INT IDENTITY(1,1) PRIMARY KEY,
[dataWydania] DATE NOT NULL,
[dataWaznosci] DATE NOT NULL,
[kategoria] VARCHAR(4) NOT NULL,
[idInstruktora] INT NOT NULL,

```



```

        CONSTRAINT [chkDatyLicencji] CHECK(dataWydania <= GETDATE() AND dataWydania <
dataWaznosci AND dataWaznosci > GETDATE()),
        CONSTRAINT [chkKategoriaLicencji] CHECK(kategoria IN('AM', 'A1', 'A2', 'A',
'B1', 'B', 'B+E', 'C', 'C1', 'C1+E', 'D', 'D1', 'D1+E', 'D+E', 'T')),
        CONSTRAINT [unqLicencjaInstruktora] UNIQUE(kategoria, idInstruktora)
    )
go

-- Create foreign keys (relationships) section

ALTER TABLE [dbo].[Egzaminy] ADD CONSTRAINT [Zdaje] FOREIGN KEY [(idKursanta)]
REFERENCES
[dbo].[Kursanci] [(idKursanta)] ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [dbo].[Kursanci] ADD CONSTRAINT [Zawiera] FOREIGN KEY [(idAdresu)]
REFERENCES
[dbo].[Adresy] [(idAdresu)] ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [dbo].[Licencje] ADD CONSTRAINT [Posiada] FOREIGN KEY [(idInstruktora)]
REFERENCES
[dbo].[Instruktorzy] [(idInstruktora)] ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [dbo].[Szkolenia] ADD CONSTRAINT [Realizuje] FOREIGN KEY [(idKursanta)]
REFERENCES
[dbo].[Kursanci] [(idKursanta)] ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [dbo].[Szkolenia] ADD CONSTRAINT [Uczestniczy] FOREIGN KEY [(idKursu)]
REFERENCES
[dbo].[Kursy] [(idKursu)] ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [dbo].[Jazdy] ADD CONSTRAINT [Doszkala1] FOREIGN KEY [(idKursanta1)]
REFERENCES
[dbo].[Kursanci] [(idKursanta)] ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [dbo].[Jazdy] ADD CONSTRAINT [Doszkala2] FOREIGN KEY [(idKursanta2)]
REFERENCES
[dbo].[Kursanci] [(idKursanta)] ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [dbo].[Jazdy] ADD CONSTRAINT [Doszkala3] FOREIGN KEY [(idKursanta3)]
REFERENCES
[dbo].[Kursanci] [(idKursanta)] ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [dbo].[Jazdy] ADD CONSTRAINT [Naucza] FOREIGN KEY [(idInstruktora)]
REFERENCES
[dbo].[Instruktorzy] [(idInstruktora)] ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [dbo].[Jazdy] ADD CONSTRAINT [Uzywa] FOREIGN KEY [(idPojazdu)] REFERENCES
[dbo].[Pojazdy] [(idPojazdu)] ON UPDATE NO ACTION ON DELETE NO ACTION
go

```

#### 4.2.1 Kod SQL wypełniający bazę danymi testowymi

----- Tabela Adresy -----

-- Próba wprowadzenia błędnego kodu pocztowego

```
INSERT INTO Adresy(miejscowosc, ulica, kodPocztowy) VALUES('Kraków', 'Słowicza', '31A320');
```

-- Próba wprowadzenia wartości ujemnej dla numer domu

```
INSERT INTO Adresy(miejscowosc, ulica, kodPocztowy, numerDomu) VALUES('Kraków', 'Mickiewicza', '31-320', -1);
```

-- Próba wprowadzenia wartości ujemnej dla numer lokalu

```
INSERT INTO Adresy(miejscowosc, ulica, kodPocztowy, numerLokalu) VALUES('Myślenice', 'Curie', '32-230', -1);
```

----- Tabela Egzaminy -----

-- Próba wprowadzenia innej litery niż T, P dla rodzaju egzaminu

```
INSERT INTO Egzaminy(dataEgzaminu, rodzaj, kategoria, idKursanta) VALUES (GETDATE()+1, 'X', 'B', 3);
```

-- Próba wprowadzenia innej litery niż T, F dla zaliczenia egzaminu

```
INSERT INTO Egzaminy(dataEgzaminu, rodzaj, kategoria, zaliczony, idKursanta) VALUES (GETDATE()+1, 'T', 'B', 'X', 4);
```

-- Próba wprowadzenia daty egzaminu, która jest wcześniejsza niż aktualna

```
INSERT INTO Egzaminy(dataEgzaminu, rodzaj, kategoria, idKursanta) VALUES (GETDATE()-1, 'T', 'B', 5);
```

-- Próba wprowadzenia innej kategorii prawa jazdy niż AM, A1, A2, A, B1, B, B+E, C, C1, C1+E, D, D1, D1+E, D+E, T

```
INSERT INTO Egzaminy(dataEgzaminu, rodzaj, kategoria, idKursanta) VALUES (GETDATE()+1, 'T', 'X', 6);
```

----- Tabela Instruktorzy -----

-- Próba wprowadzenia numeru PESEL, który zawiera litery zamiast cyfr

```
INSERT INTO Instruktorzy(imie, nazwisko, PESEL, numerTelefonu) VALUES('Szczepan', 'Mikołajczyk', '42A10500891', '883924234');
```

-- Próba wprowadzenia numeru telefonu, który zawiera litery zamiast cyfr

```
INSERT INTO Instruktorzy(imie, nazwisko, PESEL, numerTelefonu) VALUES('Szczepan', 'Mikołajczyk', '42310500891', '88A924234');
```

-- Próba wprowadzenia daty urodzenia, która jest późniejsza niż aktualna

```
INSERT INTO Instruktorzy(imie, nazwisko, PESEL, numerTelefonu, dataUrodzenia) VALUES('Szczepan', 'Mikołajczyk', '42310500891', '883924234', GETDATE()+1);
```

-- Próba wprowadzenia zduplikowanego numeru telefonu

```
INSERT INTO Instruktorzy(imie, nazwisko, PESEL, numerTelefonu) VALUES('Szczepan', 'Mikołajczyk', '42310500891', '609136229');
```

-- Próba wprowadzenia zduplikowanego numeru PESEL

```
INSERT INTO Instruktorzy(imie, nazwisko, PESEL, numerTelefonu) VALUES('Szczepan', 'Mikołajczyk', '12282706154', '883924234');
```

----- Tabela Jazdy -----

-- Próba wprowadzenia instruktora, który odbył już jazdę tego samego dnia

```
INSERT INTO Jazdy (dataJazdy, godzinaRozpoczenia, opis, liczbaGodzin, idInstruktora, idPojazdu, idKursanta1, idKursanta2, idKursanta3) VALUES(GETDATE()+32, '14:45', 'Jazda po mieście', 2, 3, 4, 2, null, null);
```

```

-- Próba wprowadzenia pojazdu, który był już używany tego samego dnia
INSERT INTO Jazdy (dataJazdy, godzinaRozpoczecia, opis, liczbaGodzin, idInstruktora,
idPojazdu, idKursanta1, idKursanta2, idKursanta3)
VALUES(GETDATE()+32, '14:45', 'Jazda po mieście', 2, 1, 3, 3, null, null);

-- Próba wprowadzenia daty jazdy, która jest wcześniejsza niż aktualna
INSERT INTO Jazdy (dataJazdy, godzinaRozpoczecia, opis, liczbaGodzin, idInstruktora,
idPojazdu, idKursanta1, idKursanta2, idKursanta3)
VALUES(GETDATE()-1, '14:45', 'Jazda po mieście', 0, 1, 3, 1, 2, 3);

-- Próba wprowadzenia liczby godzin równej 0
INSERT INTO Jazdy (dataJazdy, godzinaRozpoczecia, opis, liczbaGodzin, idInstruktora,
idPojazdu, idKursanta1, idKursanta2, idKursanta3)
VALUES(GETDATE()+33, '14:45', 'Jazda po mieście', 0, 1, 3, 1, 2, 3);

-- Próba wprowadzenia jazdy bez kursantów
INSERT INTO Jazdy (dataJazdy, godzinaRozpoczecia, opis, liczbaGodzin, idInstruktora,
idPojazdu, idKursanta1, idKursanta2, idKursanta3)
VALUES(GETDATE()+33, '14:45', 'Jazda po mieście', 3, 1, 3, null, null, null);

-- Próba wprowadzenia tych samych kursantów
INSERT INTO Jazdy (dataJazdy, godzinaRozpoczecia, opis, liczbaGodzin, idInstruktora,
idPojazdu, idKursanta1, idKursanta2, idKursanta3)
VALUES(GETDATE()+33, '14:45', 'Jazda po mieście', 3, 1, 3, 1, 1, null);

INSERT INTO Jazdy (dataJazdy, godzinaRozpoczecia, opis, liczbaGodzin, idInstruktora,
idPojazdu, idKursanta1, idKursanta2, idKursanta3)
VALUES(GETDATE()+33, '14:45', 'Jazda po mieście', 3, 1, 3, 1, 2, 1);

INSERT INTO Jazdy (dataJazdy, godzinaRozpoczecia, opis, liczbaGodzin, idInstruktora,
idPojazdu, idKursanta1, idKursanta2, idKursanta3)
VALUES(GETDATE()+33, '14:45', 'Jazda po mieście', 3, 1, 3, 1, 2, 2);

----- Tabela Kursanci -----

-- Próba wprowadzenia numeru dowodu osobistego, który zawiera w pierwszych trzech
znakach liczby
INSERT INTO Kursanci VALUES('Zofia', 'Bednarczyk', '1974-04-22', '933861776',
'A1J169663', 1);

-- Próba wprowadzenia numeru dowodu osobistego, który zawiera od czwartego znaku
litery
INSERT INTO Kursanci VALUES('Zofia', 'Bednarczyk', '1974-04-22', '933861776',
'AMJA69663', 1);

-- Próba wprowadzenia numeru telefonu, który zawiera litery zamiast cyfr
INSERT INTO Kursanci VALUES('Zofia', 'Bednarczyk', '1974-04-22', '93A861776',
'AMJ169663', 1);

-- Próba wprowadzenia daty urodzenia, która jest późniejsza niż aktualna
INSERT INTO Kursanci VALUES('Zofia', 'Bednarczyk', GETDATE()+1, '933861776',
'AMJ169663', 1);

-- Próba wprowadzenia duplikatu dla numeru telefonu
INSERT INTO Kursanci VALUES('Zofia', 'Bednarczyk', '1974-04-22', '618205993',
'AMJ310155', 1);

-- Próba wprowadzenia duplikatu dla numeru dowodu osobistego
INSERT INTO Kursanci VALUES('Zofia', 'Bednarczyk', '1974-04-22', '933861776',
'AMJ355844', 1);

```

----- Tabela Kursy -----

-- Próba wprowadzenia kategorii która jest inna niż AM, A1, A2, A, B1, B, B+E, C, C1, C1+E, D, D1, D1+E, D+E, T

INSERT INTO Kursy VALUES('X', 30, 17, 9);

-- Próba wprowadzenie liczby godzin równej 0

INSERT INTO Kursy VALUES('B1', 0, 17, 9);

-- Próba wprowadzenia minimalnej ilość lat równej 0

INSERT INTO Kursy VALUES('B1', 30, 0, 9);

----- Tabela Licencje -----

-- Proba wprowadzenia licencji, którą instruktor już posiada

INSERT INTO Licencje VALUES(GETDATE()-365, GETDATE()+365, 'AM', 1);

-- Próba wprowadzenia daty wydania, która jest późniejsza niż aktualna

INSERT INTO Licencje VALUES(GETDATE()+5, GETDATE()+365, 'AM', 5);

-- Próba wprowadzenia daty ważności, która jest wcześniejsza niż aktualna

INSERT INTO Licencje VALUES(GETDATE()-365, GETDATE()-5, 'AM', 5);

-- Próba wprowadzenia kategorii która nie jest AM, A1, A2, A, B1, B, B+E, C, C1, C1+E, D, D1, D1+E, D+E, T

INSERT INTO Licencje VALUES(GETDATE()-365, GETDATE()+365, 'X', 5);

----- Tabela Pojazdy -----

-- Próba wprowadzenia numeru rejestracyjnego, który nie posiada na początku 3 dużych liter

INSERT INTO Pojazdy VALUES('samochód osobowy', '0RK7321', GETDATE()+365, 'Toyota', 'Yaris', 2010);

-- Próba wprowadzenia innego rodzaju pojazdu niż samochód osobowy, autobus, motocykl, samochód ciężarowy

INSERT INTO Pojazdy VALUES('X', 'KBR9056', GETDATE()+365, 'Toyota', 'Yaris', 2010);

-- Próba wprowadzenia daty przeglądu, która jest późniejsza niż aktualna

INSERT INTO Pojazdy VALUES('samochód osobowy', 'KBR9056', GETDATE()-1, 'Toyota', 'Yaris', 2010);

-- Próba wprowadzenia rocznika który jest liczbą ujemną

INSERT INTO Pojazdy VALUES('samochód osobowy', 'KBR9056', GETDATE()+365, 'Toyota', 'Yaris', -1);

----- Tabela Szkolenia -----

-- Próba wprowadzenia kursanta, który ma już rozpoczęty ten sam kurs danej kategorii

INSERT INTO Szkolenia(idKursanta, idKursu) VALUES(1, 1);

-- Próba wprowadzenia daty rozpoczęcia szkolenia, która jest wcześniejsza niż aktualna

INSERT INTO Szkolenia(idKursanta, idKursu, dataRozpoczecia) VALUES(9, 3, GETDATE()-1);

-- Próba wprowadzenia daty zakończenia szkolenia, która jest wcześniejsza niż data rozpoczęcia

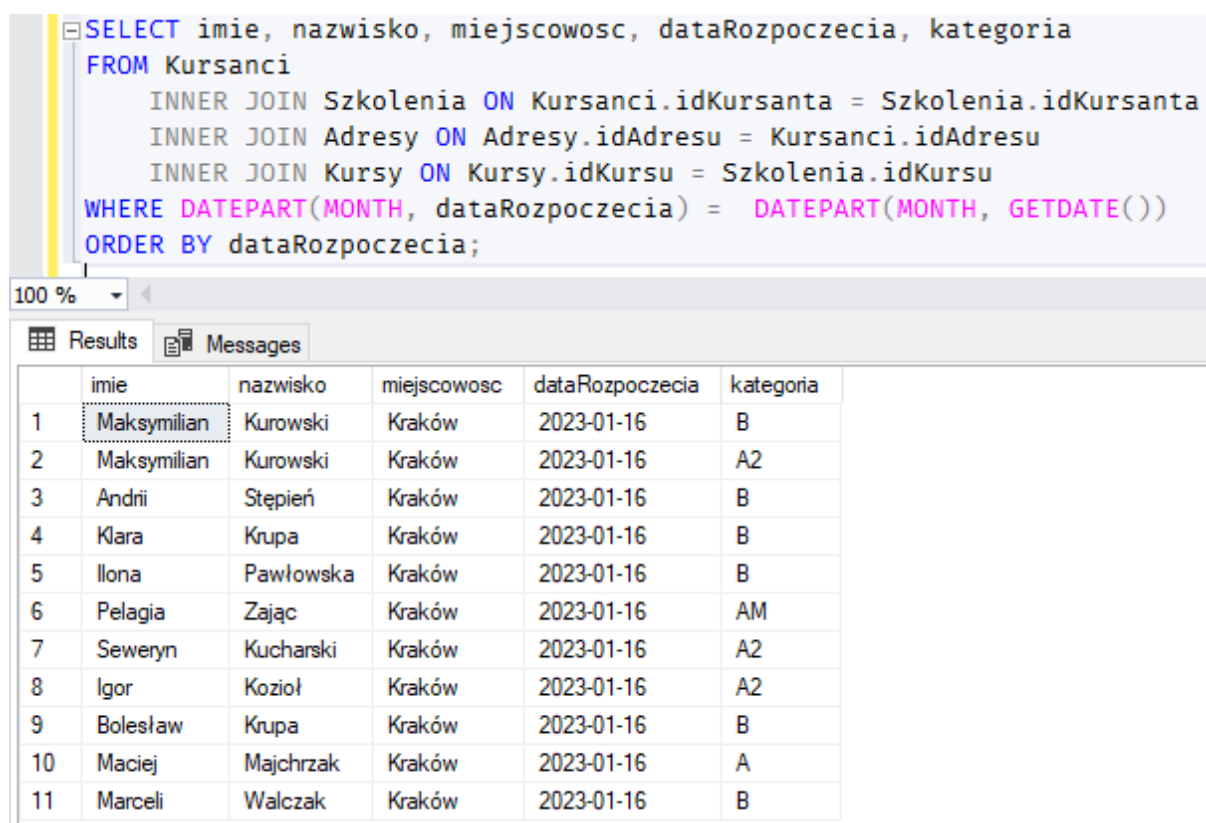
INSERT INTO Szkolenia(idKursanta, idKursu, dataRozpoczecia, dataZakonczenia) VALUES(9, 3, GETDATE()+2, GETDATE()+1);

## 5 Przykładowe zapytania SQL do bazy danych

### 5.1 Kwerenda – Kursanci zapisani na szkolenie w bieżącym miesiącu

Imiona, nazwiska, miejsca zamieszkania, daty zapisów oraz kategorie kursantów zapisanych na szkolenie w bieżącym. Dane posortowane według daty zapisu.

```
SELECT imie, nazwisko, miejscowosc, dataRozpoczenia, kategoria
FROM Kursanci
    INNER JOIN Szkolenia ON Kursanci.idKursanta = Szkolenia.idKursanta
    INNER JOIN Adresy ON Adresy.idAdresu = Kursanci.idAdresu
    INNER JOIN Kursy ON Kursy.idKursu = Szkolenia.idKursu
WHERE DATEPART(MONTH, dataRozpoczenia) = DATEPART(MONTH, GETDATE())
ORDER BY dataRozpoczenia;
```



The screenshot shows a SQL query editor with a query window and a results window. The query is the same as the one above. The results window shows a table with 6 columns: imie, nazwisko, miejscowosc, dataRozpoczenia, and kategoria. The table contains 11 rows of data, sorted by dataRozpoczenia.

	imie	nazwisko	miejscowosc	dataRozpoczenia	kategoria
1	Maksymilian	Kurowski	Kraków	2023-01-16	B
2	Maksymilian	Kurowski	Kraków	2023-01-16	A2
3	Andrii	Stępień	Kraków	2023-01-16	B
4	Klara	Krupa	Kraków	2023-01-16	B
5	Ilona	Pawłowska	Kraków	2023-01-16	B
6	Pelagia	Zajac	Kraków	2023-01-16	AM
7	Seweryn	Kucharski	Kraków	2023-01-16	A2
8	Igor	Kozioł	Kraków	2023-01-16	A2
9	Bolesław	Krupa	Kraków	2023-01-16	B
10	Maciej	Majchrzak	Kraków	2023-01-16	A
11	Marceli	Walczak	Kraków	2023-01-16	B

### 5.2 Kwerenda – Kursanci biorący udział w kursie kategorii B mieszkający w Krakowie przy ulicy Czarnolaskiej

Kursanci zapisani na kurs prawa jazdy kategorii B, którzy mieszkają w Krakowie przy ulicy Czarnolaskiej, wraz z id zapisanym jako numerKursanta, imieniem, nazwiskiem, miejscowością, ulicą, kodem pocztowym oraz kategorią. Dodatkowo dane posortowane według daty urodzenia.

```
SELECT Kursanci.idKursanta AS numerKursanta, imie, nazwisko, miejscowosc, ulica,
kodPocztowy, kategoria
FROM Kursanci
    LEFT JOIN Adresy ON Kursanci.idAdresu = Adresy.idAdresu
    INNER JOIN Szkolenia ON Szkolenia.idKursanta = Kursanci.idKursanta
    INNER JOIN Kursy ON Szkolenia.idKursu = Kursy.idKursu
WHERE kategoria = 'B' AND miejscowosc = 'Kraków' AND ulica = 'Czarnolaska'
ORDER BY dataUrodzenia;
```

```

SELECT Kursanci.idKursanta AS numerKursanta, imie, nazwisko, miejscowosc, ulica, kodPocztowy, kategoria
FROM Kursanci
LEFT JOIN Adresy ON Kursanci.idAdresu = Adresy.idAdresu
INNER JOIN Szkolenia ON Szkolenia.idKursanta = Kursanci.idKursanta
INNER JOIN Kursy ON Szkolenia.idKursu = Kursy.idKursu
WHERE kategoria = 'B' AND miejscowosc = 'Kraków' AND ulica = 'Czarnolaska'
ORDER BY dataUrodzenia;

```

	numerKursanta	imie	nazwisko	miejscowosc	ulica	kodPocztowy	kategoria
1	4	Ilona	Pawłowska	Kraków	Czarnolaska	31-619	B
2	3	Klara	Krupa	Kraków	Czarnolaska	31-619	B
3	8	Bolesław	Krupa	Kraków	Czarnolaska	31-619	B

### 5.3 Kwerenda – Instruktorzy oraz liczba posiadanych licencji

Imiona, nazwiska oraz numer telefonów instruktorów wraz z liczbą posiadanych licencji. Dane posortowane malejąco według liczby posiadanych licencji.

```

SELECT imie, nazwisko, numerTelefonu, COUNT(*) AS liczbaLicencji
FROM Instruktorzy
INNER JOIN Licencje ON Instruktorzy.idInstruktora = Licencje.idInstruktora
GROUP BY imie, nazwisko, numerTelefonu
ORDER BY liczbaLicencji DESC;

```

```

SELECT imie, nazwisko, numerTelefonu, COUNT(*) AS liczbaLicencji
FROM Instruktorzy
INNER JOIN Licencje ON Instruktorzy.idInstruktora = Licencje.idInstruktora
GROUP BY imie, nazwisko, numerTelefonu
ORDER BY liczbaLicencji DESC;

```

	imie	nazwisko	numerTelefonu	liczbaLicencji
1	Cezary	Kołodziejczyk	609136229	4
2	Włodzimierz	Czamecki	520609335	3
3	Damian	Wasilewski	600818714	1
4	Artur	Barański	402359780	1
5	Małola	Szymańska	430366277	1

### 5.4 Kwerenda – Kursanci oraz ich liczba podejść do egzaminu praktycznego prawa jazdy

Imiona, nazwiska oraz daty urodzenia kursantów wraz z kategorią oraz liczbą podejść do egzaminu praktycznego. Dane posortowane według daty urodzenia oraz liczby podejść do egzaminu praktycznego.

```

SELECT imie, nazwisko, dataUrodzenia, kategoria, COUNT(*) AS liczbaPodejsc
FROM Kursanci
INNER JOIN Egzaminy ON Kursanci.idKursanta = Egzaminy.idKursanta
WHERE rodzaj = 'P'
GROUP BY Kursanci.idKursanta, imie, nazwisko, dataUrodzenia, kategoria
ORDER BY dataUrodzenia, liczbaPodejsc;

```

```

SELECT imie, nazwisko, dataUrodzenia, kategoria, COUNT(*) AS liczbaPodejsc
FROM Kursanci
    INNER JOIN Egzaminy ON Kursanci.idKursanta = Egzaminy.idKursanta
WHERE rodzaj = 'P'
GROUP BY Kursanci.idKursanta, imie, nazwisko, dataUrodzenia, kategoria
ORDER BY dataUrodzenia, liczbaPodejsc;

```

	imie	nazwisko	dataUrodzenia	kategoria	liczbaPodejsc
1	Marceli	Walczak	2001-07-07	B	1
2	Maksymilian	Kurowski	2002-11-16	B	1
3	Andrii	Stępień	2003-12-29	B	4

### 5.5 Kwerenda – Instruktorzy, którzy posiadają licencję kategorii D+E

Imiona, nazwiska oraz numery telefonów instruktorów, którzy posiadają licencję kategorii D+E wraz z datą wydania licencji. Dodatkowo dane posortowane według daty wydania licencji.

```

SELECT imie, nazwisko, numerTelefonu, dataWydania
FROM Instruktorzy, Licencje
WHERE Licencje.idInstruktora = Instruktorzy.idInstruktora AND kategoria = 'D+E'
ORDER BY dataWydania;

```

	imie	nazwisko	numerTelefonu	dataWydania
1	Włodzimierz	Czamecki	520609335	2022-07-18
2	Cezary	Kołodziejczyk	609136229	2022-07-20

### 5.6 Kwerenda – Pojazdy, których przegląd techniczny kończy się w marcu bieżącego roku

Marki, modele, numery rejestracyjne oraz daty przeglądów pojazdów, których przegląd techniczny kończy się w marcu bieżącego roku. Dane posortowane według daty przeglądu.

```

SELECT marka, model, numerRejestracyjny, dataPrzeglądu
FROM Pojazdy
WHERE DATEPART(MONTH, dataPrzeglądu) = 3
ORDER BY dataPrzeglądu;

```

```
SELECT marka, model, numerRejestracyjny, dataPrzeglądu
FROM Pojazdy
WHERE DATEPART(MONTH, dataPrzeglądu) = 3
ORDER BY dataPrzeglądu;
```

	marka	model	numerRejestracyjny	dataPrzeglądu
1	Toyota	Yaris	KRK2342	2023-03-21
2	Hyundai	i20	KRK5243	2023-03-22
3	Hyundai	i20	KRK3242	2023-03-23

## 5.7 Kwerenda – Instruktorzy wraz z liczbą jazd w bieżącym miesiącu

Imiona, nazwiska oraz numery telefonów instruktorów wraz z liczbą przeprowadzonych jazd w bieżącym miesiącu oraz zsumowaną liczbą godzin jazd. Dane posortowane malejąco według liczby przeprowadzonych jazd w bieżącym miesiącu.

```
SELECT imie, nazwisko, numerTelefonu, COUNT(*) AS liczbaJazd, SUM(liczbaGodzin) AS
sumaGodzinJazd
FROM Instruktorzy
INNER JOIN Jazdy ON Instruktorzy.idInstruktora = Jazdy.idInstruktora
WHERE DATEPART(MONTH, dataJazdy) = DATEPART(MONTH, GETDATE())
GROUP BY imie, nazwisko, numerTelefonu
ORDER BY liczbaJazd DESC;
```

```
SELECT imie, nazwisko, numerTelefonu, COUNT(*) AS liczbaJazd, SUM(liczbaGodzin) AS sumaGodzinJazd
FROM Instruktorzy
INNER JOIN Jazdy ON Instruktorzy.idInstruktora = Jazdy.idInstruktora
WHERE DATEPART(MONTH, dataJazdy) = DATEPART(MONTH, GETDATE())
GROUP BY imie, nazwisko, numerTelefonu
ORDER BY liczbaJazd DESC;
```

	imie	nazwisko	numerTelefonu	liczbaJazd	sumaGodzinJazd
1	Cezary	Kołodziejczyk	609136229	5	14
2	Włodzimierz	Czamecki	520609335	2	6
3	Mariola	Szymańska	430366277	2	6
4	Damian	Wasilewski	600818714	1	2

## 6 Załączniki

- Dokumentacja projektu (plik: dokumentacja.pdf)
- Kod SQL tworzący bazę (plik: sql/baza.sql)
- Kod SQL z przykładowymi danymi (plik: sql/przykladowe\_dane.sql)
- Kod SQL testujący bazę (plik: sql/testy.sql)
- Kod SQL wyzwalacza/triggera (plik: sql/trigger.sql)
- Zdjęcia kwerend (plik: zdjecia/kwerendy)
- Zdjęcie modelu logicznego (plik: zdjecia/model\_logiczny.jpeg)
- Zdjęcie modelu fizycznego (plik: zdjecia/model\_fizyczny.jpeg)
- Zdjęcie diagramu ER w bazie danych MS SQL Server (plik: zdjecia/realizacja\_mssql.png)