

PROJEKT BAZY DANYCH SZKOŁY – ARTUR GÓRAK

Cel projektu:

Stworzenie bazy danych umożliwiającej zarządzanie planem zajęć uczniów oraz nauczycieli, klasami, ocenami oraz obecnościami przy pomocy triggerów, funkcji, widoków i procedur.

Możliwości projektu:

- blokujemy możliwość przenosin ucznia do innej klasy jeśli dany uczeń nie spełnia progu przyjęcia do danej klasy oraz jeśli dana klasa nie może przyjąć większej ilości uczniów
- wyliczanie średniej ucznia z danego przedmiotu jak i całego semestru
- wyliczenie średniej zadanej klasy
- wypisanie z imienia i nazwiska prymusów z danej klasy (czyli takich osób których średnia ≥ 4.75)
- wypisanie listy uczniów z najlepszą frekwencją z danej klasy
- wyliczenie ile nieobecności przekłada się na każdy przedmiot i nauczyciela
- wypisanie ile procent z wszystkich nieobecności przekłada się na każdą klasę
- wyliczenie ile procentowo nieobecności w danej klasie jest nieusprawiedliwionych
- wyliczenie średniej ilości nieobecności na ucznia
- wypisanie możliwych zastępców dla nauczyciela, którzy mogą danego dnia i o danej godzinie lekcyjnej go zastąpić (czyli potrafią prowadzić przedmiot nauczyciela którego zastępują oraz nie mają wtedy innych lekcji)
- wypisanie listy pracowni które są zajęte w danym przedziale czasowym
- wypisanie planu zajęć nauczyciela
- wypisanie listy uczniów którzy mają byli na podium lub laureatami w różnego rodzaju konkursach i zawodach o stopniu krajowym
- wypisanie listy wychowawców
- wypisanie listy nauczycieli oraz ile tygodniowo prowadzą oni zajęć
- wypisanie listy najczęściej wstawianego typu ocen
- blokowanie wpisania oceny spoza przedziału 1-6 oraz datą innego dnia niż się odbywają zajęcia z danego przedmiotu
- blokowanie wpisywania nieobecności uczniów z datą i godziną kiedy konkretne zajęcia nie mogły się odbyć
- blokowanie dodawania zajęć do planu zajęć jeśli dana klasa ma już wtedy zajęcia, dodany przedmiot nie istnieje albo dany przedmiot jest przeznaczony dla innej klasy
- blokowanie dodanie nauczyciela który nie ma wprowadzonych poprawnie nazw przedmiotów głównych i pobocznych oraz przedmiotu w poszczególnych kolumnach nie mogą się powtarzać

Założenia:

- nie ma ocen z plusami i minusami (np. 3- albo 3+) oraz ocen półwkowych (np. 3.5)
- każdy nauczyciel ma tylko 1 przedmiot główny – to znaczy może prowadzić tylko 1 przedmiot oraz dokładnie 2 przedmioty poboczne, których może uczyć na zastępstwach
- oceny i nieobecności, które dodałem przed utworzeniem triggerów mają „losowe” daty z semestru zimowego 2017. Oceny i nieobecności dodałem w celu sprawdzenia działania innych funkcji i procedur, nie było potrzeby w tym przypadku trzymać się dat.
- niektóre procedury mają dokładnie takie samo działanie jak funkcje. Spowodowane jest to problemami jakie miałem podczas prób zagnieżdżenia procedur wewnątrz funkcji. Problemy te nie występowały gdy zagnieżdżałem funkcje wewnątrz funkcji. Z tego też powodu aby ilość różnych procedur, funkcji i widoków „się zgadzała” dodałem nadmiarowe funkcje i widoki.
- jeśli chodzi o strategię pielęgnacji bazy danych to doszedłem do wniosku, że nie wszystkim tabelom można by robić kopie zapasowe z taką samą częstotliwością. Np. tabela zawierająca oceny jest dużo ważniejsza od tej zawierającej nieobecności (mniejsza szkoda jak komuś się nie wpisze nieobecności). Można by więc ustawić częstotliwość robienia kopii zapasowych dla ocen każdego dnia wieczorem, zaś dla pozostałych mniej ważnych tabel ustawić na raz w tygodniu w weekend.

Dla ułatwienia nawigacji po moim projekcie przedstawię teraz kilka najważniejszych tabel:

PlanyZajęć:

1a

	Poniedziałek	Wtorek	Środa	Czwartek	Piątek
1	J.angielski	J.polski	Mat. Roz.	Fizyka	Wos
2	Mat. Roz.	Mat. Roz.	Historia	Fizyka	J.angielski
3	Mat. Roz.	J.angielski	J.polski	Historia	Geografia
4	J.niemiecki	J.angielski	Chemia	Biologia	J.polski
5	J.niemiecki	Geografia	WF	Informatyka	Informatyka
6	Edb	WF	Informatyka		Religia
7		Religia	Wok		

1b

	Poniedziałek	Wtorek	Środa	Czwartek	Piątek
1	J.angielski	Matematyka	Historia	Matematyka	Łacina
2	Historia	J.polski	J.polski	Łacina	J.angielski
3	Matematyka	J.angielski	Chemia	Biologia	J.polski
4	J.niemiecki	J.angielski	Biologia	Fizyka	Wos
5	J.niemiecki	Wf	Wok	Chemia	Geografia
6	Łacina	Religia	Wf	J.niemiecki	Informatyka
7	Edb				Religia

1c

	Poniedziałek	Wtorek	Środa	Czwartek	Piątek
1	J.angielski	Edb	Matematyka	J.polski roz	J.polski roz
2	Matematyka	Matematyka	J.polski roz	Biologia	J.angielski
3	Historia	J.angielski	Historia	Fizyka	Informatyka
4	J.niemiecki	J.angielski	WF	Historia	Geografia
5	J.niemiecki	Wos	Chemia	Historia	Wos
6	J.polski roz	Geografia	Wok	J.niemiecki	
7		WF	Religia		
8		Religia			

Nauczyciele:

Id	Nazwisko	Imie	PrzedmiotGlowny	PrzedmiotPoboczny	PrzedmiotPoboczny2
1	Atryda	Paul	Fizyka	Informatyka	Matematyka
2	Harkonnen	Jessica	Matematyka	Matematyka rozszerzona	Chemia
3	Kynes	Chani	Matematyka rozszerzona	Matematyka	Informatyka
4	Harkonnen	Vladimir	Historia	Wiedza o społeczeństwie	Geografia

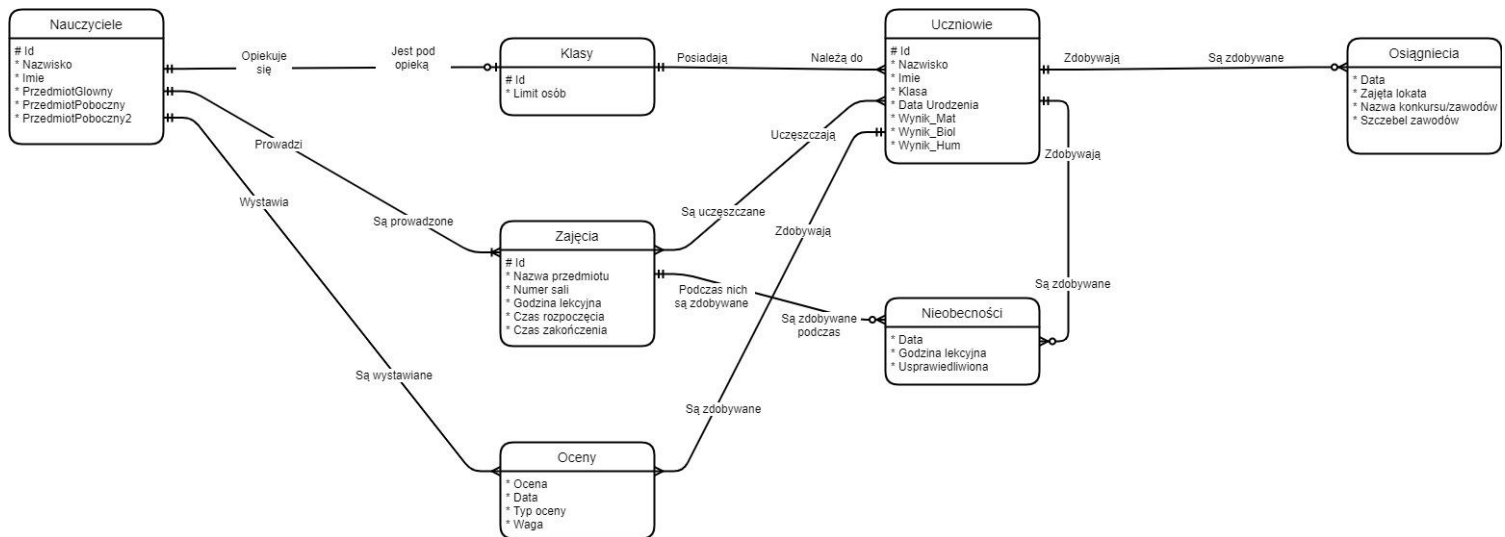
5	Yueh	Wellington	Biologia	Chemia	Wychowanie fizyczne
6	Halleck	Gurney	Wiedza o kulturze	Wiedza o społeczeństwie	Wychowanie fizyczne
7	Rautha	Feyd	Chemia	Biologia	Informatyka
8	Fifrawi	Stilgar	Język polski	Język polski rozszerzony	Religia
9	Corrino	Irulan	Język polski rozszerzony	Język polski	Wiedza o kulturze
10	Atryda	Leto	Informatyka	Matematyka	Geografia
11	Idaho	Duncan	Geografia	Język polski	Język angielski
12	Corrino	Shaddam	Wiedza o społeczeństwie	Język niemiecki	Religia
13	Atryda	Alia	Religia	Edukacja dla bezpieczeństwa	Wiedza o kulturze
14	Hawat	Thufir	Wychowanie fizyczne	Edukacja dla bezpieczeństwa	Religia
15	Fenring	Hasimir	Język angielski	Religia	Podstawy Języka Łacińskiego
16	Mohiam	Gaius	Język angielski	Fizyka	Informatyka
17	Sant	Alia	Język angielski	Geografia	Wiedza o kulturze
18	de Vries	Piter	Język niemiecki	Wychowanie fizyczne	Podstawy Języka Łacińskiego
19	Harkoonen	Feyd	Język niemiecki	Historia	Język polski
20	Rabban	Glossu	Język niemiecki	Język polski	Wiedza o kulturze
21	Nefud	Iakin	Podstawy Języka Łacińskiego	Język angielski	Biologia
22	Mohiam	Helen	Edukacja dla bezpieczeństwa	Język angielski	Chemia

Uczniowie

Id	Nazwisko	Imie	Klasa	Data Urodzenia	Wynik_Mat	Wynik_Biol	Wynik_Hum
1	Skywalker	Anakin	1b	1999-01-01	91	90	31
2	Kenobi	Obi-Wan	1b	1999-07-07	60	98	40
3	Windu	Mace	1b	1999-01-26	80	90	60
4	Ahsoka	Tano	1b	1999-02-13	51	85	33
5	Fisto	Kit	1b	1999-12-29	60	87	40
6	Kolar	Eeth	1b	1999-11-03	46	71	45
7	Tii	Shaak	1b	1999-04-12	50	71	31
8	Vos	Quinlan	1b	1999-08-19	70	80	90
9	Yoda	Master	1b	1999-04-12	82	100	70
10	Mundi	Ki-Adi	1b	1999-09-21	46	78	60
11	Koon	Plo	1b	1999-10-11	61	92	53
12	Jinn	Qui-Gon	1b	1999-02-14	76	90	40
13	Skywalker	Luke	1b	1999-05-04	82	95	60
14	Solo	Han	1b	1999-09-25	50	71	78
15	Amidala	Padme	1b	1999-08-13	60	92	97
16	Calrissian	Lando	1b	1999-05-04	73	82	43
17	Palpatine	Sheev	1a	1999-12-12	99	60	60
18	Tarkin	Wilhuff	1a	1999-05-23	88	45	80
19	Binks	Jar-Jar	1a	1999-06-28	100	34	65
20	Marek	Galen	1a	1999-02-11	88	68	37
21	Revan	Mikołaj	1a	1999-04-12	70	54	93
22	Wulff	Yularen	1a	1999-11-01	68	23	49
23	Maul	Dariusz	1a	1999-12-01	88	42	84
24	Ren	Kylo	1a	1999-04-16	91	74	48
25	Serenno	Dooku	1a	1999-01-21	81	59	47
26	Hut	Jabba	1a	1999-03-03	77	34	84
27	Bane	Cad	1a	1999-12-09	72	55	40

28	Plagaeus	Darian	1a	1999-11-18	71	57	92
29	Grievous	Qymaen	1a	1999-11-21	79	34	48
30	Opress	Savage	1a	1999-04-13	72	60	65
31	Fett	Boba	1a	1999-04-13	78	80	67
32	Kirk	James	1c	1999-12-12	34	63	90
33	McCoy	Leonard	1c	1999-06-28	45	34	70
34	Sulu	Hikaru	1c	1999-04-12	43	32	68
35	Chekov	Pavel	1c	1999-11-01	34	44	82
36	Uhura	Nyota	1c	1999-12-01	75	54	91
37	Picard	Jean-Luc	1c	1999-04-16	77	71	78
38	Riker	William	1c	1999-01-21	44	55	74
39	La Forge	Georgi	1c	1999-03-03	56	82	88
40	Crusher	Beverly	1c	1999-12-09	55	49	81
41	Troi	Deanna	1c	1999-11-18	67	54	70
42	Yar	Tasha	1c	1999-11-21	39	94	71
43	Crusher	Wesley	1c	1999-04-13	66	45	72
44	Pulaski	Catherine	1c	13.04.1999	70	64	89
45	Sisko	Benjamin	1c	1999-02-11	72	34	68

Model ER



Schemat bazy danych (wersja bez krawędzi)

Nauczyciele	
	Id int NOT NULL
Nazwisko nvarchar(50) NOT NULL	
Imię nvarchar(50) NOT NULL	
"PrzedmiotGłówny" nvarchar(50) NOT NULL	
"PrzedmiotPoboczny" nvarchar(50) NOT NULL	
"PrzedmiotPoboczny2" nvarchar(50) NOT NULL	

Klasy	
PK	Id char(2) NOT NULL
"Limit osob" int NOT NULL	
FK1	Id_wychowawcy int NOT NULL

Uczniowie	
PK	Id int NOT NULL
Nazwisko nvarchar(50) NOT NULL	
Imię nvarchar(50) NOT NULL	
FK1	Klasa char(2) NOT NULL
[Data urodzenia] date NOT NULL	
Wynik_Mat INT NOT NULL	
Wynik_Biol INT NOT NULL	
Wynik_Hum INT NOT NULL	

Osiągnięcia	
FK1	"Id Ucznia" int NOT NULL
"Data" date NOT NULL	
"Lokata" nvarchar(20) NOT NULL	
"Nazwa konkursu/zawodów" nvarchar(50) NOT NULL	
"Szczebel zawodów" nvarchar(10) NOT NULL	

Zajęcia	
PK	Id nvarchar(10) NOT NULL
"Nazwa przedmiotu" nvarchar(50) NOT NULL	
FK1	"Id nauczyciela" int NOT NULL
"Numer sal" int NOT NULL	

PlanZajęć	
FK1	"Klasa" char(2) NOT NULL
FK2	"ID Przedmiotu" nvarchar(10) NOT NULL
"Dzień" nvarchar(12) NOT NULL	
FK3	"Godzina lekcyjna" int NOT NULL

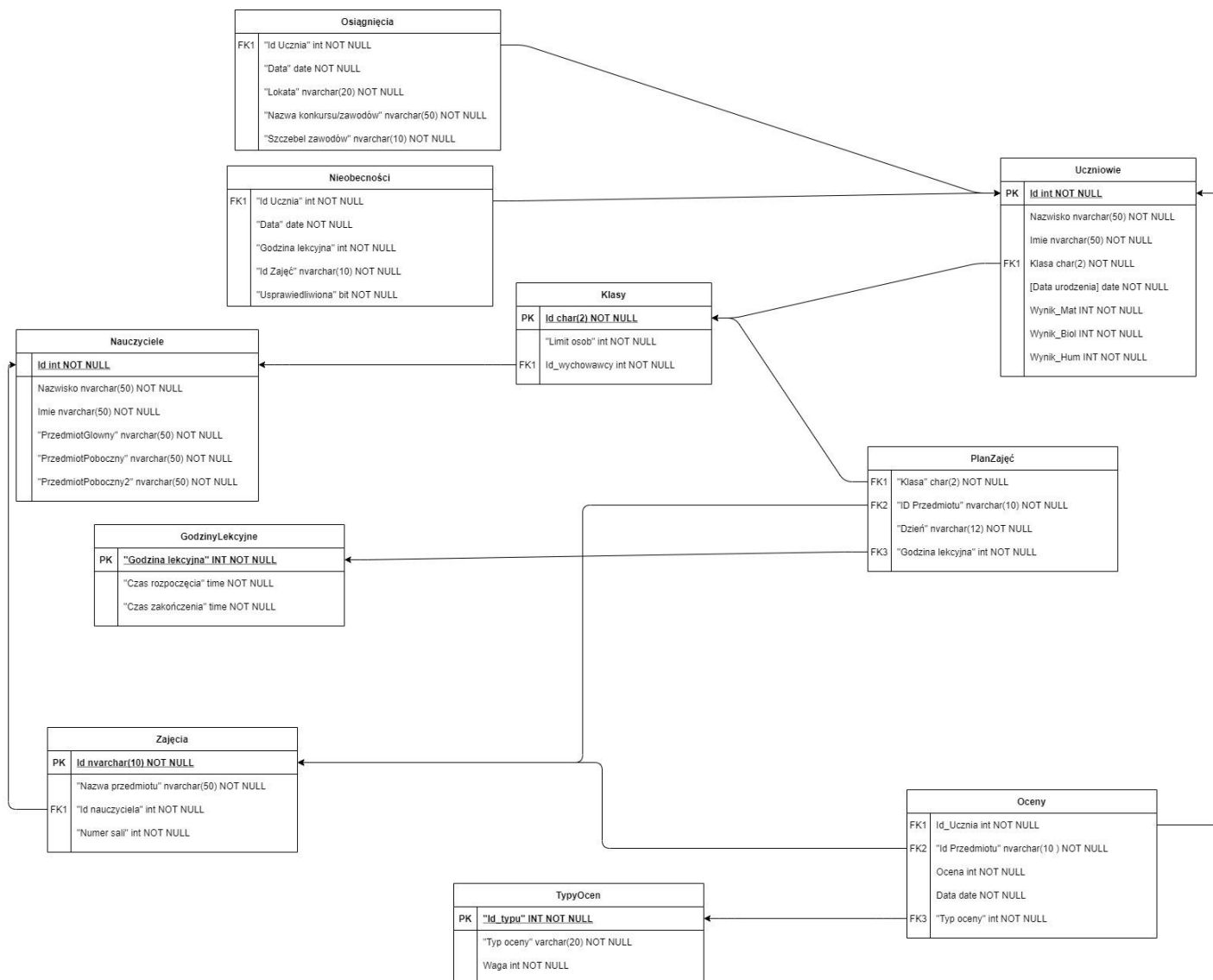
GodzinyLekcyjne	
PK	"Godzina lekcyjna" INT NOT NULL
"Czas rozpoczęcia" time NOT NULL	
"Czas zakończenia" time NOT NULL	

TypyOcen	
PK	"Id_typu" INT NOT NULL
"Typ oceny" varchar(20) NOT NULL	
Waga int NOT NULL	

Oceny	
FK1	Id_Ucznia int NOT NULL
FK2	"Id Przedmiotu" nvarchar(10) NOT NULL
Ocena int NOT NULL	
Data date NOT NULL	
FK3	"Typ oceny" int NOT NULL

Nieobecności	
FK1	"Id Ucznia" int NOT NULL
"Data" date NOT NULL	
"Godzina lekcyjna" int NOT NULL	
"Id Zajęć" nvarchar(10) NOT NULL	
"Usprawiedliwiona" bit NOT NULL	

Schemat bazy danych (z dorysowanymi krawędziami pomiędzy odpowiednimi kluczami)



Skrypt SQL:

```

CREATE TABLE Nauczyciele(
  Id INT PRIMARY KEY IDENTITY(1,1),
  Nazwisko NVARCHAR(50) NOT NULL,
  Imię NVARCHAR(50) NOT NULL,
  [Przedmiot Główny] NVARCHAR(50) NOT NULL,
  [Przedmiot Poboczny] NVARCHAR(50) DEFAULT NULL,
  [Przedmiot Poboczny 2] NVARCHAR(50) DEFAULT NULL
)

INSERT INTO Nauczyciele VALUES
('Atryda', 'Paul', 'Fizyka', 'Informatyka', 'Matematyka'),

```

```
(
'Harkonnen','Jessica','Matematyka','Matematyka rozszerzona','Chemia'),
('Kynes','Chani','Matematyka rozszerzona','Matematyka','Informatyka'),
('Harkonnen','Vladimir','Historia','Wiedza o społeczeństwie','Geografia'),
('Yueh','Wellington','Biologia','Chemia','Wychowanie fizyczne'),
('Halleck','Gurney','Wiedza o kulturze','Wiedza o społeczeństwie','Wychowanie
fizyczne'),
('Rautha','Feyd','Chemia','Biologia','Informatyka'),
('Fifrawi','Stilgar','Język polski','Język polski rozszerzony','Religia'),
('Corrino','Irulan','Język polski rozszerzony','Język polski','Wiedza o kulturze'),
('Atryda','Leto','Informatyka','Matematyka','Geografia'),
('Idaho','Duncan','Geografia','Język polski','Język angielski'),
('Corrino','Shaddam','Wiedza o społeczeństwie','Język niemiecki','Religia'),
('Atryda','Alia','Religia','Edukacja dla bezpieczeństwa','Wiedza o kulturze'),
('Hawat','Thufir','Wychowanie fizyczne','Edukacja dla bezpieczeństwa','Religia'),
('Fenring','Hasimir','Język angielski','Religia','Podstawy języka łacińskiego'),
('Mohiam','Gaius','Język angielski','Fizyka','Informatyka'),
('Sant','Alia','Język angielski','Geografia','Wiedza o kulturze'),
('de Vries','Piter','Język niemiecki','Wychowanie fizyczne','Podstawy języka
łacińskiego'),
('Harkoonen','Feyd','Język niemiecki','Historia','Język polski'),
('Rabban','Glossu','Język niemiecki','Język polski','Wiedza o kulturze'),
('Nefud','Iakin','Podstawy języka łacińskiego','Język angielski','Biologia'),
('Mohiam','Helen','Edukacja dla bezpieczeństwa','Język angielski','Chemia')
)
```

```
CREATE TABLE Klasy(
Id CHAR(2) PRIMARY KEY,
[Limit osob] INT NOT NULL,
[Id wychowawcy] INT NOT NULL FOREIGN KEY REFERENCES Nauczyciele (Id),
)
```

```
INSERT INTO Klasy VALUES
('1a', 18, 1),
('1b', 16, 2),
('1c', 16, 3)
```

```
CREATE TABLE Uczniowie(
Id INT PRIMARY KEY IDENTITY(1,1),
Nazwisko NVARCHAR(50) NOT NULL,
Imie NVARCHAR(50) NOT NULL,
Klasa CHAR(2) NOT NULL FOREIGN KEY REFERENCES Klasy (Id),
[Data urodzenia] DATE,
[Wynik Matematyka] INT NOT NULL CHECK ([Wynik Matematyka] >= 0 AND [Wynik
Matematyka] <= 100),
[Wynik Biologia] INT NOT NULL CHECK ([Wynik Biologia] >= 0 AND [Wynik Biologia] <=
100),
[Wynik Humanistyczne] INT NOT NULL CHECK ([Wynik Humanistyczne] >= 0 AND [Wynik
Humanistyczne] <= 100)
)
```

```
INSERT INTO Uczniowie VALUES
('Skywalker','Anakin','1b','1999-01-01', 91, 90, 31),
('Kenobi','Obi-Wan','1b','1999-07-07', 60, 98, 40),
('Windu','Mace','1b','1999-01-26', 80, 90, 60),
('Ahsoka','Tano','1b','1999-02-13', 51, 85, 33),
('Fisto','Kit','1b','1999-12-29', 60, 87, 40),
('Kolar','Eeth','1b','1999-11-03', 46, 71, 45),
('Tii','Shaak','1b','1999-04-12', 50, 71, 31),
('Vos','Quinlan','1b','1999-08-19', 70, 80, 90),
('Yoda','Master','1b','1999-04-12', 82, 100, 70),
('Mundi','Ki-Adi','1b','1999-09-21', 46, 78, 60),
('Koon','Plo','1b','1999-10-11', 61, 92, 53),
('Jinn','Qui-Gon','1b','1999-02-14', 76, 90, 40),
('Skywalker','Luke','1b','1999-05-04', 82, 95, 60),
('Solo','Han','1b','1999-09-25', 50, 71, 78),
('Amidala','Padme','1b','1999-08-13', 60, 92, 97),
```

```
( 'Calrissian', 'Lando', '1b', '1999-05-04', 73, 82, 43),

( 'Palpatine', 'Sheev', '1a', '1999-12-12', 99, 60, 60),
( 'Tarkin', 'Wilhuff', '1a', '1999-05-23', 88, 45, 80),
( 'Binks', 'Jar-Jar', '1a', '1999-06-28', 100, 34, 65),
( 'Marek', 'Galen', '1a', '1999-02-11', 88, 68, 37),
( 'Revan', 'Mikołaj', '1a', '1999-04-12', 70, 54, 93),
( 'Wullf', 'Yularen', '1a', '1999-11-01', 68, 23, 49),
( 'Maul', 'Dariusz', '1a', '1999-12-01', 88, 42, 84),
( 'Ren', 'Kylo', '1a', '1999-04-16', 91, 74, 48),
( 'Serenno', 'Dooku', '1a', '1999-01-21', 81, 59, 47),
( 'Hut', 'Jabba', '1a', '1999-03-03', 77, 34, 84),
( 'Bane', 'Cad', '1a', '1999-12-09', 72, 55, 40),
( 'Plagaeus', 'Darian', '1a', '1999-11-18', 71, 57, 92),
( 'Grievous', 'Qymaen', '1a', '1999-11-21', 79, 34, 48),
( 'Opress', 'Savage', '1a', '1999-04-13', 72, 60, 65),
( 'Fett', 'Boba', '1a', '1999-04-13', 78, 80, 67),

( 'Kirk', 'James', '1c', '1999-12-12', 34, 63, 90),
( 'McCoy', 'Leonard', '1c', '1999-06-28', 45, 34, 70),
( 'Sulu', 'Hikaru', '1c', '1999-04-12', 43, 32, 68),
( 'Chekov', 'Pavel', '1c', '1999-11-01', 34, 44, 82),
( 'Uhura', 'Nyota', '1c', '1999-12-01', 75, 54, 91),
( 'Picard', 'Jean-Luc', '1c', '1999-04-16', 77, 71, 78),
( 'Riker', 'William', '1c', '1999-01-21', 44, 55, 74),
( 'La Forge', 'Georgi', '1c', '1999-03-03', 56, 82, 88),
( 'Crusher', 'Beverly', '1c', '1999-12-09', 55, 49, 81),
( 'Troi', 'Deanna', '1c', '1999-11-18', 67, 54, 70),
( 'Yar', 'Tasha', '1c', '1999-11-21', 39, 94, 71),
( 'Crusher', 'Wesley', '1c', '1999-04-13', 66, 45, 72),
( 'Pulaski', 'Catherine', '1c', '1999-04-13', 70, 64, 89),
( 'Sisko', 'Benjamin', '1c', '1999-02-11', 72, 34, 68)
```

```
CREATE TABLE Zajęcia(
Id NVARCHAR(10) PRIMARY KEY,
[Nazwa przedmiotu] NVARCHAR(50) NOT NULL,
[Id nauczyciela] INT NOT NULL FOREIGN KEY REFERENCES Nauczyciele (Id),
[Numer sali] INT NOT NULL
)
```

```
INSERT INTO Zajęcia VALUES
( 'MAT', 'Matematyka', 2, 13),
( 'MAT_ROZ', 'Matematyka rozszerzona', 3, 2),
( 'JPOL', 'Język polski', 8, 10),
( 'JPOL_ROZ', 'Język polski rozszerzony', 8, 10),
( 'HIST', 'Historia', 4, 12),
( 'BIOL', 'Biologia', 5, 19),
( 'CHEM', 'Chemia', 7, 14),
( 'FIZ', 'Fizyka', 1, 15),
( 'GEO', 'Geografia', 11, 16),
( 'WOS', 'Wiedza o społeczeństwie', 12, 11),
( 'WOK', 'Wiedza o kulturze', 6, 18),
( 'INF', 'Informatyka', 10, 20),
( 'WF', 'Wychowanie fizyczne', 14, 1),
( 'EDB', 'Edukacja dla bezpieczeństwa', 22, 9),
( 'JNIEM_a', 'Język niemiecki', 18, 3),
( 'JNIEM_b', 'Język niemiecki', 19, 4),
( 'JNIEM_c', 'Język niemiecki', 20, 5),
( 'JANG_a', 'Język angielski', 17, 6),
( 'JANG_b', 'Język angielski', 16, 7),
( 'JANG_c', 'Język angielski', 15, 8),
( 'REL', 'Religia', 13, 22),
( 'PJŁ', 'Podstawy języka łacińskiego', 21, 17)
```

```
CREATE TABLE GodzinyLekcyjne(
```



```
[Godzina lekcyjna] INT PRIMARY KEY NOT NULL IDENTITY(0,1),
[Czas rozpoczęcia] TIME NOT NULL,
[Czas zakończenia] TIME NOT NULL
)
```

```
INSERT INTO GodzinyLekcyjne VALUES
```

```
('7:10', '7:55'),
('8:00', '8:45'),
('8:55', '9:40'),
('9:50', '10:35'),
('10:55', '11:40'),
('11:50', '12:35'),
('12:45', '13:30'),
('13:40', '14:25'),
('14:35', '15:20')
```

```
CREATE TABLE PlanZajęć(
```

```
[Klasa/Grupa] CHAR(2) NOT NULL FOREIGN KEY REFERENCES Klasy (Id),
[ID Przedmiotu] NVARCHAR(10) NOT NULL FOREIGN KEY REFERENCES Zajęcia (Id),
[Dzień] NVARCHAR(12) NOT NULL,
[Godzina lekcyjna] INT NOT NULL FOREIGN KEY REFERENCES GodzinyLekcyjne ([Godzina lekcyjna])
)
```

```
INSERT INTO PlanZajęć VALUES
```

```
('1a', 'MAT_ROZ', 'Poniedziałek', 2),
('1a', 'MAT_ROZ', 'Poniedziałek', 3),
('1a', 'MAT_ROZ', 'Wtorek', 2),
('1a', 'MAT_ROZ', 'Środa', 1),
('1b', 'MAT', 'Poniedziałek', 3),
('1b', 'MAT', 'Wtorek', 1),
('1b', 'MAT', 'Czwartek', 1),
('1c', 'MAT', 'Poniedziałek', 2),
('1c', 'MAT', 'Wtorek', 2),
('1c', 'MAT', 'Środa', 1),
('1a', 'JPOL', 'Wtorek', 1),
('1a', 'JPOL', 'Środa', 3),
('1a', 'JPOL', 'Piątek', 4),
('1b', 'JPOL', 'Wtorek', 2),
('1b', 'JPOL', 'Środa', 2),
('1b', 'JPOL', 'Piątek', 3),
('1c', 'JPOL_ROZ', 'Poniedziałek', 6),
('1c', 'JPOL_ROZ', 'Środa', 2),
('1c', 'JPOL_ROZ', 'Czwartek', 1),
('1c', 'JPOL_ROZ', 'Piątek', 1),
('1a', 'HIST', 'Środa', 2),
('1a', 'HIST', 'Czwartek', 3),
('1b', 'HIST', 'Poniedziałek', 2),
('1b', 'HIST', 'Środa', 1),
('1c', 'HIST', 'Poniedziałek', 3),
('1c', 'HIST', 'Środa', 3),
('1c', 'HIST', 'Czwartek', 4),
('1c', 'HIST', 'Czwartek', 5),
('1a', 'BIOL', 'Czwartek', 4),
('1b', 'BIOL', 'Środa', 4),
('1b', 'BIOL', 'Czwartek', 3),
('1c', 'BIOL', 'Czwartek', 2),
('1a', 'CHEM', 'Środa', 4),
('1b', 'CHEM', 'Środa', 3),
('1b', 'CHEM', 'Czwartek', 5),
('1c', 'CHEM', 'Środa', 5),
('1a', 'FIZ', 'Czwartek', 1),
('1a', 'FIZ', 'Czwartek', 2),
('1b', 'FIZ', 'Czwartek', 4),
('1c', 'FIZ', 'Czwartek', 3),
```

```

('1a', 'GEO', 'Wtorek', 5),
('1a', 'GEO', 'Piątek', 3),
('1b', 'GEO', 'Piątek', 5),
('1c', 'GEO', 'Wtorek', 6),
('1c', 'GEO', 'Piątek', 4),
('1a', 'WOS', 'Piątek', 1),
('1b', 'WOS', 'Piątek', 4),
('1c', 'WOS', 'Wtorek', 5),
('1c', 'WOS', 'Piątek', 5),
('1a', 'WOK', 'Środa', 7),
('1b', 'WOK', 'Środa', 5),
('1c', 'WOK', 'Środa', 6),
('1a', 'INF', 'Środa', 6),
('1a', 'INF', 'Czwartek', 5),
('1a', 'INF', 'Piątek', 5),
('1b', 'INF', 'Piątek', 6),
('1c', 'INF', 'Piątek', 3),
('1a', 'WF', 'Wtorek', 6),
('1a', 'WF', 'Środa', 5),
('1b', 'WF', 'Wtorek', 5),
('1b', 'WF', 'Środa', 6),
('1c', 'WF', 'Wtorek', 7),
('1c', 'WF', 'Środa', 4),
('1b', 'PJŁ', 'Poniedziałek', 6),
('1b', 'PJŁ', 'Czwartek', 2),
('1b', 'PJŁ', 'Piątek', 1),
('1a', 'EDB', 'Poniedziałek', 6),
('1b', 'EDB', 'Poniedziałek', 7),
('1c', 'EDB', 'Wtorek', 1),
('1a', 'REL', 'Wtorek', 7),
('1a', 'REL', 'Piątek', 6),
('1b', 'REL', 'Wtorek', 6),
('1b', 'REL', 'Piątek', 7),
('1c', 'REL', 'Wtorek', 8),
('1c', 'REL', 'Środa', 7),
('1a', 'JANG_a', 'Poniedziałek', 1),
('1a', 'JANG_a', 'Wtorek', 3),
('1a', 'JANG_a', 'Wtorek', 4),
('1a', 'JANG_a', 'Piątek', 2),
('1b', 'JANG_b', 'Poniedziałek', 1),
('1b', 'JANG_b', 'Wtorek', 3),
('1b', 'JANG_b', 'Wtorek', 4),
('1b', 'JANG_b', 'Piątek', 2),
('1c', 'JANG_c', 'Poniedziałek', 1),
('1c', 'JANG_c', 'Wtorek', 3),
('1c', 'JANG_c', 'Wtorek', 4),
('1c', 'JANG_c', 'Piątek', 2),
('1a', 'JNIEM_a', 'Poniedziałek', 4),
('1a', 'JNIEM_a', 'Poniedziałek', 5),
('1a', 'JNIEM_a', 'Czwartek', 6),
('1b', 'JNIEM_b', 'Poniedziałek', 4),
('1b', 'JNIEM_b', 'Poniedziałek', 5),
('1b', 'JNIEM_b', 'Czwartek', 6),
('1c', 'JNIEM_c', 'Poniedziałek', 4),
('1c', 'JNIEM_c', 'Poniedziałek', 5),
('1c', 'JNIEM_c', 'Czwartek', 6)

```

```

CREATE TABLE TypyOcen(
[Id typu] INT PRIMARY KEY IDENTITY(1,1),
[Typ oceny] VARCHAR(20) NOT NULL,
Waga INT NOT NULL
)

```

```

INSERT INTO TypyOcen VALUES
('Sprawdzian', 3),
('Kartkówka', 2),

```

```
( 'Odpowiedź ustna', 2),
( 'Zadanie domowe', 2),
( 'Aktywność', 1),
( 'Projekt', 2),
( 'Konkurs', 3)
```

```
CREATE TABLE Oceny(
[Id Ucznia] INT NOT NULL FOREIGN KEY REFERENCES Uczniowie(Id),
[ID Przedmiotu] NVARCHAR(10) NOT NULL FOREIGN KEY REFERENCES Zajęcia(Id),
Ocena INT NOT NULL,
Data DATE NOT NULL,
[Typ oceny] INT NOT NULL FOREIGN KEY REFERENCES TypyOcen([Id typu])
)
```

```
CREATE TABLE Nieobecności(
[Id Ucznia] INT NOT NULL FOREIGN KEY REFERENCES Uczniowie(Id),
Data DATE NOT NULL,
[Godzina lekcyjna] INT NOT NULL FOREIGN KEY REFERENCES GodzinyLekcyjne ([Godzina lekcyjna]),
[Id Zajęcia] NVARCHAR(10) NOT NULL FOREIGN KEY REFERENCES Zajęcia ([Id]),
Usprawiedliwiona BIT NOT NULL
)
```

```
INSERT INTO Nieobecności VALUES
(1, '2017-09-28', 1, 'JANG_b', 0),
(1, '2017-09-28', 2, 'HIST', 0),
(1, '2017-09-28', 3, 'MAT', 0),
(1, '2017-09-28', 4, 'JNIEM_b', 0),
(1, '2017-09-28', 5, 'JNIEM_b', 0),
(1, '2017-09-28', 6, 'PJŁ', 0),
(1, '2017-09-28', 7, 'EDB', 0),
(2, '2017-09-28', 1, 'JANG_B', 1),
(2, '2017-09-28', 2, 'HIST', 1),
(2, '2017-09-28', 3, 'MAT', 1),
(2, '2017-09-28', 4, 'JNIEM_b', 1),
(2, '2017-09-28', 5, 'JNIEM_b', 1),
(2, '2017-09-28', 6, 'PJŁ', 1),
(2, '2017-09-28', 7, 'EDB', 1),
(13, '2017-10-02', 7, 'REL', 0),
(11, '2017-09-28', 7, 'REL', 0),
(8, '2017-11-28', 4, 'BIOL', 1),
(15, '2017-10-02', 5, 'JNIEM_b', 1),
(12, '2017-10-02', 7, 'REL', 0),
(14, '2017-12-13', 6, 'WF', 0),
(13, '2017-10-02', 7, 'REL', 0),
(11, '2017-09-28', 7, 'REL', 0),
(8, '2017-12-12', 4, 'BIOL', 1),
(9, '2017-12-01', 5, 'JNIEM_b', 1),
(3, '2017-09-02', 7, 'REL', 0),
(1, '2017-09-28', 7, 'EDB', 1),
(14, '2017-10-01', 7, 'REL', 0),
(10, '2017-11-04', 7, 'REL', 0),
(3, '2017-12-13', 1, 'MAT', 1),
(8, '2017-09-15', 1, 'MAT', 1),
(11, '2017-12-16', 4, 'FIZ', 1),
(12, '2017-12-01', 5, 'WOK', 1),

(17, '2017-09-02', 3, 'JANG_a', 1),
(17, '2017-09-02', 4, 'JANG_a', 1),
(19, '2017-10-11', 6, 'HIST', 1),
(21, '2017-12-11', 8, 'MAT_ROZ', 1),
(21, '2017-12-11', 4, 'BIOL', 1),
(21, '2017-12-11', 6, 'INF', 1),
```

```

(18, '2017-09-28', 6, 'JNIEM_a', 0),
(21, '2017-10-28', 1, 'WOS', 0),
(25, '2017-11-28', 6, 'REL', 0),
(25, '2017-11-28', 7, 'WOK', 1),
(25, '2017-11-28', 8, 'REL', 1),
(24, '2017-09-15', 7, 'WOK', 1),
(27, '2017-09-15', 8, 'REL', 1),
(27, '2017-11-28', 6, 'EDB', 1),
(24, '2017-11-28', 1, 'FIZ', 1),
(13, '2017-11-02', 7, 'REL', 1),
(11, '2017-12-28', 7, 'REL', 1),
(28, '2017-12-28', 3, 'BIOL', 1),
(15, '2017-09-02', 5, 'JNIEM_a', 1),
(12, '2017-09-02', 7, 'REL', 0),
(14, '2017-09-13', 6, 'WF', 0),
(13, '2017-11-02', 7, 'REL', 1),
(11, '2017-12-28', 1, 'JANG_a', 1),
(28, '2017-10-12', 4, 'JPOL', 1),
(28, '2017-10-12', 5, 'INF', 1),
(28, '2017-10-12', 6, 'REL', 0),
(31, '2017-12-28', 1, 'FIZ', 1),
(24, '2017-12-04', 7, 'WOK', 0),
(24, '2017-12-04', 8, 'JNIEM_a', 0),
(30, '2017-10-13', 6, 'EDB', 1),
(28, '2017-12-15', 1, 'MAT_ROZ', 1),
(31, '2017-09-16', 6, 'REL', 1),
(30, '2017-09-11', 7, 'WOK', 1),

(33, '2017-12-02', 5, 'WOS', 0),
(37, '2017-12-01', 6, 'JANG_c', 1),
(39, '2017-10-11', 6, 'JPOL_ROZ', 1),
(41, '2017-12-11', 8, 'REL', 0),
(41, '2017-12-11', 7, 'WF', 0),
(38, '2017-12-11', 5, 'WOS', 1),
(32, '2017-09-28', 6, 'WOK', 0),
(32, '2017-09-28', 7, 'REL', 0),
(45, '2017-11-28', 3, 'FIZ', 1),
(45, '2017-11-28', 4, 'HIST', 1),
(45, '2017-11-28', 5, 'HIST', 1),
(45, '2017-11-28', 6, 'JNIEM_c', 1),
(38, '2017-09-15', 8, 'REL', 1),
(38, '2017-11-30', 6, 'JPOL_ROZ', 1),
(39, '2017-11-28', 7, 'REL', 1),
(33, '2017-11-02', 7, 'REL', 1),
(41, '2017-12-28', 7, 'REL', 1),
(42, '2017-12-28', 2, 'BIOL', 1),
(34, '2017-09-02', 5, 'CHEM', 1),
(40, '2017-09-02', 7, 'REL', 0),
(32, '2017-09-13', 6, 'WOK', 1),
(32, '2017-09-12', 7, 'REL', 1),
(34, '2017-12-28', 5, 'WOS', 1),
(34, '2017-10-12', 4, 'GEO', 1),
(33, '2017-10-12', 5, 'HIST', 1),
(32, '2017-10-12', 1, 'JPOL_ROZ', 0),
(45, '2017-12-28', 1, 'MAT', 1),
(44, '2017-12-04', 6, 'WOK', 0),
(44, '2017-12-04', 8, 'REL', 0),
(35, '2017-10-13', 1, 'EDB', 0),
(38, '2017-12-15', 1, 'REL', 1),
(34, '2017-09-16', 6, 'JPOL_ROZ', 1),
(36, '2017-09-11', 7, 'WF', 1)

```

```

CREATE TABLE Osiągnięcia(
[Id Ucznia] INT NOT NULL FOREIGN KEY REFERENCES Uczniowie(Id),
Data DATE NOT NULL,
Lokata NVARCHAR(20) NOT NULL,

```

```
[Nazwa konkursu/zawodów] NVARCHAR(50) NOT NULL,
[Szczegół zawodów] NVARCHAR(10) NOT NULL
)
```

```
INSERT INTO Osiągnięcia VALUES
```

```
(15, '2017-11-03', 'Laureat', 'Olimpada Biologiczna', 'Krajowy'),
(1, '2017-10-15', 'Udział', 'Olimpada Chemiczna', 'Wojewódzki'),
(22, '2017-11-04', 'Finalista', 'Olimpada Matematyczna', 'Krajowy'),
(27, '2017-09-22', 'II miejsce', 'Pływackie Mistrzostwa Województwa', 'Wojewódzki'),
(27, '2017-09-22', 'III miejsce', 'Pływackie Mistrzostwa Województwa', 'Wojewódzki'),
(29, '2017-09-22', 'Finalista', 'Pływackie Mistrzostwa Województwa', 'Wojewódzki'),
(30, '2017-09-22', 'Uczestnik', 'Pływackie Mistrzostwa Województwa', 'Wojewódzki'),
(20, '2017-12-06', 'III miejsce', 'Mikołajkowe zmagania matematyczne', 'Powiatowy'),
(30, '2017-12-06', 'I miejsce', 'Mikołajkowe zmagania matematyczne', 'Powiatowy'),
(19, '2017-12-06', 'Finalista', 'Mikołajkowe zmagania matematyczne', 'Powiatowy'),
(5, '2017-12-06', 'Uczestnik', 'Mikołajkowe zmagania matematyczne', 'Powiatowy'),
(1, '2017-12-01', 'Uczestnik', 'Ogólnopolski konkurs wiedzy o baroku', 'Wojewódzki'),
(39, '2017-12-15', 'Finalista', 'Ogólnopolski konkurs wiedzy o baroku', 'Krajowy'),
(43, '2017-12-15', 'Finalista', 'Ogólnopolski konkurs wiedzy o baroku', 'Krajowy'),
(45, '2017-12-15', 'Laureat', 'Ogólnopolski konkurs wiedzy o baroku', 'Krajowy'),
(23, '2017-11-18', 'Finalista', 'Mistrzostwa programistyczne', 'Krajowy'),
(6, '2017-11-18', 'Uczestnik', 'Mistrzostwa programistyczne', 'Krajowy'),
(28, '2017-11-18', 'Uczestnik', 'Mistrzostwa programistyczne', 'Krajowy'),
(3, '2017-11-11', 'Uczestnik', 'Maraton niepodległościowy', 'Powiatowy'),
(12, '2017-11-11', 'Uczestnik', 'Maraton niepodległościowy', 'Powiatowy'),
(13, '2017-11-11', 'Uczestnik', 'Maraton niepodległościowy', 'Powiatowy'),
(21, '2017-11-11', 'Uczestnik', 'Maraton niepodległościowy', 'Powiatowy'),
(44, '2017-11-11', 'Uczestnik', 'Maraton niepodległościowy', 'Powiatowy'),
(23, '2017-11-11', 'Uczestnik', 'Maraton niepodległościowy', 'Powiatowy')
```

```
INSERT INTO Oceny VALUES
```

```
(1, 'BIOL', 2, '2017-11-03', 1),
(2, 'BIOL', 3, '2017-11-03', 1),
(3, 'BIOL', 3, '2017-11-03', 1),
(4, 'BIOL', 4, '2017-11-03', 1),
(5, 'BIOL', 3, '2017-11-03', 1),
(6, 'BIOL', 3, '2017-11-03', 1),
(7, 'BIOL', 2, '2017-11-03', 1),
(8, 'BIOL', 4, '2017-11-03', 1),
(9, 'BIOL', 5, '2017-11-03', 1),
(10, 'BIOL', 3, '2017-11-03', 1),
(11, 'BIOL', 2, '2017-11-03', 1),
(12, 'BIOL', 4, '2017-11-03', 1),
(13, 'BIOL', 6, '2017-11-03', 1),
(14, 'BIOL', 5, '2017-11-03', 1),
(15, 'BIOL', 4, '2017-11-03', 1),
(16, 'BIOL', 2, '2017-11-03', 1),
```

```
(1, 'BIOL', 3, '2017-12-15', 1),
(2, 'BIOL', 2, '2017-12-15', 1),
(3, 'BIOL', 2, '2017-12-15', 1),
(4, 'BIOL', 3, '2017-12-15', 1),
(5, 'BIOL', 2, '2017-12-15', 1),
(6, 'BIOL', 2, '2017-12-15', 1),
(7, 'BIOL', 3, '2017-12-15', 1),
(8, 'BIOL', 4, '2017-12-15', 1),
(9, 'BIOL', 4, '2017-12-15', 1),
(10, 'BIOL', 3, '2017-12-15', 1),
(11, 'BIOL', 3, '2017-12-15', 1),
(12, 'BIOL', 6, '2017-12-15', 1),
(13, 'BIOL', 5, '2017-12-15', 1),
(14, 'BIOL', 4, '2017-12-15', 1),
(15, 'BIOL', 4, '2017-12-15', 1),
(16, 'BIOL', 3, '2017-12-15', 1),
```

(1, 'BIOL', 4, '2017-09-20', 3),
(2, 'BIOL', 4, '2017-10-13', 3),
(3, 'BIOL', 5, '2017-10-13', 3),
(4, 'BIOL', 3, '2017-11-03', 3),
(5, 'BIOL', 4, '2017-10-27', 3),
(6, 'BIOL', 5, '2017-09-20', 3),
(7, 'BIOL', 2, '2017-11-17', 3),
(8, 'BIOL', 3, '2017-11-17', 3),
(9, 'BIOL', 5, '2017-10-20', 3),
(10, 'BIOL', 1, '2017-11-10', 3),
(11, 'BIOL', 3, '2017-11-03', 3),
(12, 'BIOL', 4, '2017-10-27', 3),
(13, 'BIOL', 4, '2017-10-20', 3),
(14, 'BIOL', 3, '2017-11-10', 3),
(15, 'BIOL', 3, '2017-09-20', 3),
(16, 'BIOL', 4, '2017-11-24', 3),

(1, 'MAT', 2, '2017-09-10', 4),
(2, 'MAT', 1, '2017-11-10', 4),
(3, 'MAT', 3, '2017-10-01', 4),
(4, 'MAT', 2, '2017-12-01', 4),
(5, 'MAT', 5, '2017-12-01', 4),
(6, 'MAT', 4, '2017-09-17', 4),
(7, 'MAT', 6, '2017-10-22', 4),
(8, 'MAT', 3, '2017-11-17', 4),
(9, 'MAT', 3, '2017-10-01', 4),
(10, 'MAT', 4, '2017-10-15', 4),
(11, 'MAT', 3, '2017-12-07', 4),
(12, 'MAT', 2, '2017-09-17', 4),
(13, 'MAT', 5, '2017-10-20', 4),
(14, 'MAT', 3, '2017-10-08', 4),
(15, 'MAT', 4, '2017-09-24', 4),
(16, 'MAT', 3, '2017-10-08', 4),

(1, 'MAT', 2, '2017-11-03', 1),
(2, 'MAT', 2, '2017-11-03', 1),
(3, 'MAT', 3, '2017-11-03', 1),
(4, 'MAT', 2, '2017-11-03', 1),
(5, 'MAT', 5, '2017-11-03', 1),
(6, 'MAT', 4, '2017-11-03', 1),
(7, 'MAT', 6, '2017-11-03', 1),
(8, 'MAT', 3, '2017-11-03', 1),
(9, 'MAT', 3, '2017-11-03', 1),
(10, 'MAT', 4, '2017-11-03', 1),
(11, 'MAT', 3, '2017-11-03', 1),
(12, 'MAT', 2, '2017-11-03', 1),
(13, 'MAT', 5, '2017-11-03', 1),
(14, 'MAT', 3, '2017-11-03', 1),
(15, 'MAT', 4, '2017-11-03', 1),
(16, 'MAT', 3, '2017-11-03', 1),

(1, 'MAT', 3, '2017-12-14', 1),
(2, 'MAT', 2, '2017-12-14', 1),
(3, 'MAT', 4, '2017-12-14', 1),
(4, 'MAT', 3, '2017-12-14', 1),
(5, 'MAT', 4, '2017-12-14', 1),
(6, 'MAT', 5, '2017-12-14', 1),
(7, 'MAT', 6, '2017-12-14', 1),
(8, 'MAT', 2, '2017-12-14', 1),
(9, 'MAT', 4, '2017-12-14', 1),
(10, 'MAT', 2, '2017-12-14', 1),
(11, 'MAT', 3, '2017-12-14', 1),
(12, 'MAT', 2, '2017-12-14', 1),
(13, 'MAT', 6, '2017-12-14', 1),
(14, 'MAT', 4, '2017-12-14', 1),
(15, 'MAT', 3, '2017-12-14', 1),

(16, 'MAT', 2, '2017-12-14', 1),

(1, 'JPOL', 4, '2017-10-10', 4),
(2, 'JPOL', 5, '2017-10-10', 4),
(3, 'JPOL', 4, '2017-10-10', 4),
(4, 'JPOL', 3, '2017-10-10', 4),
(5, 'JPOL', 4, '2017-10-10', 4),
(6, 'JPOL', 4, '2017-10-10', 4),
(7, 'JPOL', 4, '2017-10-10', 4),
(8, 'JPOL', 4, '2017-10-10', 4),
(9, 'JPOL', 4, '2017-10-10', 4),
(10, 'JPOL', 5, '2017-10-10', 4),
(11, 'JPOL', 4, '2017-10-10', 4),
(12, 'JPOL', 4, '2017-10-10', 4),
(13, 'JPOL', 5, '2017-10-10', 4),
(14, 'JPOL', 5, '2017-10-10', 4),
(15, 'JPOL', 4, '2017-10-10', 4),
(16, 'JPOL', 5, '2017-10-10', 4),

(1, 'JPOL', 4, '2017-11-04', 4),
(2, 'JPOL', 3, '2017-11-04', 4),
(3, 'JPOL', 3, '2017-11-04', 4),
(4, 'JPOL', 4, '2017-11-04', 4),
(5, 'JPOL', 5, '2017-11-04', 4),
(6, 'JPOL', 3, '2017-11-04', 4),
(7, 'JPOL', 3, '2017-11-04', 4),
(8, 'JPOL', 4, '2017-11-04', 4),
(9, 'JPOL', 5, '2017-11-04', 4),
(10, 'JPOL', 4, '2017-11-04', 4),
(11, 'JPOL', 3, '2017-11-04', 4),
(12, 'JPOL', 5, '2017-11-04', 4),
(13, 'JPOL', 3, '2017-11-04', 4),
(14, 'JPOL', 4, '2017-11-04', 4),
(15, 'JPOL', 3, '2017-11-04', 4),
(16, 'JPOL', 5, '2017-11-04', 4),

(1, 'JPOL', 3, '2017-12-05', 4),
(2, 'JPOL', 2, '2017-12-05', 4),
(3, 'JPOL', 4, '2017-12-05', 4),
(4, 'JPOL', 3, '2017-12-05', 4),
(5, 'JPOL', 4, '2017-12-05', 4),
(6, 'JPOL', 4, '2017-12-05', 4),
(7, 'JPOL', 1, '2017-12-05', 4),
(8, 'JPOL', 5, '2017-12-05', 4),
(9, 'JPOL', 6, '2017-12-05', 4),
(10, 'JPOL', 3, '2017-12-05', 4),
(11, 'JPOL', 2, '2017-12-05', 4),
(12, 'JPOL', 4, '2017-12-05', 4),
(13, 'JPOL', 4, '2017-12-05', 4),
(14, 'JPOL', 5, '2017-12-05', 4),
(15, 'JPOL', 4, '2017-12-05', 4),
(16, 'JPOL', 6, '2017-12-05', 4),

(1, 'CHEM', 4, '2017-10-11', 1),
(2, 'CHEM', 4, '2017-10-11', 1),
(3, 'CHEM', 2, '2017-10-11', 1),
(4, 'CHEM', 3, '2017-10-11', 1),
(5, 'CHEM', 2, '2017-10-11', 1),
(6, 'CHEM', 2, '2017-10-11', 1),
(7, 'CHEM', 3, '2017-10-11', 1),
(8, 'CHEM', 2, '2017-10-11', 1),
(9, 'CHEM', 5, '2017-10-11', 1),
(10, 'CHEM', 2, '2017-10-11', 1),
(11, 'CHEM', 3, '2017-10-11', 1),
(12, 'CHEM', 2, '2017-10-11', 1),
(13, 'CHEM', 4, '2017-10-11', 1),

(14, 'CHEM', 4, '2017-10-11', 1),
(15, 'CHEM', 3, '2017-10-11', 1),
(16, 'CHEM', 3, '2017-10-11', 1),

(1, 'CHEM', 2, '2017-12-02', 1),
(2, 'CHEM', 3, '2017-12-02', 1),
(3, 'CHEM', 4, '2017-12-02', 1),
(4, 'CHEM', 3, '2017-12-02', 1),
(5, 'CHEM', 3, '2017-12-02', 1),
(6, 'CHEM', 3, '2017-12-02', 1),
(7, 'CHEM', 2, '2017-12-02', 1),
(8, 'CHEM', 4, '2017-12-02', 1),
(9, 'CHEM', 6, '2017-12-02', 1),
(10, 'CHEM', 5, '2017-12-02', 1),
(11, 'CHEM', 4, '2017-12-02', 1),
(12, 'CHEM', 3, '2017-12-02', 1),
(13, 'CHEM', 2, '2017-12-02', 1),
(14, 'CHEM', 3, '2017-12-02', 1),
(15, 'CHEM', 4, '2017-12-02', 1),
(16, 'CHEM', 5, '2017-12-02', 1),

(1, 'CHEM', 4, '2017-12-16', 6),
(2, 'CHEM', 2, '2017-12-16', 6),
(3, 'CHEM', 2, '2017-12-16', 6),
(4, 'CHEM', 5, '2017-12-16', 6),
(5, 'CHEM', 4, '2017-12-16', 6),
(6, 'CHEM', 4, '2017-12-16', 6),
(7, 'CHEM', 3, '2017-12-16', 6),
(8, 'CHEM', 3, '2017-12-16', 6),
(9, 'CHEM', 5, '2017-12-16', 6),
(10, 'CHEM', 5, '2017-12-16', 6),
(11, 'CHEM', 2, '2017-12-16', 6),
(12, 'CHEM', 4, '2017-12-16', 6),
(13, 'CHEM', 5, '2017-12-16', 6),
(14, 'CHEM', 4, '2017-12-16', 6),
(15, 'CHEM', 3, '2017-12-16', 6),
(16, 'CHEM', 4, '2017-12-16', 6),

(1, 'HIST', 3, '2017-09-30', 1),
(2, 'HIST', 5, '2017-09-30', 1),
(3, 'HIST', 5, '2017-09-30', 1),
(4, 'HIST', 2, '2017-09-30', 1),
(5, 'HIST', 2, '2017-09-30', 1),
(6, 'HIST', 3, '2017-09-30', 1),
(7, 'HIST', 4, '2017-09-30', 1),
(8, 'HIST', 5, '2017-09-30', 1),
(9, 'HIST', 5, '2017-09-30', 1),
(10, 'HIST', 2, '2017-09-30', 1),
(11, 'HIST', 3, '2017-09-30', 1),
(12, 'HIST', 4, '2017-09-30', 1),
(13, 'HIST', 4, '2017-09-30', 1),
(14, 'HIST', 5, '2017-09-30', 1),
(15, 'HIST', 3, '2017-09-30', 1),
(16, 'HIST', 3, '2017-09-30', 1),

(1, 'HIST', 2, '2017-10-30', 1),
(2, 'HIST', 3, '2017-10-30', 1),
(3, 'HIST', 4, '2017-10-30', 1),
(4, 'HIST', 3, '2017-10-30', 1),
(5, 'HIST', 3, '2017-10-30', 1),
(6, 'HIST', 4, '2017-10-30', 1),
(7, 'HIST', 3, '2017-10-30', 1),
(8, 'HIST', 4, '2017-10-30', 1),
(9, 'HIST', 6, '2017-10-30', 1),
(10, 'HIST', 4, '2017-10-30', 1),
(11, 'HIST', 4, '2017-10-30', 1),

(12, 'HIST', 3, '2017-10-30', 1),
(13, 'HIST', 2, '2017-10-30', 1),
(14, 'HIST', 2, '2017-10-30', 1),
(15, 'HIST', 4, '2017-10-30', 1),
(16, 'HIST', 4, '2017-10-30', 1),

(1, 'HIST', 3, '2017-11-29', 1),
(2, 'HIST', 4, '2017-11-29', 1),
(3, 'HIST', 4, '2017-11-29', 1),
(4, 'HIST', 3, '2017-11-29', 1),
(5, 'HIST', 4, '2017-11-29', 1),
(6, 'HIST', 4, '2017-11-29', 1),
(7, 'HIST', 3, '2017-11-29', 1),
(8, 'HIST', 5, '2017-11-29', 1),
(9, 'HIST', 5, '2017-11-29', 1),
(10, 'HIST', 4, '2017-11-29', 1),
(11, 'HIST', 4, '2017-11-29', 1),
(12, 'HIST', 5, '2017-11-29', 1),
(13, 'HIST', 3, '2017-11-29', 1),
(14, 'HIST', 3, '2017-11-29', 1),
(15, 'HIST', 5, '2017-11-29', 1),
(16, 'HIST', 5, '2017-11-29', 1),

(1, 'FIZ', 3, '2017-10-16', 1),
(2, 'FIZ', 2, '2017-10-16', 1),
(3, 'FIZ', 3, '2017-10-16', 1),
(4, 'FIZ', 4, '2017-10-16', 1),
(5, 'FIZ', 2, '2017-10-16', 1),
(6, 'FIZ', 3, '2017-10-16', 1),
(7, 'FIZ', 4, '2017-10-16', 1),
(8, 'FIZ', 4, '2017-10-16', 1),
(9, 'FIZ', 5, '2017-10-16', 1),
(10, 'FIZ', 4, '2017-10-16', 1),
(11, 'FIZ', 2, '2017-10-16', 1),
(12, 'FIZ', 3, '2017-10-16', 1),
(13, 'FIZ', 3, '2017-10-16', 1),
(14, 'FIZ', 5, '2017-10-16', 1),
(15, 'FIZ', 4, '2017-10-16', 1),
(16, 'FIZ', 3, '2017-10-16', 1),

(1, 'FIZ', 2, '2017-12-14', 1),
(2, 'FIZ', 4, '2017-12-14', 1),
(3, 'FIZ', 4, '2017-12-14', 1),
(4, 'FIZ', 5, '2017-12-14', 1),
(5, 'FIZ', 4, '2017-12-14', 1),
(6, 'FIZ', 3, '2017-12-14', 1),
(7, 'FIZ', 2, '2017-12-14', 1),
(8, 'FIZ', 3, '2017-12-14', 1),
(9, 'FIZ', 6, '2017-12-14', 1),
(10, 'FIZ', 3, '2017-12-14', 1),
(11, 'FIZ', 4, '2017-12-14', 1),
(12, 'FIZ', 5, '2017-12-14', 1),
(13, 'FIZ', 4, '2017-12-14', 1),
(14, 'FIZ', 6, '2017-12-14', 1),
(15, 'FIZ', 3, '2017-12-14', 1),
(16, 'FIZ', 4, '2017-12-14', 1),

(1, 'GEO', 2, '2017-10-21', 1),
(2, 'GEO', 5, '2017-10-21', 1),
(3, 'GEO', 5, '2017-10-21', 1),
(4, 'GEO', 3, '2017-10-21', 1),
(5, 'GEO', 4, '2017-10-21', 1),
(6, 'GEO', 3, '2017-10-21', 1),
(7, 'GEO', 3, '2017-10-21', 1),
(8, 'GEO', 5, '2017-10-21', 1),
(9, 'GEO', 6, '2017-10-21', 1),

```
(10, 'GEO', 3, '2017-10-21', 1),
(11, 'GEO', 4, '2017-10-21', 1),
(12, 'GEO', 5, '2017-10-21', 1),
(13, 'GEO', 4, '2017-10-21', 1),
(14, 'GEO', 3, '2017-10-21', 1),
(15, 'GEO', 2, '2017-10-21', 1),
(16, 'GEO', 2, '2017-10-21', 1),
```

```
(1, 'GEO', 3, '2017-12-19', 1),
(2, 'GEO', 4, '2017-12-19', 1),
(3, 'GEO', 4, '2017-12-19', 1),
(4, 'GEO', 2, '2017-12-19', 1),
(5, 'GEO', 4, '2017-12-19', 1),
(6, 'GEO', 3, '2017-12-19', 1),
(7, 'GEO', 5, '2017-12-19', 1),
(8, 'GEO', 4, '2017-12-19', 1),
(9, 'GEO', 5, '2017-12-19', 1),
(10, 'GEO', 3, '2017-12-19', 1),
(11, 'GEO', 4, '2017-12-19', 1),
(12, 'GEO', 5, '2017-12-19', 1),
(13, 'GEO', 2, '2017-12-19', 1),
(14, 'GEO', 5, '2017-12-19', 1),
(15, 'GEO', 4, '2017-12-19', 1),
(16, 'GEO', 4, '2017-12-19', 1)
```

-- FUNCKJE I WIDOKI

```
-----
IF OBJECT_ID('dbo.MozliwiZastepcy') IS NOT NULL
DROP FUNCTION dbo.MozliwiZastepcy
GO
```

-- Funkcja wypisuje listę nauczycieli, którzy danego dnia i danej godzinie mogą zastąpić danego nauczyciela na jego zajęciach (nie prowadzą własnych lekcji)

```
CREATE FUNCTION MozliwiZastepcy ( @Nazwisko NVARCHAR(50),@Imie NVARCHAR(50), @Dzien
NVARCHAR(12), @NumerZajec INT, @Przedmiot NVARCHAR(50))
RETURNS @ZastepcyTab TABLE
(
    Nazwisko nvarchar(50),
    Imie nvarchar(50)
)
AS
BEGIN
```

-- wypisanie listy nauczycieli, którzy mogą zastąpić danego naczuciela na danym przedmiocie i sprawdzenie czy nie są na liście nauczycieli którzy prowadzą wtedy zajęcia

```
INSERT INTO @ZastepcyTab
SELECT DISTINCT N.Nazwisko, N.Imie
FROM Nauczyciele AS N JOIN (Zajęcia AS Z JOIN PlanZajęc AS PZ ON Z.Id = PZ.[ID
Przedmiotu]) ON N.Id = Z.[Id nauczyciela]
WHERE (N.Nazwisko <> @Nazwisko OR N.Imie <> @Imie) AND (N.[Przedmiot Główny] =
@Przedmiot OR N.[Przedmiot Poboczny] = @Przedmiot OR N.[Przedmiot Poboczny 2] =
@Przedmiot)
AND NOT EXISTS (
    SELECT N2.Nazwisko, N2.Imie
    FROM Nauczyciele AS N2 JOIN Zajęcia AS Z2 ON N2.Id = Z2.[Id nauczyciela] JOIN
PlanZajęc AS PZ2 ON PZ2.[ID Przedmiotu] = Z.Id
```

```

        WHERE PZ2.Dzień = @Dzien AND PZ2.[Godzina lekcyjna] = @NumerZajec
    )
    RETURN
    END

GO

-- Przykładowe zapytania
SELECT *
FROM MozliwiZastepcy('Atryda', 'Paul', 'Poniedziałek', 2, 'Informatyka')

SELECT *
FROM MozliwiZastepcy('Harkoonen', 'Vladimir', 'Środa', 2, 'Historia')

-----

IF OBJECT_ID('dbo.Najlepsza_frekwencja') IS NOT NULL
DROP FUNCTION dbo.Najlepsza_frekwencja
GO

-- Funkcja wypisuje listę uczniów, którzy osiągnęli najlepszą frekwencję w klasie

CREATE FUNCTION dbo.Najlepsza_frekwencja(@Id_Klasy CHAR(2))
RETURNS @Najlepsza_frekwencja_tab TABLE(
    Nazwisko NVARCHAR(50) NOT NULL,
    Imie NVARCHAR(50) NOT NULL,
    [Ile nieobecności] INT NOT NULL
)
AS
BEGIN
    --obliczamy minimalną ilość nieobecności
    DECLARE @minimum INT
    SET @minimum= (SELECT MIN(TMP.Ile_nieobecności)
    FROM (SELECT U.Id, COUNT(N.[Id Zajęć]) Ile_nieobecności
    FROM Uczniowie AS U LEFT JOIN Nieobecności AS N ON N.[Id Ucznia] = U.Id
    WHERE U.Klasa = @Id_Klasy
    GROUP BY U.Id) AS TMP)

    --dodajemy do wynikowej tablicy rekordy z uczniami posiadającymi liczbę nieobecności
    równą tej minimalnej
    INSERT INTO @Najlepsza_frekwencja_tab
    SELECT U.Nazwisko, U.Imie, COUNT(N.[Id Zajęć])
    FROM Uczniowie AS U LEFT JOIN Nieobecności AS N ON N.[Id Ucznia] = U.Id
    WHERE U.Klasa = @Id_Klasy
    GROUP BY U.Id, U.Nazwisko, U.Imie
    HAVING COUNT(N.[Id Zajęć]) = @minimum

    RETURN
    END
GO

-- Przykładowe zapytanie
SELECT *
FROM Najlepsza_frekwencja('1c') -- powinien się wypisać Crusher Wesley

-----
-
IF OBJECT_ID('dbo.Najczęściej_opuszczane_zajęcia') IS NOT NULL
DROP FUNCTION dbo.Najczęściej_opuszczane_zajęcia
GO

-- Funkcja wypisuje ilość nieobecności uczniów na zajęciach konkretnego nauczyciela

CREATE FUNCTION dbo.Najczęściej_opuszczane_zajęcia()
RETURNS @Najczęściej_opuszczane_zajęcia_tab TABLE(

```

```

"Nazwa przedmiotu" NVARCHAR(50) NOT NULL,
[Nazwisko nauczyciela] NVARCHAR(50) NOT NULL,
[Imie nauczyciela] NVARCHAR(50) NOT NULL,
[Ile nieobecności] INT
)
AS
BEGIN

INSERT INTO @Najczęściej_opuszczane_zajęcia_tab
SELECT Z.[Nazwa przedmiotu], Na.Nazwisko, Na.Imie, TMP.Ile_Nieobecności
FROM Zajęcia AS Z LEFT JOIN (
SELECT N.[Id Zajęć], COUNT(*) Ile_Nieobecności
FROM Nieobecności AS N
GROUP BY N.[Id Zajęć]) AS TMP ON Z.[Id] = TMP.[Id Zajęć]
JOIN Nauczyciele AS Na ON Na.Id = Z.[Id nauczyciela]

-- dla czytelności zastąpimy wszystkie NULLe zerami
UPDATE @Najczęściej_opuszczane_zajęcia_tab
SET [Ile nieobecności] = 0
WHERE [Ile nieobecności] IS NULL

RETURN
END
GO

-- Przykładowe zapytanie
SELECT *
FROM Najczęściej_opuszczane_zajęcia()
ORDER BY [Ile nieobecności] DESC

-----

IF OBJECT_ID('DBO.Srednia_z_przedmiotu_fun') IS NOT NULL
IF OBJECT_ID('DBO.Srednia_z_przedmiotu_fun') IS NOT NULL
DROP FUNCTION DBO.Srednia_z_przedmiotu_fun
GO
-- Wylicza dla danego ucznia średnią z przedmiotu
CREATE FUNCTION Srednia_z_przedmiotu_fun (@Id_ucznia INT, @Przedmiot NVARCHAR(50))
RETURNS FLOAT
AS
BEGIN

DECLARE @Srednia FLOAT = 0.0
DECLARE @SUMA_WAG FLOAT = 0.0
DECLARE @SUMA_OCEN FLOAT = 0.0

-- Obliczanie sumy wag
SET @SUMA_WAG = (SELECT SUM(Typ0.Waga)
FROM dbo.Oceny AS O JOIN dbo.Zajęcia AS Z ON O.[ID Przedmiotu] = Z.Id
JOIN dbo.TypyOcen AS Typ0 ON Typ0.[Id typu] = O.[Typ oceny]
WHERE Z.[Nazwa przedmiotu] = @Przedmiot AND O.[Id Ucznia] = @Id_ucznia)

-- Obliczanie sumy ocen razem z wagą
SET @SUMA_OCEN = (SELECT SUM(TMP.Ocena_z_waga)
FROM (SELECT Typ0.Waga * O.Ocena Ocena_z_waga
FROM dbo.Oceny AS O JOIN dbo.Zajęcia AS Z ON O.[ID Przedmiotu] = Z.Id
JOIN dbo.TypyOcen AS Typ0 ON Typ0.[Id typu] = O.[Typ oceny]
WHERE Z.[Nazwa przedmiotu] = @Przedmiot AND O.[Id Ucznia] = @Id_ucznia) AS TMP)

SET @Srednia = ROUND(@SUMA_OCEN / @SUMA_WAG, 2)

RETURN @Srednia
END
GO

```

```

-- Przykładowe zapytanie
SELECT dbo.Srednia_z_przedmiotu_fun(1, 'Biologia')
SELECT dbo.Srednia_z_przedmiotu_fun(2, 'Biologia')
SELECT dbo.Srednia_z_przedmiotu_fun(14, 'Biologia')
SELECT dbo.Srednia_z_przedmiotu_fun(4, 'Matematyka')

-----

IF OBJECT_ID('dbo.OcenyKoncowe') IS NOT NULL
DROP FUNCTION dbo.OcenyKoncowe
GO

-- wypisuje do tablicy oceny końcowe z każdego przedmiotu, gdzie są wpisane
jakikolwiek oceny cząstkowe
CREATE FUNCTION dbo.OcenyKoncowe(@Id_Ucznia INT)
RETURNS @OcenyKoncowe_tab TABLE(
Przedmiot NVARCHAR(50),
Ocena FLOAT
)
AS
BEGIN

-- dla większej czytelności kodu zapiszmy klasę ucznia do zmiennej @Klasa_ucznia
DECLARE @Klasa_Ucznia CHAR(2)
SET @Klasa_Ucznia =
(SELECT Klasa FROM Uczniowie
WHERE Id = @Id_Ucznia)

DECLARE @Przedmiot NVARCHAR(50)

-- tworzymy kursor w którym przechowujemy kolejne nazwy przedmiotów
DECLARE Akt_Przedmiot CURSOR FOR
(SELECT DISTINCT Z.[Nazwa przedmiotu]
FROM PlanZajęć AS PZ JOIN Zajęcia AS Z ON PZ.[ID Przedmiotu] = Z.Id
WHERE PZ.[Klasa/Grupa] = @Klasa_Ucznia)

OPEN Akt_przedmiot
FETCH Akt_przedmiot INTO @Przedmiot

WHILE @@FETCH_STATUS = 0
BEGIN
    DECLARE @TMP FLOAT
    SET @TMP = 0.0

    -- liczenie sredniej przy pomocy zagnieżdżonej funkcji
    SET @TMP = dbo.Srednia_z_przedmiotu_fun(@Id_Ucznia, @Przedmiot)
    SET @TMP = ROUND(@TMP,0)

    -- na wypadek gdy z jakiegoś przedmiotu nie ma ocen
    IF @TMP is not NULL
        INSERT INTO @OcenyKoncowe_tab VALUES(@Przedmiot, @TMP)

    FETCH Akt_przedmiot INTO @Przedmiot
END

CLOSE Akt_przedmiot
DEALLOCATE Akt_przedmiot

RETURN
END
GO

-- Przykładowe zapytania

```

```

SELECT *
FROM OcenKoncowe('1')

SELECT *
FROM OcenKoncowe('2')

SELECT *
FROM OcenKoncowe('14')
-----
IF OBJECT_ID('DBO.SredniaOcenFun') IS NOT NULL
DROP FUNCTION DBO.SredniaOcenFun

GO
-- wypisuje średnią ocen dla danego ucznia
CREATE FUNCTION dbo.SredniaOcenFun (@Id_ucznia INT)
RETURNS FLOAT
AS
BEGIN
DECLARE @Srednia FLOAT
-- korzystamy z zagnieżdżonej funkcji
SET @Srednia = (SELECT AVG(OK.Ocena) FROM
(SELECT * FROM OcenKoncowe(@Id_ucznia)) AS OK
)

SET @Srednia = ROUND(@Srednia, 2)
RETURN @Srednia
END

GO
-- Przykładowe zapytanie

SELECT dbo.SredniaOcenFun(2)
-----
IF OBJECT_ID('dbo.Prymusi') IS NOT NULL
DROP FUNCTION dbo.Prymusi
GO
-- Wypisuje listę prymusów z danej klasy (czyli takie osoby, które osiągnęły średnią
co najmniej 4.75)
CREATE FUNCTION dbo.Prymusi(@IdKlasy CHAR(2))
RETURNS @Prymusi_tab TABLE(
[Nazwisko] NVARCHAR(50) NOT NULL,
[Imie] NVARCHAR(50) NOT NULL,
[Srednia] FLOAT NOT NULL
)
AS
BEGIN

-- tworzymy kursor w którym będziemy przetrzymywac kolejne id ucznia
DECLARE Akt_id CURSOR FOR
(SELECT Id
FROM Uczniowie
WHERE Klasa = @IdKlasy)

DECLARE @id_ucznia INT
OPEN Akt_id
FETCH Akt_id INTO @id_ucznia

WHILE @@FETCH_STATUS = 0
BEGIN
    DECLARE @TMP FLOAT
    -- używamy zagnieżdżonej funkcji
    SET @TMP = (SELECT dbo.SredniaOcenFun(@id_ucznia))

    IF @TMP >= 4.75
    INSERT INTO @Prymusi_tab VALUES

```

```
((SELECT Nazwisko FROM Uczniowie WHERE Id = @id_ucznia),(SELECT Imie FROM
Uczniowie WHERE Id = @id_ucznia), @TMP)
```

```
FETCH Akt_id INTO @id_ucznia
```

```
END
```

```
CLOSE Akt_id
```

```
DEALLOCATE Akt_id
```

```
RETURN
```

```
END
```

```
GO
```

```
-- Przykładowe zapytanie
```

```
SELECT *
FROM Prymusi('1b')
```

```
-----
IF OBJECT_ID('dbo.IleProcentowoNieobecnosci') IS NOT NULL
DROP FUNCTION dbo.IleProcentowoNieobecnosci
GO
```

```
-- wypisuje ile proc z wszystkich nieobecności przypada na konkretną klasę
```

```
CREATE FUNCTION dbo.IleProcentowoNieobecnosci()
RETURNS @IleProcentowoNieobecnosci_tab TABLE(
[ID klasy] CHAR(2) NOT NULL,
[Proc nieobecnosci] FLOAT NOT NULL
)
AS
BEGIN
```

```
-- dla przejrzystości kodu liczbę wszystkich nieobecności będziemy trzymać w zmiennej
@WszystkieNieob
```

```
DECLARE @WszystkieNieob FLOAT
SET @WszystkieNieob = (SELECT COUNT(*) FROM Nieobecności)
```

```
-- tworzymy kursor w którym będziemy trzymać kolejne klasy
```

```
DECLARE Akt_klasa CURSOR FOR
(SELECT Id
FROM Klasy
)
```

```
DECLARE @id_kl CHAR(2)
OPEN Akt_klasa
FETCH Akt_klasa INTO @id_kl
```

```
WHILE @@FETCH_STATUS = 0
BEGIN
```

```
-- liczymy ilość nieobecności dla każdej klasy
```

```
DECLARE @TMP FLOAT
SET @TMP = (SELECT COUNT(*) FROM Nieobecności AS N JOIN Uczniowie AS U ON N.[Id
Ucznia] = U.Id WHERE U.Klasa = @id_kl)
```

```
-- wyliczamy jaki procent z wszystkich nieobecności stanowią te z danej klasy
```

```
INSERT INTO @IleProcentowoNieobecnosci_tab VALUES
(@id_kl, (ROUND(@TMP/@WszystkieNieob, 4) * 100))
```

```
FETCH Akt_klasa INTO @id_kl
```

```
END
```

```
CLOSE Akt_klasa
```

```
DEALLOCATE Akt_klasa
```

```
RETURN
END
GO
```

-- Przykładowe zapytanie

```
SELECT *
FROM IleProcentowoNieobecnosci()

-----

IF OBJECT_ID('dbo.ListaZajetychPracowni') IS NOT NULL
DROP FUNCTION dbo.ListaZajetychPracowni
GO

-- Funkcja wypisze w tabeli listę zajętych sal lekcyjnych zajętych w danym przedziale
-- czasowym
CREATE FUNCTION dbo.ListaZajetychPracowni(@Dzien NVARCHAR(12), @OdKiedy TIME, @DoKiedy
TIME)
RETURNS @ListaZajetychPracowni_tab TABLE(
[Nr klasy] INT NOT NULL
)
AS
BEGIN

INSERT INTO @ListaZajetychPracowni_tab
SELECT DISTINCT Z.[Numer sali] FROM PlanZajęć AS PZ JOIN Zajęcia AS Z ON PZ.[ID
Przedmiotu] = Z.Id
JOIN GodzinyLekcyjne AS GL ON GL.[Godzina lekcyjna] = PZ.[Godzina lekcyjna]
WHERE PZ.Dzień = @Dzien AND (
(GL.[Czas rozpoczęcia] >= @OdKiedy AND GL.[Czas zakończenia] <= @DoKiedy) OR
(@OdKiedy >= GL.[Czas rozpoczęcia] AND @OdKiedy <= GL.[Czas zakończenia]) OR
(@DoKiedy >= GL.[Czas rozpoczęcia] AND @DoKiedy <= GL.[Czas zakończenia]))

RETURN
END
GO
```

-- Przykładowe zapytania

```
SELECT *
FROM ListaZajetychPracowni('Wtorek', '8:00', '9:45')

SELECT *
FROM ListaZajetychPracowni('Czwartek', '8:00', '15:45')

-----

IF OBJECT_ID('dbo.ZajęciaNauczyciela') IS NOT NULL
DROP FUNCTION dbo.ZajęciaNauczyciela
GO

-- Wypisze plan zajęć nauczyciela
CREATE FUNCTION dbo.ZajęciaNauczyciela(@Nazwisko NVARCHAR(50), @Imie NVARCHAR(50))
RETURNS @ZajęciaNauczyciela_tab TABLE(
"Nazwa przedmiotu" NVARCHAR(50) NOT NULL,
"Klasa/Grupa" CHAR(2) NOT NULL,
"Dzień" NVARCHAR(12) NOT NULL,
[Godzina lekcyjna] INT NOT NULL
)
AS
BEGIN

INSERT INTO @ZajęciaNauczyciela_tab
SELECT Z.[Nazwa przedmiotu], PZ.[Klasa/Grupa], PZ.Dzień, PZ.[Godzina lekcyjna]
FROM PlanZajęć AS PZ JOIN Zajęcia AS Z ON PZ.[ID Przedmiotu] = Z.Id
JOIN Nauczyciele AS N ON N.Id = Z.[Id nauczyciela]
WHERE N.Imie = @Imie AND N.Nazwisko = @Nazwisko
ORDER BY PZ.Dzień, PZ.[Godzina lekcyjna]
```



```
RETURN
END
GO
```

-- Przykładowe zapytania

```
SELECT *
FROM ZajęciaNauczyciela('Atryda', 'Paul')
ORDER BY Dzień, [Godzina lekcyjna]
GO
```

```
SELECT *
FROM ZajęciaNauczyciela('Corrino', 'Shaddam')
ORDER BY Dzień, [Godzina lekcyjna]
```

-- widok zawierający uczniów którzy zajęli jakieś miejsca w konkursach/zawodach o szczeblu krajowym

```
CREATE VIEW Laureaci AS
SELECT U.Nazwisko, U.Imie, U.Klasa, O.Lokata, O.[Nazwa konkursu/zawodów]
FROM Osiągnięcia AS O JOIN Uczniowie AS U ON O.[Id Ucznia] = U.Id
WHERE O.Lokata IN ('I miejsce', 'II miejsce', 'III miejsce', 'Laureat') AND
O.[Szczebel zawodów] = 'Krajowy'
GO
```

```
SELECT * FROM Laureaci
```

-- widok wypisujący wychowawców poszczególnych klas

```
CREATE VIEW Wychowawcy AS
SELECT N.Nazwisko, N.Imie, K.Id 'Klasa'
FROM Nauczyciele AS N JOIN Klasy AS K ON N.Id = K.[Id wychowawcy]
GO
```

```
SELECT * FROM Wychowawcy
```

-- widok wypisujący ilość godzin lekcyjnych ile mają poszczególni nauczyciele

```
CREATE VIEW NajbardziejZapracowaniNauczyciele AS
SELECT N.Nazwisko, N.Imie, Z.[Nazwa przedmiotu], COUNT(*) 'ile godzin lekcyjnych'
FROM Nauczyciele AS N JOIN Zajęcia AS Z ON N.Id = Z.[Id nauczyciela] JOIN PlanZajęć AS
PZ ON PZ.[ID Przedmiotu] = Z.Id
GROUP BY N.Nazwisko, N.Imie, Z.[Nazwa przedmiotu]
GO
```

```
SELECT * FROM NajbardziejZapracowaniNauczyciele
ORDER BY [ile godzin lekcyjnych] DESC
```

-- widok obrazujący najczęściej wstawiany typ ocen wraz z ilością ich wstawienia

```
CREATE VIEW NajczęściejWystawianeRodzajeOcen AS
SELECT TYO.[Typ oceny], COUNT(*) 'Ilość wystawionych ocen' FROM TypyOcen AS TYO JOIN
Oceny AS O ON TYO.[Id typu] = O.[Typ oceny]
GROUP BY TYO.[Typ oceny]
GO
```

```
SELECT * FROM NajczęściejWystawianeRodzajeOcen
ORDER BY 'Ilość wystawionych ocen' DESC
```

-- PROCEDURE

```
-----
IF OBJECT_ID('DBO.Srednia_z_przedmiotu','P') IS NOT NULL
DROP PROC DBO.Srednia_z_przedmiotu
GO

-- Procedura zwracająca średnią z zadanego przedmiotu
CREATE PROC Srednia_z_przedmiotu (@Id_ucznia INT, @Przedmiot NVARCHAR(50), @Srednia
FLOAT OUTPUT)
AS
DECLARE @SUMA_WAG FLOAT = 0.0
DECLARE @SUMA_OCEN FLOAT = 0.0

-- Obliczanie sumy wag
SET @SUMA_WAG = (SELECT SUM(Typ0.Waga)
FROM dbo.Oceny AS O JOIN dbo.Zajęcia AS Z ON O.[ID Przedmiotu] = Z.Id
JOIN dbo.TypyOcen AS Typ0 ON Typ0.[Id typu] = O.[Typ oceny]
WHERE Z.[Nazwa przedmiotu] = @Przedmiot AND O.[Id Ucznia] = @Id_ucznia)

-- Obliczanie sumy ocen razem z waga
SET @SUMA_OCEN = (SELECT SUM(TMP.Ocena_z_waga)
FROM (SELECT Typ0.Waga * O.Ocena Ocena_z_waga
FROM dbo.Oceny AS O JOIN dbo.Zajęcia AS Z ON O.[ID Przedmiotu] = Z.Id
JOIN dbo.TypyOcen AS Typ0 ON Typ0.[Id typu] = O.[Typ oceny]
WHERE Z.[Nazwa przedmiotu] = @Przedmiot AND O.[Id Ucznia] = @Id_ucznia) AS TMP)

SET @Srednia = ROUND(@SUMA_OCEN / @SUMA_WAG, 2)

GO

--Przykładowe zapytanie
declare @tmp FLOAT
exec Srednia_z_przedmiotu 1, 'Język polski', @tmp OUTPUT
SELECT @tmp
-----

IF OBJECT_ID('DBO.SredniaOcen','P') IS NOT NULL
DROP PROC DBO.SredniaOcen
GO

-- Procedura zwraca średnią ocen końcowych danego ucznia
CREATE PROC SredniaOcen (@Id_ucznia INT, @Srednia FLOAT OUTPUT)
AS
SET @Srednia = (SELECT AVG(OK.Ocena) FROM
(SELECT * FROM OcenyKoncowe(@Id_ucznia)) AS OK
)

SET @Srednia = ROUND(@Srednia, 2)

GO

--Przykładowe zapytanie
declare @xxx FLOAT
exec SredniaOcen 6, @xxx OUTPUT
SELECT @xxx
-----

IF OBJECT_ID('DBO.Usprawiedliwionych','P') IS NOT NULL
DROP PROC DBO.Usprawiedliwionych
GO

-- procedura zwraca jaki procent nieobecności z danej klasy jest usprawiedliwionych
CREATE PROC Usprawiedliwionych (@IdKlasy CHAR(2), @Result FLOAT OUTPUT)
AS

DECLARE @WszystkieNieob FLOAT
```

```

DECLARE @Usprawiedliwione FLOAT

SET @WszystkieNieob = (
SELECT COUNT(*)
FROM Nieobecności AS N JOIN Uczniowie AS U ON N.[Id Ucznia] = U.Id
WHERE U.Klasa = @IdKlasy)

SET @Usprawiedliwione = (
SELECT COUNT(*)
FROM Nieobecności AS N JOIN Uczniowie AS U ON N.[Id Ucznia] = U.Id
WHERE N.Usprawiedliwiona = 1 AND U.Klasa = @IdKlasy)

SET @Result = ROUND(@Usprawiedliwione/@WszystkieNieob, 2) * 100

GO

-- Przykładowe zapytanie
declare @xxx FLOAT
exec Usprawiedliwionych '1c' ,@xxx OUTPUT
SELECT @xxx
-----
IF OBJECT_ID('DBO.SredniaKlasy','P') IS NOT NULL
DROP PROC DBO.SredniaKlasy
GO

-- Procedura zwraca średnią zadanej klasy
CREATE PROC SredniaKlasy (@IdKlasy CHAR(2) ,@Srednia FLOAT OUTPUT)
AS

DECLARE @tmp INT
SET @tmp = (SELECT COUNT(*) FROM Uczniowie WHERE Klasa = @IdKlasy)

DECLARE @Suma FLOAT
SET @Suma = 0.0

DECLARE AktId CURSOR FOR
(SELECT Id
FROM Uczniowie
WHERE Klasa = @IdKlasy)

DECLARE @idUcznia INT
OPEN AktId
FETCH AktId INTO @idUcznia

WHILE @@FETCH_STATUS = 0
BEGIN

    SET @Suma = @Suma + (SELECT dbo.SredniaOcenFun(@idUcznia))
    FETCH AktId INTO @idUcznia

END

CLOSE AktId
DEALLOCATE AktId
SET @Srednia = ROUND(@Suma/@tmp,2)

GO

declare @xxx FLOAT
exec SredniaKlasy '1b' ,@xxx OUTPUT
SELECT @xxx

```

```

-----
IF OBJECT_ID('DBO.SredniaIloscNieobecnosci','P') IS NOT NULL
DROP PROC DBO.SredniaIloscNieobecnosci
GO
-- Procedura zwraca średnią ilość nieobecności "na łebka" z zadanej klasy
CREATE PROC SredniaIloscNieobecnosci (@IdKlasy CHAR(2) ,@Srednia FLOAT OUTPUT)
AS

DECLARE @Suma FLOAT
DECLARE @IleUczniow FLOAT
SET @Suma = 0.0

SET @IleUczniow = (SELECT COUNT(*) FROM Uczniowie WHERE Klasa = @IdKlasy)

SET @Suma = (SELECT COUNT(*) FROM
Nieobecności AS N JOIN Uczniowie AS U ON N.[Id Ucznia] = U.Id
WHERE U.Klasa = @IdKlasy)

SET @Srednia = ROUND(@Suma/@IleUczniow, 2)

GO

declare @xxx FLOAT
exec SredniaIloscNieobecnosci '1b' ,@xxx OUTPUT
SELECT @xxx

```

----- -- TRIGGERY -----

```

IF OBJECT_ID('Tr1','TR') IS NOT NULL
DROP TRIGGER Tr1
go

-- Sprawdza czy dany uczeń spełnia progi oraz czy klasa jest w stanie kogoś przyjąć
CREATE TRIGGER Tr1 ON Uczniowie
INSTEAD OF INSERT, UPDATE
AS

DECLARE @Naz NVARCHAR(50)
DECLARE @Im NVARCHAR(50)
DECLARE @Kl CHAR(2)
DECLARE @D DATE
DECLARE @WM INT
DECLARE @WB INT
DECLARE @WH INT

DECLARE @Progi Table (MAT INT, BIOL INT, HUM INT)
INSERT INTO @Progi VALUES
((SELECT MIN([Wynik Matematyka]) FROM Uczniowie WHERE Klasa = '1a'), (SELECT
MIN([Wynik Biologia]) FROM Uczniowie WHERE Klasa = '1b'), (SELECT MIN([Wynik
Humanistyczne]) FROM Uczniowie WHERE Klasa = '1c'))

DECLARE Kandydat CURSOR FOR SELECT Nazwisko, Imie, Klasa, [Data urodzenia], [Wynik
Matematyka], [Wynik Humanistyczne], [Wynik Humanistyczne] FROM inserted
OPEN Kandydat
FETCH NEXT FROM Kandydat INTO @Naz, @Im, @Kl, @D, @WM, @WB, @WH

```

```

WHILE @@FETCH_STATUS = 0
BEGIN

    DECLARE @IleUczniow INT
    SET @IleUczniow = (SELECT COUNT(*) FROM Uczniowie WHERE Klasa = @Kl)

    DECLARE @IleMaxUczniow INT
    SET @IleMaxUczniow = (SELECT [Limit osob] FROM Klasy WHERE Id = @Kl)

    IF (@IleUczniow >= @IleMaxUczniow)
    BEGIN
        ROLLBACK
        RAISERROR('Klasa %s ma już maksymalną ilość osób',16,1, @Kl)
    END

    IF (@Kl = '1a' AND @WM < (SELECT MAT FROM @Progi))
    BEGIN
        ROLLBACK
        RAISERROR('Kandydat %s %s nie spełnia progu na profil matematyczny',16,1,
@Naz, @Im)
    END
    ELSE IF (@Kl = '1b' AND @WB < (SELECT BIOL FROM @Progi))
    BEGIN
        ROLLBACK
        RAISERROR('Kandydat %s %s nie spełnia progu na profil biologiczny',16,1,
@Naz, @Im)
    END
    ELSE IF (@Kl = '1c' AND @WH < (SELECT HUM FROM @Progi))
    BEGIN
        ROLLBACK
        RAISERROR('Kandydat %s %s nie spełnia progu na profil
humanistyczny',16,1, @Naz, @Im)
    END
    INSERT INTO Uczniowie VALUES (@Naz, @Im,@Kl, @D, @WM, @WB, @WH )
    FETCH NEXT FROM Kandydat INTO @Naz, @Im, @Kl, @D, @WM, @WB, @WH
END

CLOSE Kandydat
DEALLOCATE Kandydat

GO

INSERT INTO Uczniowie VALUES
('Skywalker','Anakin', '1c', '1999-01-01', 10, 10, 90),
('Skywalker','Anakin', '1b', '1999-01-01', 91, 10, 31),
('Skywalker','Anakin', '1a', '1999-01-01', 10, 90, 31)
SELECT * FROM Uczniowie WHERE Klasa = '1c'

-----
IF OBJECT_ID('Tr2','TR') IS NOT NULL
DROP TRIGGER Tr2
go

-- sprawdza czy ocena jest z przedziału 1-6 oraz czy dnia danego datą mogły odbyć się
dane zajęcia
CREATE TRIGGER Tr2 ON Oceny
INSTEAD OF INSERT
AS

DECLARE @IdU INT
DECLARE @IdP NVARCHAR(10)
DECLARE @Oc INT
DECLARE @Dat Date

```

```
DECLARE @TO INT
```

```
DECLARE Grades CURSOR FOR SELECT [Id Ucznia], [ID Przedmiotu], Ocena, Data, [Typ  
oceny] FROM inserted  
OPEN Grades  
FETCH NEXT FROM Grades INTO @IdU, @IdP, @Oc, @Dat, @TO
```

```
WHILE @@FETCH_STATUS = 0  
BEGIN
```

```
    -- sprawdzamy czy ocena jest z odpowiedniego zakresu  
    IF @Oc > 6 OR @Oc < 1  
    BEGIN  
        ROLLBACK  
        RAISERROR('Ocena może być tylko z przedziału 1-6',16,1)  
    END
```

```
    -- pobieramy dzień tygodnia  
    DECLARE @DzienTygodnia INT  
    SET @DzienTygodnia = DATEPART(WEEKDAY, @Dat)
```

```
    DECLARE @DzienTygodniaChar NVARCHAR(12)
```

```
    IF @DzienTygodnia = 2  
        SET @DzienTygodniaChar = 'Poniedziałek'  
    ELSE IF @DzienTygodnia = 3  
        SET @DzienTygodniaChar = 'Wtorek'  
    ELSE IF @DzienTygodnia = 4  
        SET @DzienTygodniaChar = 'Środa'  
    ELSE IF @DzienTygodnia = 5  
        SET @DzienTygodniaChar = 'Czwartek'  
    ELSE IF @DzienTygodnia = 6  
        SET @DzienTygodniaChar = 'Piątek'  
    ELSE IF @DzienTygodnia = 7  
        SET @DzienTygodniaChar = 'Sobota'  
    ELSE  
        SET @DzienTygodniaChar = 'Niedziela'
```

```
    IF (NOT EXISTS (SELECT *  
FROM PlanZajęć AS PZ JOIN Uczniowie AS U ON PZ.[Klasa/Grupa] = U.Klasa  
WHERE PZ.[ID Przedmiotu] = @IdP AND PZ.Dzień = @DzienTygodniaChar AND U.Id =  
@IdU))  
    BEGIN  
        ROLLBACK  
        RAISERROR('Tego dnia nie ma takich zajęć o tej godzinie dla tej  
klasy',16,1)  
    END
```

```
    INSERT INTO Oceny VALUES(@IdU, @IdP, @Oc, @Dat, @TO)
```

```
    FETCH NEXT FROM Grades INTO @IdU, @IdP, @Oc, @Dat, @TO
```

```
END
```

```
CLOSE Grades  
DEALLOCATE Grades
```

```
GO
```

```
INSERT INTO Oceny VALUES  
(1, 'BIOL', 1, '2017-11-03', 1) -- 2017-11-03 to piątek, więc klasa 1b nie ma wtedy  
biologii
```

```
INSERT INTO Oceny VALUES
```

```
(1, 'JPOL', 1, '2017-11-03', 1) -- jednak język polski już ma
```

```
-----  
IF OBJECT_ID('Tr3','TR') IS NOT NULL  
DROP TRIGGER Tr3  
go
```

```
-- sprawdza czy danego dnia o danej godzinie lekcyjnie mogły się odbyć takie zajęcia  
CREATE TRIGGER Tr3 ON Nieobecności  
INSTEAD OF INSERT  
AS
```

```
DECLARE @IdU INT  
DECLARE @Dat Date  
DECLARE @GL INT  
DECLARE @IdZ NVARCHAR(10)  
DECLARE @U BIT
```

```
DECLARE NB CURSOR FOR SELECT [Id Ucznia], Data,[Godzina lekcyjna], [Id Zajęć],  
Usprawiedliwiona FROM inserted  
OPEN NB  
FETCH NEXT FROM NB INTO @IdU, @Dat, @GL, @IdZ, @U
```

```
WHILE @@FETCH_STATUS = 0  
BEGIN  
    IF (EXISTS (SELECT * FROM Nieobecności  
WHERE [Id Ucznia] = @IdU AND Data = @Dat AND [Godzina lekcyjna] = @GL AND [Id  
Zajęć] = @IdZ))  
        BEGIN  
            ROLLBACK  
            RAISERROR('Uczeń ma już wpisana wtedy nieobecność',16,1)  
        END
```

```
-- pobieramy dzień tygodnia  
DECLARE @DzienTygodnia INT  
SET @DzienTygodnia = DATEPART(WEEKDAY, @Dat)
```

```
DECLARE @DzienTygodniaChar NVARCHAR(12)
```

```
IF @DzienTygodnia = 2  
    SET @DzienTygodniaChar = 'Poniedziałek'  
ELSE IF @DzienTygodnia = 3  
    SET @DzienTygodniaChar = 'Wtorek'  
ELSE IF @DzienTygodnia = 4  
    SET @DzienTygodniaChar = 'Środa'  
ELSE IF @DzienTygodnia = 5  
    SET @DzienTygodniaChar = 'Czwartek'  
ELSE IF @DzienTygodnia = 6  
    SET @DzienTygodniaChar = 'Piątek'  
ELSE IF @DzienTygodnia = 7  
    SET @DzienTygodniaChar = 'Sobota'  
ELSE  
    SET @DzienTygodniaChar = 'Niedziela'
```

```
IF (NOT EXISTS (SELECT *  
FROM PlanZajęć AS PZ JOIN Uczniowie AS U ON PZ.[Klasa/Grupa] = U.Klasa  
WHERE PZ.[ID Przedmiotu] = @IdZ AND PZ.Dzień = @DzienTygodniaChar  
AND U.Id = @IdU AND PZ.[Godzina lekcyjna] = @GL  
))
```

```

        BEGIN
        ROLLBACK
        RAISERROR('Tego dnia o tej godzinie ten uczeń nie ma takich zajęć',16,1)
    END
    INSERT INTO Nieobecności VALUES (@IdU, @Dat, @GL, @IdZ, @U)
    FETCH NEXT FROM NB INTO @IdU, @Dat, @GL, @IdZ, @U
END

CLOSE NB
DEALLOCATE NB

GO

INSERT INTO Nieobecności VALUES
(1, '2017-11-16', 1, 'MAT', 0) -- 2017-09-28 to czwartek 1 lekcja, uczen z klasy 1b ma
wówczas matematykę

INSERT INTO Nieobecności VALUES
(1, '2017-11-16', 1, 'BIOL', 0)
-----
IF OBJECT_ID('Tr4','TR') IS NOT NULL
DROP TRIGGER Tr4
go

CREATE TRIGGER Tr4 ON PlanZajęć
INSTEAD OF INSERT
AS

DECLARE @K1 CHAR(2)
DECLARE @IdP NVARCHAR(10)
DECLARE @Dz NVARCHAR(12)
DECLARE @GL INT

DECLARE PZ CURSOR FOR SELECT "Klasa/Grupa", "ID Przedmiotu", "Dzień", [Godzina
lekcyjna] FROM inserted
OPEN PZ
FETCH NEXT FROM PZ INTO @K1, @IdP, @Dz, @GL

WHILE @@FETCH_STATUS = 0
BEGIN

    IF (@IdP IN ('PJŁ', 'JNIEM_b', 'JANG_b') AND @K1 NOT LIKE '%b')
    BEGIN
        ROLLBACK
        RAISERROR('Ten przedmiot przeznaczony jest dla klasy biologicznej',16,1)
    END
    ELSE IF (@IdP IN ('JNIEM_a', 'JANG_a', 'MAT_ROZ') AND @K1 NOT LIKE '%a')
    BEGIN
        ROLLBACK
        RAISERROR('Ten przedmiot przeznaczony jest dla klasy matematycznej',16,1)
    END
    ELSE IF (@IdP IN ('JNIEM_c', 'JANG_c', 'JPOL_ROZ') AND @K1 NOT LIKE '%c')
    BEGIN
        ROLLBACK
        RAISERROR('Ten przedmiot przeznaczony jest dla klasy
humanistycznej',16,1)
    END
    ELSE IF (@IdP = 'MAT' AND @K1 LIKE '%a')
    BEGIN
        ROLLBACK
        RAISERROR('Klasa o profilu matematycznym może mieć matematykę tylko w
wariantcie rozszerzonym ',16,1)
    END

```



```

ELSE IF (@IdP = 'JPOL' AND @Kl LIKE '%c')
BEGIN
    ROLLBACK
    RAISERROR('Klasa o profilu humanistycznym może mieć język polski tylko w
wariantcie rozszerzonym ',16,1)
END

```

```

IF (EXISTS (SELECT * FROM PlanZajęc
WHERE @Kl = [Klasa/Grupa] AND @Dz = Dzień AND @GL = [Godzina lekcyjna]))
BEGIN
    ROLLBACK
    RAISERROR('Klasa %s ma wówczas już jakieś zajęcia',16,1, @Kl)
END

```

```

IF (NOT EXISTS(SELECT * FROM Zajęcia WHERE Id = @IdP))
BEGIN
    ROLLBACK
    RAISERROR('Nie ma takiego przedmiotu',16,1)
END

```

```

INSERT INTO PlanZajęc VALUES (@Kl, @IdP, @Dz, @GL)

```

```

    FETCH NEXT FROM PZ INTO @Kl, @IdP, @Dz, @GL
END

```

```

CLOSE PZ
DEALLOCATE PZ

```

```

GO

```

```

INSERT INTO PlanZajęc VALUES
('1a', 'MAT', 'Poniedziałek',8)

```

```

INSERT INTO PlanZajęc VALUES
('1a', 'JPOL_ROZ', 'Poniedziałek',8)

```

```

INSERT INTO PlanZajęc VALUES
('1a', 'MAT_ROZ', 'Poniedziałek',8)

```

```

-----

IF OBJECT_ID('Tr5','TR') IS NOT NULL
DROP TRIGGER Tr5
go

```

```

CREATE TRIGGER Tr5 ON Nauczyciele
INSTEAD OF INSERT
AS

```

```

DECLARE @Naz NVARCHAR(50)
DECLARE @Im NVARCHAR(50)
DECLARE @PG NVARCHAR(50)
DECLARE @PB NVARCHAR(50)
DECLARE @PB2 NVARCHAR(50)

```

```

DECLARE Nauczyciel CURSOR FOR SELECT Nazwisko, Imie, [Przedmiot Główny], [Przedmiot
Poboczny], [Przedmiot Poboczny 2] FROM inserted
OPEN Nauczyciel

```

```

FETCH NEXT FROM Nauczyciel INTO @Naz, @Im, @PG, @PB, @PB2

WHILE @@FETCH_STATUS = 0
BEGIN

    IF (@PG = @PB OR @PB = @PB2 OR @PB2 = @PG)
    BEGIN
        ROLLBACK
        RAISERROR('W każdej kolumnie musi być inna wartość',16,1)
    END

    IF (NOT EXISTS (SELECT DISTINCT [Nazwa przedmiotu] FROM Zajęcia WHERE [Nazwa
przedmiotu] = @PG))
    BEGIN
        ROLLBACK
        RAISERROR('Nie ma takiego przedmiotu głównego jak %s',16,1, @PG)
    END

    IF (NOT EXISTS (SELECT DISTINCT [Nazwa przedmiotu] FROM Zajęcia WHERE [Nazwa
przedmiotu] = @PB))
    BEGIN
        ROLLBACK
        RAISERROR('Nie ma takiego przedmiotu pobocznego jak %s',16,1, @PB)
    END

    IF (NOT EXISTS (SELECT DISTINCT [Nazwa przedmiotu] FROM Zajęcia WHERE [Nazwa
przedmiotu] = @PB2))
    BEGIN
        ROLLBACK
        RAISERROR('Nie ma takiego przedmiotu pobocznego 2 jak %s',16,1, @PB2)
    END
    INSERT INTO Nauczyciele VALUES(@Naz, @Im, @PG, @PB, @PB2)
    FETCH NEXT FROM Nauczyciel INTO @Naz, @Im, @PG, @PB, @PB2
END

CLOSE Nauczyciel
DEALLOCATE Nauczyciel

GO

INSERT INTO Nauczyciele VALUES
('Atryda','Paul', 'Matematyka', 'Fizyka', 'Język polski')

INSERT INTO Nauczyciele VALUES
('Atryda','Paul', 'Fizyka', 'Fizyka', 'Język polski')

```

-- INDEKSY

```

CREATE INDEX Uczniowie_Index ON Uczniowie (Id, Nazwisko, Imie)

SELECT Id, Nazwisko, Imie FROM Uczniowie
WHERE Nazwisko LIKE '[A-G]%'

CREATE INDEX Oceny_INDEX ON Oceny (Id_Ucznia, [ID Przedmiotu]) INCLUDE (Ocena, Data,
[Typ Oceny])

SELECT Id_Ucznia, [ID Przedmiotu], Ocena, Data, [Typ oceny] FROM Oceny
WHERE Id_Ucznia = 1 AND [ID Przedmiotu] = 'BIOL'

```

-- DZIEDZICZENIE

---Table per class hierarchy

```
CREATE TABLE Uczniowie(  
  Id INT PRIMARY KEY IDENTITY(1,1),  
  Nazwisko NVARCHAR(50) NOT NULL,  
  Imie NVARCHAR(50) NOT NULL,  
  Klasa CHAR(2) NOT NULL FOREIGN KEY REFERENCES Klasy (Id),  
  [Data urodzenia] DATE,  
  [Wynik Matematyka] INT NOT NULL CHECK ([Wynik Matematyka] >= 0 AND [Wynik  
Matematyka] <= 100),  
  [Wynik Biologia] INT NOT NULL CHECK ([Wynik Biologia] >= 0 AND [Wynik Biologia] <=  
100),  
  [Wynik Humanistyczne] INT NOT NULL CHECK ([Wynik Humanistyczne] >= 0 AND [Wynik  
Humanistyczne] <= 100)  
  Stanowisko NVARCHAR(20) NULL  
)
```

INSERT INTO Uczniowie VALUES

```
('Skywalker','Anakin','1b','1999-01-01',91,90,31,NULL),  
('Kenobi','Obi-Wan','1b','1999-07-07',60,98,40,NULL),  
('Windu','Mace','1b','1999-01-26',80,90,60,NULL),  
('Ahsoka','Tano','1b','1999-02-13',51,85,33,NULL),  
('Fisto','Kit','1b','1999-12-29',60,87,40,NULL),  
('Kolar','Eeth','1b','1999-11-03',46,71,45,'Zastępcą'),  
('Tii','Shaak','1b','1999-04-12',50,71,31,NULL),  
('Vos','Quinlan','1b','1999-08-19',70,80,90,NULL),  
('Yoda','Master','1b','1999-04-12',82,100,70,NULL),  
('Mundi','Ki-Adi','1b','1999-09-21',46,78,60,NULL),  
('Koon','Plo','1b','1999-10-11',61,92,53,NULL),  
('Jinn','Qui-Gon','1b','1999-02-14',76,90,40,NULL),  
('Skywalker','Luke','1b','1999-05-04',82,95,60,NULL),  
('Solo','Han','1b','1999-09-25',50,71,78,NULL),  
('Amidala','Padme','1b','1999-08-13',60,92,97,NULL),  
('Calrissian','Lando','1b','1999-05-04',73,82,43,NULL),
```

```
('Palpatine','Sheev','1a','1999-12-12',99,60,60,NULL),  
('Tarkin','Wilhuff','1a','1999-05-23',88,45,80,NULL),  
('Binks','Jar-Jar','1a','1999-06-28',100,34,65,NULL),  
('Marek','Galen','1a','1999-02-11',88,68,37,NULL),  
('Revan','Mikołaj','1a','1999-04-12',70,54,93,NULL),  
('Wullf','Yularen','1a','1999-11-01',68,23,49,NULL),  
('Maul','Dariusz','1a','1999-12-01',88,42,84,NULL),  
('Ren','Kylo','1a','1999-04-16',91,74,48,NULL),  
('Serenno','Dooku','1a','1999-01-21',81,59,47,NULL),  
('Hut','Jabba','1a','1999-03-03',77,34,84,NULL),  
('Bane','Cad','1a','1999-12-09',72,55,40,'Skarbik'),  
('Plagaeus','Darian','1a','1999-11-18',71,57,92,NULL),  
('Grievous','Qymaen','1a','1999-11-21',79,34,48,NULL),  
('Opress','Savage','1a','1999-04-13',72,60,65,NULL),  
('Fett','Boba','1a','1999-04-13',78,80,67,NULL),
```

```
('Kirk','James','1c','1999-12-12',34,63,90,NULL),  
('McCoy','Leonard','1c','1999-06-28',45,34,70,NULL),
```

```
( 'Sulu','Hikaru', '1c', '1999-04-12', 43, 32, 68, NULL),
( 'Chekov','Pavel', '1c', '1999-11-01', 34, 44, 82, NULL),
( 'Uhura','Nyota', '1c', '1999-12-01', 75, 54, 91, NULL),
( 'Picard','Jean-Luc', '1c', '1999-04-16', 77, 71, 78, NULL),
( 'Riker','William', '1c', '1999-01-21', 44, 55, 74, 'Przewodniczący'),
( 'La Forge','Georgi', '1c', '1999-03-03', 56, 82, 88, NULL),
( 'Crusher','Beverly', '1c', '1999-12-09', 55, 49, 81, NULL),
( 'Troi','Deanna', '1c', '1999-11-18', 67, 54, 70, NULL),
( 'Yar','Tasha', '1c', '1999-11-21', 39, 94, 71, NULL),
( 'Crusher','Wesley', '1c', '1999-04-13', 66, 45, 72, NULL),
( 'Pulaski','Catherine', '1c', '1999-04-13', 70, 64, 89, NULL),
( 'Sisko','Benjamin', '1c', '1999-02-11', 72, 34, 68, NULL)
```

