

Baza danych szkoły

Alicja Kuk i Tomasz Miśkiewicz

1. Założenia projektu.

Projekt ma realizować podstawowe funkcje bazy danych do zarządzania szkołą.

Gromadzi ona **dane uczniów i nauczycieli**. Każda osoba dziedziczy po encji People, która nadaje hasło i identyfikator konta każdemu użytkownikowi bazy danych. W bazie zapisane są informacje dotyczące

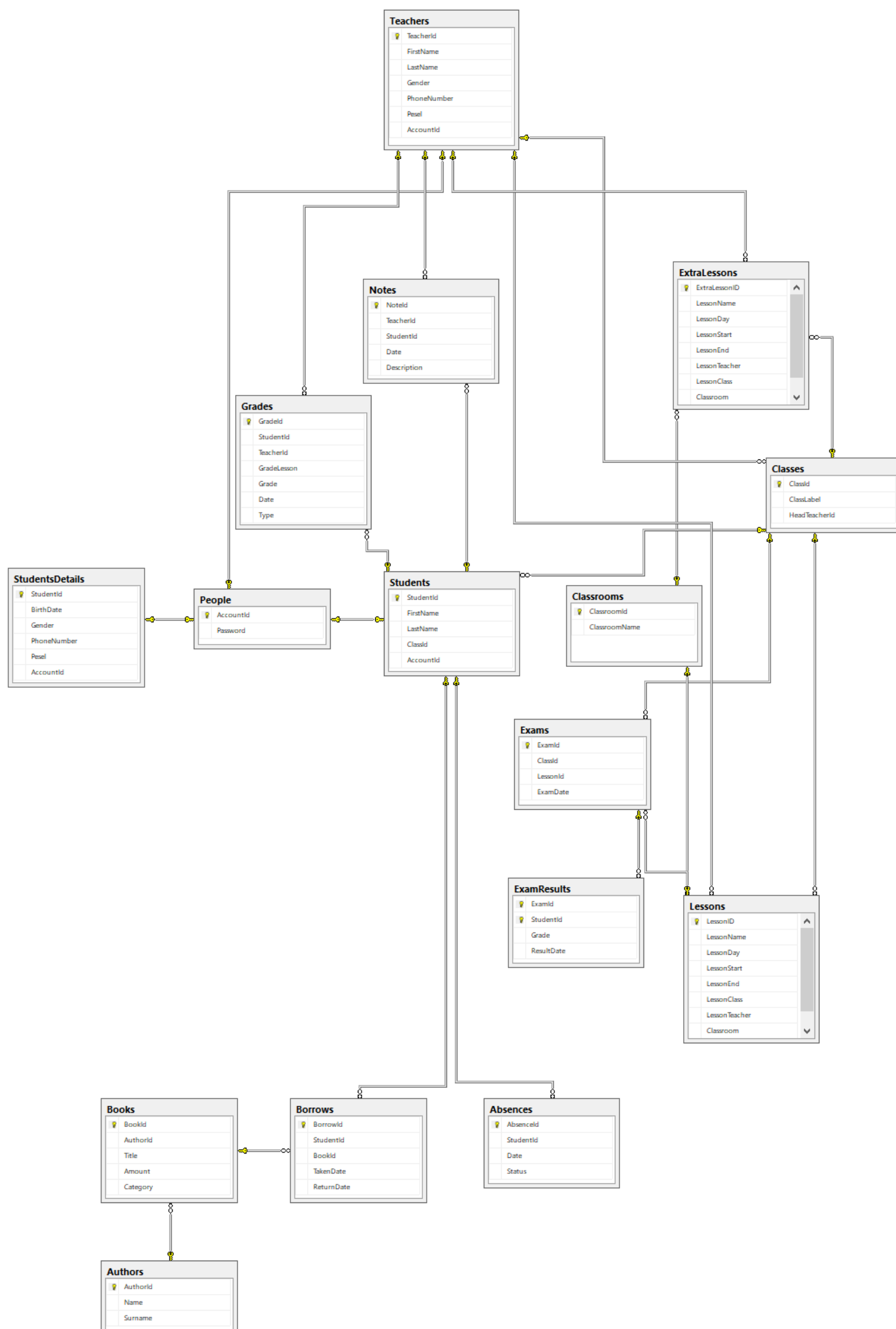
- **ocen, nieobecności i uwag** każdego z uczniów
- **poszczególnych klas**, możliwe jest przypisywanie egzaminów dla danych klas oraz uzupełnianie w systemie ich wyników dla poszczególnych uczniów
- **wszystkich lekcji, zajęć dodatkowych** organizowanych w szkole
- **pomieszczeń**, w których odbywają się lekcje.

W bazie danych znajduje się **system obsługi biblioteki szkolnej**, który umożliwia wypożyczanie oraz oddawanie książek. Śledzi on ilość dostępnych książek, a także rozlicza uczniów oddających książkę po terminie wpisując im uwagę.

Ograniczenia przyjęte przy projektowaniu bazy danych

- Istnieje tylko jeden uczeń o danym imieniu i nazwisku
- Każda książka ma tylko jednego autora

2. Schemat bazy danych



3. Dodatkowe więzy integralności

student_phone_valid - sprawdza poprawność wprowadzonego numeru telefonu ucznia

student_pesel_valid - sprawdza poprawność peselu ucznia

phone_valid - sprawdza poprawność numeru telefonu nauczyciela

pesel_valid - sprawdza poprawność peselu nauczyciela

password_length - sprawdza czy hasło ma długość co najmniej 5

4. Opis stworzonych widoków

4.1 Lista wszystkich danych uczniów szkoły

```
CREATE VIEW [Students Details] AS
SELECT S.StudentId, S.FirstName, S.LastName, S.ClassId, SD.BirthDate,SD.Gender,SD.PhoneNumber,SD.Pesel FROM Students S
JOIN StudentsDetails SD
ON S.AccountId = SD.AccountId
GO
```

4.2 Lista uczniów wraz z ich średnią ocen

```
CREATE VIEW [Students Average Grade] AS
SELECT S.FirstName, S.LastName, Round(AVG(Grade),2) AS [Average grade] FROM Students S
LEFT JOIN Grades G
ON S.StudentId = G.StudentId
GROUP BY S.FirstName, S.LastName
GO
```

4.3 Lista zagrożonych uczniów

```
CREATE VIEW FailingGradeStudents AS
SELECT S.StudentID, S.ClassID, AVG(G.Grade) GradeAverage FROM Students S
JOIN Grades G
ON S.StudentID = G.StudentID
GROUP BY S.StudentID, S.ClassID
HAVING AVG(G.Grade) < 2.0
GO
```

4.4 Lista uczniów ze świadectwem z paskiem

```
CREATE VIEW DiplomaWithHonoursStudents AS
SELECT S.StudentID, S.FirstName, S.LastName, S.ClassID, AVG(G.Grade) GradeAverage FROM Students S
JOIN Grades G
ON S.StudentID = G.StudentID
GROUP BY S.StudentID, S.FirstName, S.LastName, S.ClassID
HAVING AVG(G.Grade) >= 4.75
GO
```

4.5 Lista nauczycieli wraz z nauczonym przedmiotem

```
CREATE VIEW TeachersAndTheirSubjects AS
SELECT T.TeacherID, T.FirstName, T.LastName, L.LessonName, 'No' IsExtra FROM Teachers T
JOIN Lessons L
ON T.TeacherID = L.LessonTeacher
UNION
SELECT T.TeacherID, T.FirstName, T.LastName, EL.LessonName, 'Yes' IsExtra FROM Teachers T
JOIN ExtraLessons EL
ON T.TeacherID = EL.LessonTeacher
GO
```

5. Opis stworzonych funkcji

5.1 Funkcja zwracająca ilość godzin, jaką dany nauczyciel pracuje w ciągu tygodnia

```
CREATE FUNCTION get_teacher_hours(  
    @TeacherId INT  
)  
RETURNS INT  
AS  
BEGIN  
    DECLARE @suma1 INT  
    DECLARE @suma2 INT  
    SELECT @suma1 = COUNT(*) FROM Lessons  
    WHERE LessonTeacher = @TeacherId;  
    SELECT @suma2 = COUNT(*) FROM ExtraLessons  
    WHERE LessonTeacher = @TeacherId;  
    RETURN @suma1 + @suma2  
END  
GO
```

5.2 Funkcja zwracająca plan lekcji danej klasy w danym dniu

```
CREATE FUNCTION LessonsPlan (@ClassID int, @Day nvarchar(12))  
RETURNS @LessonsTable TABLE (  
    ClassID int,  
    LessonID int,  
    LessonName nvarchar(50),  
    LessonStart time(0),  
    LessonEnd time(0),  
    TeacherFirstName nvarchar(50),  
    TeacherLastName nvarchar(50)  
)  
AS  
BEGIN  
    INSERT INTO @LessonsTable  
    SELECT C.ClassID, L.LessonID, L.LessonName, L.LessonStart, L.LessonEnd, T.FirstName, T.LastName  
    FROM Classes C  
    JOIN Lessons L  
    ON C.ClassID = L.LessonClass  
    JOIN Teachers T  
    ON L.LessonTeacher = T.TeacherID  
    WHERE C.ClassId = @ClassID AND L.LessonDay = @Day  
    RETURN  
END  
GO
```

5.3 Funkcja zwracająca plan lekcji danego nauczyciela danego dnia

```
CREATE FUNCTION TeachersLessons (@TeacherID int, @Day nvarchar(12))  
RETURNS @LessonsTable TABLE (  
    TeacherID int,  
    FirstName nvarchar(50),  
    LastName nvarchar(50),  
    LessonName nvarchar(50),  
    LessonStart time(0),  
    LessonEnd time(0),  
    IsExtra nvarchar(3)  
)  
AS  
BEGIN  
    INSERT INTO @LessonsTable  
    SELECT T.TeacherID, T.FirstName, T.LastName, L.LessonName, L.LessonStart, L.LessonEnd, 'No'  
    FROM Teachers T  
    JOIN Lessons L  
    ON T.TeacherID = L.LessonTeacher  
    WHERE T.TeacherID = @TeacherID AND L.LessonDay = @Day  
    INSERT INTO @LessonsTable  
    SELECT T.TeacherID, T.FirstName, T.LastName, EL.LessonName, EL.LessonStart, EL.LessonEnd, 'Yes'  
    FROM Teachers T  
    JOIN ExtraLessons EL  
    ON T.TeacherID = EL.LessonTeacher  
    WHERE T.TeacherID = @TeacherID AND EL.LessonDay = @Day  
    RETURN  
END  
GO
```

5.4 Funkcja zwracająca nieusprawiedliwione nieobecności danego ucznia

```
CREATE FUNCTION UnauthorizedAbsences (@StudentID int)
RETURNS @LessonsTable TABLE (
    Date DATE NOT NULL
)
AS
BEGIN
    INSERT INTO @LessonsTable
    SELECT A.Date
    FROM Absences A
    WHERE A.StudentId = @StudentID AND A.Status = N'N'
    RETURN
END
GO
```

5.5 Funkcja zwracająca książki nieoddane przez danego ucznia

```
CREATE FUNCTION NonreturnedBooks (
    @StudentId INT
)
RETURNS TABLE
AS
RETURN
    SELECT S.FirstName, S.LastName, B.TakenDate, Books.Title
    FROM Students S
    JOIN Borrows B
    ON S.StudentId = B.StudentId
    JOIN Books
    ON B.BookId = Books.BookId
    WHERE @StudentId = S.StudentId AND B.ReturnDate IS NULL
GO
```

6. Opis stworzonych procedur

6.1 Procedura dodająca nieobecność

```
CREATE PROCEDURE AddAbsence @StudentID int, @Date date
AS
INSERT INTO Absences
VALUES (@StudentID, @Date, 'N')
GO
```

6.2 Procedura usprawiedliwiająca nieobecność

```
CREATE PROCEDURE JustifyAbsence @AccountId INT, @Password nvarchar(30), @AbsenceId INT
AS
IF EXISTS(SELECT AccountId FROM People WHERE AccountId = @AccountId AND Password = @Password) BEGIN
    UPDATE Absences
    SET Status = 'U'
    WHERE AbsenceId = @AbsenceId
END
GO
```

6.3 Procedura dodająca ocenę uczniowi

```
CREATE PROCEDURE AddGrade @StudentID int, @TeacherID int, @GradeLesson nvarchar(20), @Grade float, @Date date, @Type nvarchar(50)
AS
INSERT INTO Grades
VALUES(@StudentID, @TeacherID, @GradeLesson, @Grade, @Date, @Type)
GO
```

6.4 Procedura wypożyczająca książkę

Procedura wymaga podania loginu i hasła konta ucznia oraz tytułu wypożyczanej książki. Jeżeli na stanie jest dostępna odpowiednia ilość książek to rozpocznie się transakcja dodająca odpowiednie wypożyczenie oraz zmniejszająca aktualny stan liczby książek. W przeciwnym wypadku wypisze się komunikat o niewystarczającej liczbie książek.

```

CREATE PROCEDURE BorrowBook @AccountId int, @Password nvarchar(30), @Title nvarchar(30)
AS
    IF EXISTS(SELECT AccountId FROM People WHERE AccountId = @AccountId AND Password = @Password) BEGIN
        BEGIN TRANSACTION
        IF (SELECT Amount FROM Books WHERE Title = @Title) > 0 BEGIN
            INSERT INTO Borrows VALUES
                ((SELECT StudentId FROM Students S JOIN People A ON S.AccountId = A.AccountId WHERE A.AccountId = @AccountId),
                (SELECT BookId FROM Books WHERE Title = @Title),
                GETDATE(), NULL)
            UPDATE Books
                SET Amount = (SELECT Amount FROM Books WHERE Title = @Title) - 1
                WHERE Title = @Title
            END
        ELSE PRINT N'Niewystarczająca liczba książek w bibliotece o podanym tytule';
        COMMIT TRANSACTION
    END
    ELSE BEGIN
        PRINT N'Nie ma użytkownika o takim hasle/loginie';
    END
END

```

6.5 Procedura zwracająca książkę

Procedura wymaga podania identyfikatora oraz hasła ucznia zwracającego książkę oraz tytułu książki. W przypadku podania prawidłowych danych rozpoczyna się transakcja ustawiająca datę oddania książki oraz zwiększająca stan dostępności tej książki w bibliotece.

```

CREATE PROCEDURE return_book @AccountId INT, @Password nvarchar(30), @Title nvarchar(30)
AS
    IF EXISTS(SELECT AccountId FROM People WHERE AccountId = @AccountId AND Password = @Password) BEGIN
        BEGIN TRANSACTION
        UPDATE Borrows
            SET ReturnDate = GETDATE()
            FROM Borrows B
            JOIN Books Bo ON Bo.BookId = B.BookId
            JOIN Students S ON S.StudentId = B.StudentId
            JOIN People P ON S.AccountId = P.AccountId
            WHERE Bo.Title = @Title AND P.AccountId = @AccountId AND P.Password = @Password
        UPDATE Books
            SET Amount = (SELECT Amount FROM Books WHERE Title = @Title) + 1
            WHERE Title = @Title
        COMMIT TRANSACTION
    END
    ELSE BEGIN
        PRINT N'Nieprawidłowe dane';
    END
END

```

6.6 Procedura dodająca zajęcia dodatkowe

```

CREATE PROCEDURE AddExtraLessons @LessonName nvarchar(50), @Day nvarchar(9), @LessonStart time(7), @LessonEnd time(7), @LessonTeacher int, @Classroom int
AS
INSERT INTO ExtraLessons
VALUES (@LessonName, @Day, @LessonStart, @LessonEnd, @LessonTeacher, @Classroom)
GO

```

7. Opis stworzonych wyzwalaczy

7.1 Wyzwalacz wypisujący komunikat, gdy uczeń dostaje piątą uwagę

W przypadku kiedy liczba uwag danego ucznia wyniesie 5 wypisany zostanie komunikat.

```

CREATE TRIGGER FifthNote
ON Notes
AFTER INSERT
AS
    IF(SELECT COUNT(*) FROM Notes WHERE StudentId = (SELECT StudentId FROM INSERTED)) = 5
        BEGIN
            PRINT('Uczen dostal piata uwage.')
        END
GO

```

7.2 Wyzwalacz wypisujący komunikat, gdy uczeń staje się zagrożony z danego przedmiotu

Jeżeli zostaje wpisana ocena, która powoduje, że uczeń zaczyna być zagrożony z danego przedmiotu zostaje wypisany komunikat z dokładnymi informacjami.

```

CREATE TRIGGER FailingGrade ON Grades
AFTER INSERT
AS
    SELECT * FROM INSERTED
    DECLARE @TempStudentID int = (SELECT TOP 1 StudentID FROM INSERTED)
    DECLARE @FirstName nvarchar(50) = (SELECT FirstName FROM Students WHERE StudentID = @TempStudentID)
    DECLARE @LastName nvarchar(50) = (SELECT LastName FROM Students WHERE StudentID = @TempStudentID)
    DECLARE @TempTeacherID nvarchar(50) = (SELECT TOP 1 TeacherID FROM INSERTED)
    DECLARE @TempLessonName nvarchar(50) = (SELECT TOP 1 GradeLesson FROM INSERTED)
    IF (SELECT AVG(Grade) FROM Grades WHERE StudentId = @TempStudentID AND GradeLesson = @TempLessonName) < 2.0
    BEGIN
        PRINT('Uczniowi ' + @FirstName + ' ' + @LastName + ' (id studenta ' + CAST(@TempStudentID AS NVARCHAR(5)) +
            ') grozi zagrożenie z przedmiotu ' + @TempLessonName + '!')
    END
END

```

7.3 Wyzwalacz blokujący dodanie dodatkowych zajęć, gdy nauczyciel prowadzący jest wtedy zajęty lub klasa biorąca udział ma wtedy lekcje

W przypadku, gdy chcemy dodać zajęcia dodatkowe, ale nauczyciel prowadzący ma w tym czasie inne zajęcia lub ma miejsce sytuacja, gdy klasa mająca w nich uczestniczyć ma lekcje pokrywające się z godzinami zajęć, procedura dodania tych zajęć dodatkowych zostaje anulowana i zostaje wyświetlony odpowiedni komunikat o błędzie.

```

CREATE TRIGGER CancelAddingExtraLesson ON ExtraLessons
INSTEAD OF INSERT
AS
    DECLARE @LessonName nvarchar(50) = (SELECT LessonName FROM INSERTED)
    DECLARE @LessonClassroom int = (SELECT Classroom FROM INSERTED)
    DECLARE @LessonDay nvarchar(12) = (SELECT LessonDay FROM INSERTED)
    DECLARE @TeacherID int = (SELECT LessonTeacher FROM INSERTED)
    DECLARE @LessonClass int = (SELECT LessonClass FROM INSERTED)
    DECLARE @LessonStart time(0) = (SELECT LessonStart FROM INSERTED)
    DECLARE @LessonEnd time(0) = (SELECT LessonEnd FROM INSERTED)
    DECLARE @TeacherError int = (SELECT COUNT(*) FROM (SELECT * FROM Lessons WHERE LessonTeacher = @TeacherID
        AND ((LessonStart >= @LessonStart AND LessonEnd <= @LessonEnd))) T)
    DECLARE @TeacherExtraError int = (SELECT COUNT(*) FROM (SELECT * FROM ExtraLessons WHERE LessonTeacher = @TeacherID
        AND ((LessonStart >= @LessonStart AND LessonEnd <= @LessonEnd))) T)
    DECLARE @ClassError int = (SELECT COUNT(*) FROM (SELECT * FROM Lessons WHERE LessonClass = @LessonClass
        AND ((LessonStart >= @LessonStart AND LessonEnd <= @LessonEnd))) T)
    DECLARE @ClassExtraError int = (SELECT COUNT(*) FROM (SELECT * FROM ExtraLessons WHERE LessonClass = @LessonClass
        AND ((LessonStart >= @LessonStart AND LessonEnd <= @LessonEnd))) T)
    IF (@TeacherError + @TeacherExtraError > 0)
    BEGIN
        RAISERROR('Nie powiodło się dopisanie - nauczyciel ma już wtedy lekcje', 16,1)
    END
    ELSE IF (@ClassError + @ClassExtraError > 0)
    BEGIN
        RAISERROR('Nie powiodło się dopisanie - klasa ma już wtedy lekcje', 16,1)
    END
    ELSE
    BEGIN
        INSERT INTO ExtraLessons VALUES
            (@LessonName, @LessonDay, @LessonStart, @LessonEnd, @TeacherID, @LessonClass, @LessonClassroom)
    END
GO

```

7.4 Wyzwalacz zwracający komunikat, gdy nauczyciel dostaje więcej niż 10 lekcji do prowadzenia tygodniowo

Wyzwalacz pokazuje komunikat, gdy zostają dodane zajęcia, które sprawiają, że nauczyciel prowadzący ma ponad 10 godzin lekcyjnych w tygodniu.

```

CREATE TRIGGER TeacherOverloaded
ON Lessons
AFTER INSERT
AS
    IF(SELECT COUNT(*) FROM Lessons WHERE LessonTeacher = (SELECT LessonTeacher FROM INSERTED)) > 10
    BEGIN
        PRINT('Nauczyciel ma zbyt duzo godzin pracujacych.')
    END
GO

```

7.5 Wyzwalacz, który zwraca komunikat, gdy uczeń ma więcej niż 5 nieusprawiedliwionych nieobecności

Wyzwalacz wyświetla ostrzegawczy komunikat wraz ze szczegółowymi informacjami, gdy dany uczeń dostaje piątą nieusprawiedliwioną nieobecność.

```

CREATE TRIGGER TooManyAbsences ON Absences
AFTER INSERT
AS
    SELECT * FROM INSERTED
    DECLARE @TempStudentID int = (SELECT StudentID FROM INSERTED)
    DECLARE @FirstName nvarchar(50) = (SELECT FirstName FROM Students WHERE StudentID = @TempStudentID)
    DECLARE @LastName nvarchar(50) = (SELECT LastName FROM Students WHERE StudentID = @TempStudentID)
    IF (SELECT COUNT(*) FROM Absences A WHERE A.StudentId = @TempStudentID AND A.Status = N'N') >= 5
    BEGIN

```

```
PRINT('Uczen ' + @FirstName + ' ' + @LastName + ' (id studenta ' + CAST(@TempStudentID AS NVARCHAR(5)) +
') ma 5 lub wiecej nieusprawiedliwionych nieobecności!')
END
```

7.6 Wyzwalacz, który wpisuje uwagę, gdy uczeń nie zwrócił książki w terminie

W przypadku, gdy uczeń oddał książkę do biblioteki po przekroczeniu terminu 30 dni na zwrot zostaje wpisana mu odpowiednia uwaga.

```
CREATE TRIGGER OverdueBorrow
ON Borrows
AFTER INSERT, UPDATE
AS
    IF(SELECT DATEDIFF(day, ReturnDate, TakenDate) FROM INSERTED) > 30
        BEGIN
            INSERT INTO Notes VALUES
                (0,(SELECT StudentId FROM INSERTED),
                GETDATE(), 'Uczen nie oddal książki w terminie')
        END
GO
```

Strategia pielęgnacyjna

Raz w tygodniu w soboty w nocy odbywa się utworzenie pełnej zapasowej kopii bazy danych. Dodatkowo codziennie po południu odbywa się zapisanie różnicowej kopii zapasowej.

```
CREATE PROCEDURE backup_database
AS
BACKUP DATABASE Szkoła
TO DISK = 'C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS\MSSQL\Backup\Szkoła.bak'
GO
```

```
CREATE PROCEDURE restore_database
AS
EXEC disconnect
RESTORE DATABASE Szkoła
FROM DISK = 'C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS\MSSQL\Backup\Szkoła.bak'
WITH REPLACE
GO
```

Typowe zapytania

```
SELECT * FROM Students

SELECT * FROM Books WHERE Amount > 0

SELECT * FROM ExamResults E
JOIN Students S ON E.StudentId = S.StudentId
WHERE Grade = 1

SELECT * FROM Teachers WHERE EXISTS(SELECT * FROM CLASSES WHERE HeadTeacherId = TeacherId)

SELECT ClassLabel, FirstName, LastName FROM Students S
JOIN Classes C ON C.ClassLabel = S.ClassId
ORDER BY ClassLabel

SELECT * FROM Exams
```

Skrypt tworzący tabele

```
USE SchoolDB;

IF OBJECT_ID('Absences', 'U') IS NOT NULL DROP TABLE Absences
IF OBJECT_ID('Borrows', 'U') IS NOT NULL DROP TABLE Borrows
IF OBJECT_ID('Notes', 'U') IS NOT NULL DROP TABLE Notes
IF OBJECT_ID('Books', 'U') IS NOT NULL DROP TABLE Books
IF OBJECT_ID('Authors', 'U') IS NOT NULL DROP TABLE Authors
IF OBJECT_ID('Grades', 'U') IS NOT NULL DROP TABLE Grades
IF OBJECT_ID('ExamResults', 'U') IS NOT NULL DROP TABLE ExamResults
```



```

IF OBJECT_ID('Exams', 'U') IS NOT NULL DROP TABLE Exams
IF OBJECT_ID('ExtraLessons', 'U') IS NOT NULL DROP TABLE ExtraLessons
IF OBJECT_ID('Lessons', 'U') IS NOT NULL DROP TABLE Lessons
IF OBJECT_ID('Classrooms', 'U') IS NOT NULL DROP TABLE Classrooms
IF OBJECT_ID('Students', 'U') IS NOT NULL DROP TABLE Students
IF OBJECT_ID('StudentsDetails', 'U') IS NOT NULL DROP TABLE StudentsDetails
IF OBJECT_ID('Classes', 'U') IS NOT NULL DROP TABLE Classes
IF OBJECT_ID('Teachers', 'U') IS NOT NULL DROP TABLE Teachers
IF OBJECT_ID('People', 'U') IS NOT NULL DROP TABLE People
GO

CREATE TABLE People (
    AccountId INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    Password NVARCHAR(50) NOT NULL,
    CONSTRAINT password_length CHECK(LEN>Password) >= 5)
)
GO

CREATE TABLE Teachers (
    TeacherId INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    FirstName NVARCHAR(50) NOT NULL,
    LastName NVARCHAR(50) NOT NULL,
    Gender NVARCHAR(1) NOT NULL,
    PhoneNumber VARCHAR(9) NOT NULL,
    Pesel VARCHAR(11) NOT NULL UNIQUE,
    AccountId INTEGER NOT NULL,
    CONSTRAINT TeachersFK FOREIGN KEY(AccountId) REFERENCES People(AccountID)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    CONSTRAINT phone_valid CHECK(ISNUMERIC(PhoneNumber) = 1 AND LEN(PhoneNumber) = 9),
    CONSTRAINT pesel_valid CHECK(ISNUMERIC(Pesel) = 1 AND LEN(Pesel) = 11)
)
GO

CREATE TABLE Classes (
    ClassId INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    ClassLabel NVARCHAR(2) NOT NULL,
    HeadTeacherId INTEGER NOT NULL,
    CONSTRAINT HeadTeacherFK FOREIGN KEY (HeadTeacherId) REFERENCES Teachers(TeacherId)
)
GO

CREATE TABLE Students(
    StudentId INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    FirstName NVARCHAR(50) NOT NULL,
    LastName NVARCHAR(50) NOT NULL,
    ClassId INTEGER NOT NULL,
    AccountId INTEGER NOT NULL,
    CONSTRAINT StudentsAcFK FOREIGN KEY(AccountId) REFERENCES People(AccountId)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    CONSTRAINT StudentClassFK FOREIGN KEY(ClassId) REFERENCES Classes(ClassId)
)
GO

CREATE TABLE StudentsDetails(
    StudentId INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    BirthDate DATE NOT NULL,
    Gender NVARCHAR(1) NOT NULL,
    PhoneNumber VARCHAR(9) NOT NULL,
    Pesel VARCHAR(11) NOT NULL UNIQUE,
    AccountId Integer NOT NULL,
    CONSTRAINT StudentsFK FOREIGN KEY(AccountId) REFERENCES People(AccountId)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    CONSTRAINT student_phone_valid CHECK(ISNUMERIC(PhoneNumber) = 1 AND LEN(PhoneNumber) = 9),
    CONSTRAINT student_pesel_valid CHECK(ISNUMERIC(Pesel) = 1 AND LEN(Pesel) = 11)
)
GO

CREATE TABLE Classrooms (
    ClassroomId INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    ClassroomName NVARCHAR(50) NOT NULL
)
GO

CREATE TABLE Lessons (
    LessonID INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    LessonName NVARCHAR(50) NOT NULL,
    LessonDay NVARCHAR(20),
    LessonStart TIME NOT NULL,
    LessonEnd TIME NOT NULL,
    LessonClass INT NOT NULL,
    LessonTeacher INTEGER NOT NULL,
    Classroom INTEGER NOT NULL,
    CONSTRAINT ClassFK FOREIGN KEY(LessonClass) REFERENCES Classes(ClassId),
    CONSTRAINT TeacherFK FOREIGN KEY(LessonTeacher) REFERENCES Teachers(TeacherId),
    CONSTRAINT ClasroomFK FOREIGN KEY(Classroom) REFERENCES Classrooms(ClassroomID)
)
GO

CREATE TABLE ExtraLessons (
    ExtraLessonID INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    LessonName NVARCHAR(50) NOT NULL,
    LessonDay NVARCHAR(20),
    LessonStart TIME NOT NULL,
    LessonEnd TIME NOT NULL,
    LessonTeacher INTEGER NOT NULL,

```

```

        LessonClass INT NOT NULL,
        Classroom INTEGER NOT NULL,
        CONSTRAINT ClasseFK FOREIGN KEY(LessonClass) REFERENCES Classes(ClassId),
        CONSTRAINT TeacherEFK FOREIGN KEY(LessonTeacher) REFERENCES Teachers(TeacherId),
        CONSTRAINT ClassroomEFK FOREIGN KEY(Classroom) REFERENCES Classrooms(ClassroomId)
    )
GO

CREATE TABLE Grades(
    GradeId INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    StudentId INTEGER NOT NULL,
    TeacherId INTEGER NOT NULL,
    GradeLesson NVARCHAR(20) NOT NULL,
    Grade FLOAT NOT NULL,
    Date DATE NOT NULL,
    Type NVARCHAR(50),
    CONSTRAINT StudentGradeFK FOREIGN KEY(StudentId) REFERENCES Students(StudentId),
    CONSTRAINT TeacherGradeFK FOREIGN KEY(TeacherId) REFERENCES Teachers(TeacherId),
    CONSTRAINT grade_valid CHECK(Grade BETWEEN 1 AND 6)
)
GO

CREATE TABLE Absences (
    AbsenceId INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    StudentId INTEGER NOT NULL,
    Date DATE NOT NULL,
    Status NVARCHAR(1) NOT NULL,
    CONSTRAINT AbsentStudentFK FOREIGN KEY(StudentId) REFERENCES Students(StudentId)
    ON UPDATE CASCADE
    ON DELETE CASCADE
)
GO

CREATE TABLE Notes(
    NoteId INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    TeacherId INTEGER NOT NULL,
    StudentId INTEGER NOT NULL,
    Date DATE NOT NULL,
    Description NVARCHAR(150) NOT NULL,
    CONSTRAINT NotingTeacherId FOREIGN KEY(TeacherId) REFERENCES Teachers(TeacherId),
    CONSTRAINT NotedStudentId FOREIGN KEY(StudentId) REFERENCES Students(StudentId)
)
GO

CREATE TABLE Exams (
    ExamId INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    ClassId INTEGER NOT NULL,
    LessonId INTEGER NOT NULL,
    ExamDate DATE NOT NULL,
    CONSTRAINT ExamsFK FOREIGN KEY(ClassId) REFERENCES Classes(ClassId),
    CONSTRAINT CourseFK FOREIGN KEY(LessonId) REFERENCES Lessons(LessonId)
)
GO

CREATE TABLE ExamResults (
    ExamId INTEGER NOT NULL,
    StudentId INTEGER NOT NULL,
    Grade FLOAT NOT NULL,
    ResultDate DATE NOT NULL,
    CONSTRAINT ExamResultsPK PRIMARY KEY(ExamId, StudentId),
    CONSTRAINT ExamsResultsFK FOREIGN KEY(ExamId) REFERENCES Exams(ExamId),
    CONSTRAINT exam_grade_valid CHECK(Grade BETWEEN 1 AND 6)
)
GO

CREATE TABLE Authors (
    AuthorId INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    Name NVARCHAR(50) NOT NULL,
    Surname NVARCHAR(50) NOT NULL,
)
GO

CREATE TABLE Books (
    BookId INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    AuthorId INTEGER NOT NULL,
    Title NVARCHAR(50) NOT NULL,
    Amount INTEGER NOT NULL,
    Category NVARCHAR(50) NOT NULL,
    CONSTRAINT AuthorFK FOREIGN KEY(AuthorId) REFERENCES Authors(AuthorId)
)
GO

CREATE TABLE Borrows (
    BorrowId INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    StudentId INTEGER NOT NULL,
    BookId INTEGER NOT NULL,
    TakenDate DATE NOT NULL,
    ReturnDate DATE,
    CONSTRAINT BookFK FOREIGN KEY(BookId) REFERENCES Books(BookId),
    CONSTRAINT StudentBorrowsFK FOREIGN KEY(StudentId) REFERENCES Students(StudentId)
)

```

Skrypt wpisujący dane do tabeli

```

INSERT INTO People VALUES
('E7Po]Y'),
('KK4?a: '),
('M5P?e; '),
('c<l2uM'),
(' [F2qz0'),
('TN73n{)'),
('c|X.$y'),
('V=j&h'),
('5d*0Nf'),
('hfI>?ytc'),
('Y[&mz?q?'),
('zaS#Nan'),
('>{-0w2"N'),
('b<eYSS?'),
('?pFy1Jrj'),
('Y02;juho'),
('fpX?n'),
('<`kk)7'),
('EEz0x}'),
('q70(7!>g'),
('I$arx, '),
('ud@F%vO<'),
('<FzI9i '),
('k;(2^G4t'),
('6Uop{o>D'),
('IEAkab[f'),
('0eIXnn)&'),
('>sIGGFcC'),
('tB]aDx1M'),
('p!Sj8'),
('oTD-RbU'),
('CIy!@|r%'),
('u7v_>'),
('d9[M`Fjz'),
('EN{vTj}|'),
('H,^soQb'),
('},pk2'),
('bo#Yxx'),
('kr1{ }0'),
('VEPGB7!+'),
('PM`6#.Q6'),
(':',_}LH'),
('5jeZs'),
('8Ci}mtpn'),
('>bMzn6&'),
('&\\}v]u1{'),
('VDIz8&6i'),
(']xAx7'),
('L2`)u'),
('a_9_s. %'),
('1BpjM3'),
('fF)M= '),
('zn_aXe3F'),
('R20J#'),
('K1VNPq?2'),
('f"$o51Q#'),
('4F3T"+H8'),
(' ,tZ[Z'),
(']IIL0'),
('WbN)H'),
('}e. %>'),
('R2>t/'),
('08,aR. '),
('SItk[ lC'),
('HSnxjx'),
('kNnhF'),
('+ ,F@%o;X'),
('Gnvp\#'),
('gFmLT'),
('][tw5<M['),
('0qAbIPeS'),
('')0G]sT'),
('Lx,0*Tt'),
('1>_L\'),
('-hdKk%q'),
('%fZa#'),
(' [uk:DCMz'),
('3^vYlKo'),
('z<:"\nj)'),
(' :Wa>>L'),
('R@BhiQ+v'),
('%_x8bG,p'),
('QHvkNRh&'),
('{C9^"!5'),
('sz^lsTo'),
('X}=85&Mf'),
('Kiw(@sb-'),
('u60}no'),
('D-[-v'),
('||Q$BI#'),
('A!m{5'),
('? "i\y'),
('d`uT[-'),

```

```
('d2Y"&A'),
('_',l7)1P-Z'),
('F9U7{5n'),
('v*F]j'),
('NGlke}'),
('u1M<@N=v'),
('9`T.A0'),
('Cv4Q8'),
('Wa7NB'),
('%vujWw>0'),
('\W#KL'),
('c[]vx'),
('|oy[+'),
('>w>FRsR'),
('*I%vfJ6'),
('&X@07r-'),
('/ky7v{+*'),
('Fz7+3S'),
('W.M8;'),
('FHH,x'),
('E<-P'),
('KUx#U'),
('TvHm{M2'),
('wc4Q0Kq'),
('*JkP8'),
('v*"E7E2'),
('bw{1l$'),
('"FP6`#'),
('M>tNce,'),
('ueQ;@7?'),
('"LRGq@E'),
('[a%h8N'),
('cFin(t\S'),
('nXoEyH'),
('^[7QL$'),
('dE@GE`?*'),
('G>2K"TS'),
('cfuD$*Ff'),
('i?P<RW"'),
('I3?>e'),
('%C0]t^Q_'),
(',xRz>dP'),
('1@`9{T'),
('wE"bn'),
('NCiI2vH'),
('y#tf_'),
('2<rg^T9p'),
('e6zXe?'),
(']G`iR'),
('-',-uE:_'),
('|@M<G&Ei'),
('0`0VU#'),
('|?7i>'),
('k?;%bdAL'),
('ixv[VL?'),
('Pl`1hgN'),
('bZH5a"')

INSERT INTO Teachers VALUES
('Zdobysław', 'Długosz', 'M', '469082971', '59527842732', 1),
('Witalis', 'Niewiadomski', 'M', '940484262', '00385707575', 2),
('Teresa', 'Siudak', 'K', '832711338', '04370534272', 3),
('Romana', 'Cacko', 'K', '981601972', '18313090026', 4),
('Sydonia', 'Gnieciecka-Krysztofowicz', 'K', '115072557', '04112499634', 5),
('Waldemar', 'Jastrzębski', 'M', '438936937', '20195796046', 6),
('Hanna', 'Świderek', 'K', '805594880', '99886202156', 7),
('Wit', 'Kadziłłowski', 'M', '609044297', '45107807899', 8),
('Jarosława', 'Caspaeri-Dąbska', 'K', '572662066', '39080563819', 9),
('Gerarda', 'Bauć', 'K', '008749737', '17223382462', 10),
('Maksym', 'Czepkowski', 'M', '625063396', '62795644816', 11),
('Magda', 'Chojecka', 'K', '323032632', '61026802915', 12),
('Sergiusz', 'Siciarek', 'M', '168681867', '97271342714', 13),
('Eliza', 'Nowosielska', 'K', '460575248', '34425160686', 14),
('Adam', 'Stopiński', 'M', '731350655', '27059124454', 15),
('Felicyta', 'Kuś-Czech', 'K', '993721353', '72795622137', 16),
('Ferdynand', 'Dobrowolski', 'M', '688191188', '88671032143', 17),
('Wojciecha', 'Józefowicz-Paszewska', 'K', '329498694', '88005165392', 18),
('Marcin', 'Mikołajczyk', 'M', '722443432', '17936547001', 19),
('Pankracy', 'Chroński', 'M', '110322920', '86281817351', 20),
('Kazimierz', 'Dylak', 'M', '895563907', '45480933296', 21),
('Anastazja', 'Golcew', 'K', '917371349', '01822913281', 22),
('Radosław', 'Józefczuk', 'M', '638881035', '46119789525', 23),
('Madlena', 'Furmańczyk', 'K', '255323514', '22045733716', 24),
('Symeon', 'Łysakowski', 'M', '899780960', '75545466892', 25),
('Kacper', 'Stańczak', 'M', '595259900', '92589120920', 26),
('Orfeusz', 'Kusak', 'M', '839578320', '83978719122', 27),
('Jarogniew', 'Bartosia', 'M', '476562533', '34094938447', 28),
('Tatiana', 'Mech', 'K', '479505031', '42208522620', 29),
('Wera', 'Kaczorowska', 'K', '981344014', '67889765616', 30)

INSERT INTO Classes VALUES
('1A', 20),
('1B', 9),
('2A', 1),
('2B', 25),
('3A', 7),
('3B', 14)
```

```
INSERT INTO Students VALUES
('Godfryd', 'Gudel', 1, 31),
('Genowefa', 'Kwacz', 1, 32),
('Ireneusz', 'Warach', 1, 33),
('Anastazy', 'Sokorski', 1, 34),
('Witolda', 'Czepakowska', 1, 35),
('Modest', 'Kusz', 1, 36),
('Felicyta', 'Suszyńska', 1, 37),
('Apolinary', 'Samborski', 1, 38),
('Hubert', 'Głogowski', 1, 39),
('Luiza', 'Młodzińska', 1, 40),
('Remigiusz', 'Broniszewski', 1, 41),
('Witolda', 'Fert', 1, 42),
('Irwin', 'Tarasiewicz', 1, 43),
('Ada', 'Herod', 1, 44),
('Oktawia', 'Muszyńska', 1, 45),
('Olgierd', 'Szperl', 1, 46),
('Gertruda', 'Muciak', 1, 47),
('Ireneusz', 'Czepułowski', 1, 48),
('Szczepan', 'Sałagan', 1, 49),
('Aga', 'Kicun', 1, 50),
('Kleopatra', 'Kulik', 2, 51),
('Brunon', 'Cendrowski', 2, 52),
('Robert', 'Banach', 2, 53),
('Dianna', 'Czepułowska', 2, 54),
('Bronisława', 'Kwacz', 2, 55),
('Kacper', 'Lewicka', 2, 56),
('Eliza', 'Marszałek', 2, 57),
('Sobiesława', 'Cuper', 2, 58),
('Emiliana', 'Pura', 2, 59),
('Mikołaj', 'Alchimionek', 2, 60),
('Ildefons', 'Kutzmann', 2, 61),
('Hubert', 'Fennig', 2, 62),
('Felicja', 'Dmochowska', 2, 63),
('Wasył', 'Urban', 2, 64),
('Marcjusz', 'Chojniak', 2, 65),
('Wincenty', 'Fereniec', 2, 66),
('Leon', 'Reiter', 2, 67),
('Otto', 'Muszyński', 2, 68),
('Elwira', 'Stanik', 2, 69),
('Makary', 'Iwanicki', 2, 70),
('Franciszka', 'Duda', 3, 71),
('Aldona', 'Szemetyłło', 3, 72),
('Agata', 'Wach', 3, 73),
('Edyta', 'Kust', 3, 74),
('Roderyk', 'Słowiński', 3, 75),
('Wodzisław', 'Pidek', 3, 76),
('Melania', 'Baczuk', 3, 77),
('Radomił', 'Osica', 3, 78),
('Albert', 'Stańczak', 3, 79),
('Beniamin', 'Nachulewicz', 3, 80),
('Franciszek', 'Krzyżewski', 3, 81),
('Adam', 'Łepkowski', 3, 82),
('Przemysław', 'Papiernik', 3, 83),
('Gwidon', 'Caban', 3, 84),
('Lesława', 'Madejek', 3, 85),
('Zuzanna', 'Korycińska', 3, 86),
('Kornelia', 'Noga', 3, 87),
('Wojciech', 'Terlecki', 3, 88),
('Józefa', 'Piechowska', 3, 89),
('Melchior', 'Piechocki', 3, 90),
('Jagoda', 'Górczak', 4, 91),
('Anastazy', 'Obara', 4, 92),
('Daria', 'Olkowski', 4, 93),
('Mirabella', 'Sepioło', 4, 94),
('Antonina', 'Kiełczawa', 4, 95),
('Krystian', 'Kochański', 4, 96),
('Elwira', 'Bastrzyk', 4, 97),
('Werner', 'Kłyckiński', 4, 98),
('Emil', 'Dziubak', 4, 99),
('Ambroży', 'Rejman', 4, 100),
('Unisław', 'Pająk', 4, 101),
('Emilia', 'Dzidowska', 4, 102),
('Aleksandra', 'Jabłonowska', 4, 103),
('Rudolfina', 'Gojniczek', 4, 104),
('Arleta', 'Jędrzejczak', 4, 105),
('Robert', 'Korycińska', 4, 106),
('Wanda', 'Skarżyńska', 4, 107),
('Emanuela', 'Kłuziak', 4, 108),
('Janina', 'Siudak', 4, 109),
('Gertruda', 'Staniszewska', 4, 110),
('Wanda', 'Warfołomiejew', 5, 111),
('Eustachy', 'Gromadzińska', 5, 112),
('Modest', 'Ledwożyw', 5, 113),
('Franciszek', 'Ślusarski', 5, 114),
('Oliwier', 'Sałuda', 5, 115),
('Judyta', 'Kieszkowska', 5, 116),
('Unisław', 'Machała', 5, 117),
('Emil', 'Musiał', 5, 118),
('Wilhelm', 'Gendis', 5, 119),
('Walerian', 'Szulczyk', 5, 120),
('Maria', 'Wardęga', 5, 121),
('Ksenia', 'Piskorska', 5, 122),
('Eulalia', 'Górny', 5, 123),
('Zdzisław', 'Urbaniak', 5, 124),
('Rajmunda', 'Bartnik', 5, 125),
('Władysława', 'Kusak', 5, 126),
('Beata', 'Smolarczyk', 5, 127),
```

('Eleonora', 'Kimak', 5, 128),
('Bożena', 'Jaske', 5, 129),
('Żanna', 'Józefowicz', 5, 130),
('Zygmunt', 'Romanowski', 6, 131),
('Ilona', 'Korolczuk', 6, 132),
('Irwin', 'Kijowski', 6, 133),
('Marcela', 'Cabaj', 6, 134),
('Leonia', 'Kuznicka', 6, 135),
('Gustawa', 'Karasińska', 6, 136),
('Bogumiła', 'Koniarska', 6, 137),
('Metody', 'Świdrak', 6, 138),
('Luba', 'Brudny', 6, 139),
('Mirella', 'Kujawa', 6, 140),
('Malina', 'Górka', 6, 141),
('Damian', 'Dębska', 6, 142),
('Ireneusz', 'Piątkowski', 6, 143),
('Bogdan', 'Kochański', 6, 144),
('Konstantyn', 'Czepik', 6, 145),
('Krystyna', 'Smardzewska', 6, 146),
('Luiza', 'Moczulska', 6, 147),
('Rafaela', 'Bator', 6, 148),
('Elwira', 'Szwarnowiecka', 6, 149),
('Mira', 'Trela', 6, 150)

INSERT INTO StudentsDetails VALUES
('2005-04-27', 'M', '115361388', '84348873154', 31),
('2005-08-08', 'K', '168237754', '67238353558', 32),
('2005-04-15', 'M', '745324757', '17686572638', 33),
('2005-06-10', 'M', '648825183', '43555855423', 34),
('2005-05-21', 'K', '178547263', '26541416147', 35),
('2005-09-18', 'M', '772713321', '52131741531', 36),
('2005-03-24', 'K', '316187422', '62635766568', 37),
('2005-10-09', 'M', '384436366', '58887872583', 38),
('2005-05-20', 'M', '857812175', '42723612332', 39),
('2005-10-18', 'K', '714287318', '51671312335', 40),
('2005-07-03', 'M', '886233446', '78573547574', 41),
('2005-08-03', 'K', '382486266', '26716143857', 42),
('2005-11-27', 'M', '181511833', '14713166351', 43),
('2005-04-13', 'K', '338568475', '61438271561', 44),
('2005-04-10', 'K', '854365663', '68412445584', 45),
('2005-10-18', 'M', '227723177', '73667428753', 46),
('2005-08-18', 'K', '735882157', '16886618713', 47),
('2005-08-12', 'M', '421737138', '35653676882', 48),
('2005-06-04', 'M', '435568255', '84487271465', 49),
('2005-10-01', 'K', '473138583', '38517551624', 50),
('2005-03-01', 'K', '475656558', '32742768857', 51),
('2005-04-22', 'M', '412851158', '72614186346', 52),
('2005-02-05', 'M', '814756745', '14486328416', 53),
('2005-04-24', 'K', '131564387', '86546847571', 54),
('2005-11-06', 'K', '184363276', '18776381155', 55),
('2005-09-25', 'M', '684827677', '24774378863', 56),
('2005-03-07', 'K', '176663588', '85537132671', 57),
('2005-10-20', 'K', '486834526', '38566377125', 58),
('2005-06-25', 'K', '774116636', '43416445663', 59),
('2005-05-09', 'M', '242731326', '17542534357', 60),
('2005-04-22', 'M', '664223787', '61168131355', 61),
('2005-10-11', 'M', '417685727', '23627753847', 62),
('2005-05-16', 'K', '256765642', '74472631432', 63),
('2005-01-14', 'M', '464256333', '47456441632', 64),
('2005-07-16', 'M', '536174376', '63613353132', 65),
('2005-11-27', 'M', '687712734', '58631532132', 66),
('2005-04-19', 'M', '153257217', '25624425437', 67),
('2005-07-24', 'M', '673336568', '85445787576', 68),
('2005-03-18', 'K', '848517732', '36284534161', 69),
('2005-06-24', 'M', '285763235', '23363637657', 70),
('2004-04-02', 'K', '234424452', '15735717544', 71),
('2004-03-12', 'K', '456345477', '77243836235', 72),
('2004-11-04', 'K', '348745885', '83285177513', 73),
('2004-07-27', 'K', '737136827', '57662368681', 74),
('2004-09-12', 'M', '623553745', '16111256642', 75),
('2004-09-07', 'M', '651721142', '26727843123', 76),
('2004-08-23', 'K', '572532678', '12812347543', 77),
('2004-06-07', 'M', '122632317', '62132246433', 78),
('2004-04-17', 'M', '782326888', '63546222557', 79),
('2004-09-14', 'M', '536123667', '17535848537', 80),
('2004-10-17', 'M', '733366744', '22672884527', 81),
('2004-09-27', 'M', '671778313', '17262212751', 82),
('2004-06-10', 'M', '341286733', '74616855385', 83),
('2004-03-06', 'M', '114847221', '16352826767', 84),
('2004-09-16', 'K', '165352868', '47362716273', 85),
('2004-08-16', 'K', '117821726', '64316643736', 86),
('2004-10-05', 'K', '357321633', '47847686848', 87),
('2004-01-08', 'M', '243763756', '31468688876', 88),
('2004-08-21', 'K', '786662536', '63328621441', 89),
('2004-10-20', 'M', '318516316', '41725636858', 90),
('2004-10-08', 'K', '726155232', '25445326347', 91),
('2004-04-13', 'M', '735355874', '31555684764', 92),
('2004-04-06', 'K', '322151842', '47236446846', 93),
('2004-09-03', 'K', '436646255', '58225463265', 94),
('2004-11-21', 'K', '624781546', '81113875464', 95),
('2004-10-17', 'M', '737615553', '12574461266', 96),
('2004-07-23', 'K', '422245874', '55315526871', 97),
('2004-10-16', 'M', '563432884', '41187876553', 98),
('2004-07-11', 'M', '388862263', '76232124845', 99),
('2004-03-19', 'M', '442773227', '71143818385', 100),
('2004-01-22', 'M', '586714446', '57736636836', 101),
('2004-01-02', 'K', '565134543', '22662335575', 102),
('2004-05-18', 'K', '845163436', '53421541383', 103),


```
INSERT INTO Classrooms VALUES
```

```
INSERT INTO Lessons VALUES
```

```
('Wiedza o społeczeństwie', 'Piątek', '09:45:00', '10:25:00', 2, 12, 9),
```

```
('Edukacja dla bezpieczeństwa', 'Poniedziałek', '08:00:00', '08:45:00', 2, 13, 8),
('Informatyka', 'Poniedziałek', '08:50:00', '09:35:00', 2, 14, 7),
('Zajęcia wychowawcze', 'Poniedziałek', '09:45:00', '10:25:00', 2, 15, 6),
('Geografia', 'Środa', '08:00:00', '08:45:00', 3, 1, 20),
('Język polski', 'Środa', '08:50:00', '09:35:00', 3, 2, 19),
('Chemia', 'Środa', '09:45:00', '10:25:00', 3, 3, 18),
('Religia', 'Czwartek', '08:00:00', '08:45:00', 3, 4, 17),
('Historia', 'Czwartek', '08:50:00', '09:35:00', 3, 5, 16),
('Biologia', 'Czwartek', '09:45:00', '10:25:00', 3, 6, 15),
('Matematyka', 'Piątek', '08:00:00', '08:45:00', 3, 7, 14),
('Fizyka', 'Piątek', '08:50:00', '09:35:00', 3, 8, 13),
('Język angielski', 'Piątek', '09:45:00', '10:25:00', 3, 9, 12),
('Historia', 'Poniedziałek', '08:00:00', '08:45:00', 3, 10, 11),
('Muzyka', 'Poniedziałek', '08:50:00', '09:35:00', 3, 11, 10),
('Wiedza o społeczeństwie', 'Poniedziałek', '09:45:00', '10:25:00', 3, 12, 9),
('Edukacja dla bezpieczeństwa', 'Wtorek', '08:00:00', '08:45:00', 3, 13, 8),
('Informatyka', 'Wtorek', '08:50:00', '09:35:00', 3, 14, 7),
('Zajęcia wychowawcze', 'Wtorek', '09:45:00', '10:25:00', 3, 15, 6),
```

```
('Geografia', 'Czwartek', '08:00:00', '08:45:00', 4, 1, 20),
('Język polski', 'Czwartek', '08:50:00', '09:35:00', 4, 2, 19),
('Chemia', 'Czwartek', '09:45:00', '10:25:00', 4, 3, 18),
('Religia', 'Piątek', '08:00:00', '08:45:00', 4, 4, 17),
('Historia', 'Piątek', '08:50:00', '09:35:00', 4, 5, 16),
('Biologia', 'Piątek', '09:45:00', '10:25:00', 4, 6, 15),
('Matematyka', 'Poniedziałek', '08:00:00', '08:45:00', 4, 7, 14),
('Fizyka', 'Poniedziałek', '08:50:00', '09:35:00', 4, 8, 13),
('Język angielski', 'Poniedziałek', '09:45:00', '10:25:00', 4, 9, 12),
('Historia', 'Wtorek', '08:00:00', '08:45:00', 4, 10, 11),
('Muzyka', 'Wtorek', '08:50:00', '09:35:00', 4, 11, 10),
('Wiedza o społeczeństwie', 'Wtorek', '09:45:00', '10:25:00', 4, 12, 9),
('Edukacja dla bezpieczeństwa', 'Środa', '08:00:00', '08:45:00', 4, 13, 8),
('Informatyka', 'Środa', '08:50:00', '09:35:00', 4, 14, 7),
('Zajęcia wychowawcze', 'Środa', '09:45:00', '10:25:00', 4, 15, 6),
```

```
('Geografia', 'Piątek', '08:00:00', '08:45:00', 5, 1, 20),
('Język polski', 'Piątek', '08:50:00', '09:35:00', 5, 2, 19),
('Chemia', 'Piątek', '09:45:00', '10:25:00', 5, 3, 18),
('Religia', 'Poniedziałek', '08:00:00', '08:45:00', 5, 4, 17),
('Historia', 'Poniedziałek', '08:50:00', '09:35:00', 5, 5, 16),
('Biologia', 'Poniedziałek', '09:45:00', '10:25:00', 5, 6, 15),
('Matematyka', 'Wtorek', '08:00:00', '08:45:00', 5, 7, 14),
('Fizyka', 'Wtorek', '08:50:00', '09:35:00', 5, 8, 13),
('Język angielski', 'Wtorek', '09:45:00', '10:25:00', 5, 9, 12),
('Historia', 'Środa', '08:00:00', '08:45:00', 5, 10, 11),
('Muzyka', 'Środa', '08:50:00', '09:35:00', 5, 11, 10),
('Wiedza o społeczeństwie', 'Środa', '09:45:00', '10:25:00', 5, 12, 9),
('Edukacja dla bezpieczeństwa', 'Czwartek', '08:00:00', '08:45:00', 5, 13, 8),
('Informatyka', 'Czwartek', '08:50:00', '09:35:00', 5, 14, 7),
('Zajęcia wychowawcze', 'Czwartek', '09:45:00', '10:25:00', 5, 15, 6),
```

```
('Język polski', 'Piątek', '08:00:00', '08:45:00', 6, 2, 19),
('Chemia', 'Piątek', '08:50:00', '09:35:00', 6, 3, 18),
('Geografia', 'Piątek', '09:45:00', '10:25:00', 5, 1, 20),
('Biologia', 'Poniedziałek', '08:00:00', '08:45:00', 5, 4, 17),
('Religia', 'Poniedziałek', '08:50:00', '09:35:00', 5, 5, 16),
('Historia', 'Poniedziałek', '09:45:00', '10:25:00', 5, 6, 15),
('Fizyka', 'Wtorek', '08:00:00', '08:45:00', 5, 7, 14),
('Język angielski', 'Wtorek', '08:50:00', '09:35:00', 5, 8, 13),
('Matematyka', 'Wtorek', '09:45:00', '10:25:00', 5, 9, 12),
('Muzyka', 'Środa', '08:00:00', '08:45:00', 5, 10, 11),
('Wiedza o społeczeństwie', 'Środa', '08:50:00', '09:35:00', 5, 11, 10),
('Historia', 'Środa', '09:45:00', '10:25:00', 5, 12, 9),
('Informatyka', 'Czwartek', '08:00:00', '08:45:00', 5, 13, 8),
('Zajęcia wychowawcze', 'Czwartek', '08:50:00', '09:35:00', 5, 14, 7),
('Edukacja dla bezpieczeństwa', 'Czwartek', '09:45:00', '10:25:00', 5, 15, 6)
```

```
INSERT INTO Authors VALUES
('Adam','Mickiewicz'),
('Juliusz', 'Słowacki'),
('Ernest','Hemingway'),
('Fiodor','Dostojewski'),
('Henryk', 'Sienkiewicz'),
('Thomas', 'Cormen')
```

```
INSERT INTO Books VALUES
(1, 'Dziady III',10, 'Dramat'),
(1, 'Pan Tadeusz',20,'Epopeja'),
(2, 'Kordian', 20, 'Dramat'),
(3, 'Stary człowiek i morze', 5, 'Powieść'),
(4, 'Zbrodnia i Kara', 15, 'Literatura rosyjska'),
(4, 'Bracia Karamazow', 3, 'Literatura rosyjska'),
(5, 'Ogniem i mieczem', 10, 'Powieść historyczna'),
(5, 'Quo voids',11, 'Powieść'),
(6, 'Introduction to Algorithms', 1, 'Algorytmy i struktury danych')
```

```
INSERT INTO Borrows VALUES
(23, 8, '2020-12-12', NULL),
(12, 3, '2020-12-14', '2021-01-06'),
(1, 1, '2021-01-01', NULL),
(4, 4, '2020-11-01', NULL),
(20, 9, '2020-10-01', NULL),
(2, 2, '2021-01-09', '2021-01-10'),
(23, 5, '2020-12-12', '2020-12-20')
```

```
INSERT INTO Grades VALUES
--(1, 1, 'Chemia', 5.0, '2020-12-12', 'aktywnosc'),
(1, 2, 'Informatyka', 4.5, '2021-01-02', 'kartkowka'),
```



```

(1, 1, 'Fizyka', 5.0, '2020-10-13', 'sprawdzian'),
(1, 1, 'Historia', 5.0, '2020-03-19', 'sprawdzian'),
(2, 1, 'Biologia', 4.5, '2020-06-12', 'aktywnosc'),
(2, 2, 'Geografia', 5.0, '2021-02-10', 'sprawdzian'),
(2, 3, 'Muzyka', 4.5, '2021-01-02', 'kartkowka'),
(2, 3, 'Język polski', 6.0, '2020-12-25', 'konkurs'),
(3, 1, 'Matematyka', 3.0, '2020-11-11', 'aktywnosc'),
(3, 2, 'Historia', 1.0, '2020-09-01', 'zadanie domowe'),
(15, 30, 'Fizyka', 3, '2020-02-16', 'kartkowka'),
(16, 30, 'Fizyka', 2.5, '2020-02-16', 'kartkowka'),
(17, 30, 'Fizyka', 3.5, '2020-02-16', 'kartkowka'),
(18, 30, 'Fizyka', 5, '2020-02-16', 'kartkowka'),
(19, 30, 'Fizyka', 4.5, '2020-02-16', 'kartkowka'),
(20, 30, 'Fizyka', 3, '2020-02-16', 'kartkowka'),
(21, 30, 'Fizyka', 1, '2020-02-16', 'kartkowka'),
(22, 30, 'Fizyka', 2, '2020-02-16', 'kartkowka'),
(23, 30, 'Fizyka', 4, '2020-02-16', 'kartkowka'),
(24, 30, 'Fizyka', 5, '2020-02-16', 'kartkowka'),
(25, 30, 'Fizyka', 5, '2020-02-16', 'kartkowka'),
(100, 29, 'Język polski', 3, '2020-01-12', 'wypracowanie'),
(101, 29, 'Język polski', 4, '2020-01-12', 'wypracowanie'),
(102, 29, 'Język polski', 1, '2020-01-12', 'wypracowanie'),
(103, 29, 'Język polski', 1, '2020-01-12', 'wypracowanie'),
(104, 29, 'Język polski', 2, '2020-01-12', 'wypracowanie'),
(105, 29, 'Język polski', 3.5, '2020-01-12', 'wypracowanie'),
(106, 29, 'Język polski', 5, '2020-01-12', 'wypracowanie'),
(107, 29, 'Język polski', 4.5, '2020-01-12', 'wypracowanie'),
(108, 29, 'Język polski', 5, '2020-01-12', 'wypracowanie'),
(109, 29, 'Język polski', 3, '2020-01-12', 'wypracowanie'),
(110, 29, 'Język polski', 4, '2020-01-12', 'wypracowanie'),
(111, 29, 'Język polski', 3.5, '2020-01-12', 'wypracowanie'),
(112, 29, 'Język polski', 2, '2020-01-12', 'wypracowanie'),
(113, 29, 'Język polski', 5, '2020-01-12', 'wypracowanie'),
(114, 29, 'Język polski', 4, '2020-01-12', 'wypracowanie'),
(115, 29, 'Język polski', 4, '2020-01-12', 'wypracowanie'),
(116, 29, 'Język polski', 3.5, '2020-01-12', 'wypracowanie'),
(117, 29, 'Język polski', 2, '2020-01-12', 'wypracowanie'),
(118, 29, 'Język polski', 1, '2020-01-12', 'wypracowanie'),
(119, 29, 'Język polski', 2, '2020-01-12', 'wypracowanie'),
(120, 29, 'Język polski', 1, '2020-01-12', 'wypracowanie')

INSERT INTO Absences VALUES
(14, '2020-03-19', 'N'),
(7, '2020-09-11', 'U'),
(13, '2021-09-27', 'U'),
(19, '2020-05-10', 'U'),
(6, '2020-03-16', 'U'),
(19, '2020-02-11', 'N'),
(2, '2020-09-22', 'N'),
(22, '2020-04-27', 'N'),
(22, '2021-04-18', 'U')

INSERT INTO Exams VALUES
(1, 1, '2021-01-12'),
(1, 3, '2021-01-19'),
(2, 1, '2021-01-11'),
(2, 3, '2021-01-18'),
(3, 1, '2021-01-10'),
(3, 3, '2021-01-17')

INSERT INTO ExamResults VALUES
(1, 4, 1, '2021-01-19'),
(1, 6, 1.5, '2021-01-19'),
(1, 10, 2, '2021-01-19'),
(1, 13, 3, '2021-01-19'),
(1, 19, 4, '2021-01-19'),
(1, 23, 5, '2021-01-19'),
(1, 28, 6, '2021-01-19'),
(1, 29, 2, '2021-01-19'),
(2, 4, 1, '2021-01-19'),
(2, 6, 1.5, '2021-01-19'),
(2, 10, 2, '2021-01-19'),
(2, 13, 3, '2021-01-19'),
(2, 19, 4, '2021-01-19'),
(2, 23, 5, '2021-01-19'),
(2, 28, 6, '2021-01-19'),
(2, 29, 2, '2021-01-19'),
(3, 8, 1, '2021-01-18'),
(3, 11, 1.5, '2021-01-18'),
(3, 14, 2, '2021-01-18'),
(3, 20, 3, '2021-01-18'),
(3, 22, 4, '2021-01-18'),
(3, 24, 5, '2021-01-18'),
(3, 27, 6, '2021-01-18'),
(4, 8, 1, '2021-01-18'),
(4, 11, 1.5, '2021-01-18'),
(4, 14, 2, '2021-01-18'),
(4, 20, 3, '2021-01-18'),
(4, 22, 4, '2021-01-18'),
(4, 24, 5, '2021-01-18'),
(4, 27, 6, '2021-01-18'),
(5, 1, 1, '2021-01-17'),
(5, 2, 1.5, '2021-01-17')

INSERT INTO ExtraLessons VALUES
('Koło teatralne', 'Wtorek', '13:00:00', '14:30:00', 12, 3, 17),
('Koło matematyczne', 'Czwartek', '12:15:00', '13:15:00', 17, 4, 7),

```

```
('Koło fizyczne', 'Piątek', '11:15:00', '12:00:00', 3, 6, 12),
('SKS - Koszykówka', 'Poniedziałek', '14:00:00', '15:30:00', 21, 2, 29),
('Koło chemiczne', 'Środa', '12:15:00', '13:15:00', 23, 1, 27)
```