

# Kategoria Studies

## Tabela Studies

Zawiera informacje o dostępnych studiach:

- **StudiesID** [int] - klucz główny, identyfikator studiów
- **CoordinatorID** [int] - identyfikator kordynatora studiów
- **Name** [nvarchar(30)] - nazwa studiów
- **Description** [nvarchar(max)] - opis studiów
- **StartDate** [date] - data rozpoczęcia studiów
- **EndDate** [date] - data zakończenia studiów
- **Price** [money] - cena wpisowego na studiach
  - warunki: Price > 0
  - wartość domyślna: 1200

```
CREATE TABLE Studies (  
  StudiesID int NOT NULL,  
  CoordinatorID int NOT NULL,  
  Name nvarchar(30) NOT NULL,  
  Description nvarchar(max) NOT NULL,  
  StartDate date NOT NULL,  
  EndDate date NOT NULL,  
  Price money NOT NULL DEFAULT 1200 CHECK (Price > 0),  
  CONSTRAINT Studies_pk PRIMARY KEY (StudiesID)  
);
```

## Tabela Subjects

Zawiera informacje o przedmiotach:

- **SubjectID** [int] - klucz główny, identyfikator przedmiotu
- **StudiesID** [int] - klucz obcy, identyfikator studiów
- **Name** [nvarchar(50)] - nazwa przedmiotu
- **Description** [ntext] - opis przedmiotu

```
CREATE TABLE Subjects (  
  SubjectID int NOT NULL IDENTITY(1, 1),  
  StudiesID int NOT NULL,  
  Name nvarchar(50) NOT NULL,  
  Description nvarchar(max) NOT NULL,  
  CONSTRAINT Subjects_pk PRIMARY KEY (SubjectID)  
);
```

## Tabela Meetings

Zawiera informacje o pojedynczym spotkaniu na studiach z danego przedmiotu:

- **MeetingID** [int] - klucz główny, identyfikator spotkania
- **TeacherID** [int] - klucz obcy, identyfikator nauczyciela
- **SubjectID** [int] - klucz obcy, identyfikator przedmiotu, z którego jest dane spotkanie
- **ReunionID** [int] - klucz obcy, identyfikator zjazdu
- **Date** [date] - data spotkania
- **BeginningTime** [time(0)] - godzina rozpoczęcia spotkania
- **Duration** [time(0)] - czas trwania spotkania
  - warunki: Duration > '00:00:00'
  - wartość domyślna: '01:30:00'
- **Price** [money] - cena pojedynczego spotkania
  - warunki: Price > 0
  - wartość domyślna: 120
- **TypeID** [int] - klucz obcy, identyfikator typu spotkania np. stacjonarne itd.

```
CREATE TABLE Meetings (  
  MeetingID int NOT NULL,  
  TeacherID int NOT NULL,  
  SubjectID int NOT NULL,  
  ReunionID int NOT NULL,  
  DateAndBeginningTime datetime NOT NULL,  
  Duration time(0) NOT NULL DEFAULT '01:30:00' CHECK (Duration > '00:00:00'),  
  Price money NOT NULL DEFAULT 120 CHECK (Price > 0),
```

```
TypeID int NOT NULL,  
CONSTRAINT MeetingID PRIMARY KEY (MeetingID)  
);
```

## Tabela **In-person\_Meetings**

Zawiera dodatkowe informacje dla spotkań stacjonarnych:

- **MeetingID** [int] - klucz główny, identyfikator spotkania
- **Classroom** [int] - numer sali spotkania
- **TranslatorID** [int, nullable] - identyfikator tłumacza
- **LanguageID** [int] - klucz obcy, identyfikator języka w jakim odbywa się spotkanie
- **Limit** [int] - limit miejsc na spotkaniu
  - warunki: Limit > 0
  - wartość domyślna: 25

```
CREATE TABLE In_person_Meetings (  
    MeetingID int NOT NULL,  
    Classroom int NOT NULL,  
    TranslatorID int NULL,  
    LanguageID int NOT NULL,  
    Limit int NOT NULL DEFAULT 25 CHECK (Limit > 0),  
    CONSTRAINT In_person_Meetings_pk PRIMARY KEY (MeetingID)  
);
```

## Tabela **Online\_Sync\_Meetings**

Zawiera dodatkowe informacje dla spotkań online synchronicznie:

- **MeetingID** [int] - klucz główny, identyfikator spotkania
- **MeetingLink** [nvarchar(100), unique] - link do spotkania
- **RecordingLink** [nvarchar(100)] - link do nagrania spotkania
- **TranslatorID** [int, nullable] - identyfikator tłumacza
- **LanguageID** [int] - identyfikator języka w jakim odbywa się spotkanie

```
CREATE TABLE Online_Sync_Meetings (  
    MeetingID int NOT NULL,  
    MeetingLink nvarchar(100) NOT NULL,  
    RecordingLink nvarchar(100) NULL,  
    TranslatorID int NULL,  
    LanguageID int NOT NULL,  
    CONSTRAINT OnlineSyncMeetingMeetingLink UNIQUE (MeetingLink),  
    CONSTRAINT Online_Sync_Meetings_pk PRIMARY KEY (MeetingID)  
);
```

## Tabela **Online\_Async\_Meetings**

Zawiera dodatkowe informacje dla spotkań online asynchronicznie:

- **MeetingID** [int] - klucz główny, identyfikator spotkania
- **RecordingLink** [nvarchar(100)] - link do nagrania

```
CREATE TABLE Online_Async_Meetings (  
    MeetingID int NOT NULL,  
    RecordingLink nvarchar(100) NOT NULL,  
    CONSTRAINT Online_Async_Meetings_pk PRIMARY KEY (MeetingID)  
);
```

## Tabela **User\_Meetings\_Attendance**

Zawiera informacje o obecności studenta na spotkaniu:

- **UserID** [int] - część klucza głównego, identyfikator studenta
- **MeetingID** [int] - część klucza głównego, identyfikator spotkania
- **Present** [bit] - informacja o obecności studenta na spotkaniu

```
CREATE TABLE Users_Meetings_Attendance (  
    UserID int NOT NULL,
```

```

MeetingID int NOT NULL,
Present bit NULL,
CONSTRAINT Users_Meetings_Attendance_pk PRIMARY KEY (MeetingID,UserID)
);

```

## Tabela Practices

Zawiera informacje o firmach gdzie mogą być odbywane praktyki:

- **PracticeID** [int] - klucz główny, identyfikator praktyki
- **Description** [ntext] - opis firmy gdzie odbywają się praktyki
- **CompanyName** [nvarchar(30)] - nazwa firmy
- **Country** [nvarchar(30)] - kraj gdzie jest zarejestrowana firma
- **City** [nvarchar(30)] - miasto siedziby firmy
- **Address** [nvarchar(30)] - adres firmy
- **Phone** [varchar(20)] - numer telefonu do firmy
  - warunki: LEN(Phone) <= 20
- **Email** [nvarchar(50)] - email firmy
  - warunki: Email LIKE '%\_@%.%'

```

CREATE TABLE Practices (
    PracticeID int NOT NULL IDENTITY(1, 1),
    Description nvarchar(max) NOT NULL,
    CompanyName nvarchar(30) NOT NULL,
    Country nvarchar(30) NOT NULL,
    City nvarchar(30) NOT NULL,
    Address nvarchar(50) NOT NULL,
    Phone varchar(20) NOT NULL CHECK (LEN(Phone) <= 20),
    Email nvarchar(50) NOT NULL CHECK (Email LIKE '%_@%.%'),
    CONSTRAINT PracticePhone UNIQUE (Phone),
    CONSTRAINT PracticeEmail UNIQUE (Email),
    CONSTRAINT Practices_pk PRIMARY KEY (PracticeID)
);

```

## Tabela Users\_Practices\_Attendance

Zawiera informacje o zdaniu praktyk przez danego studenta:

- **UserID** [int] - część klucza głównego, identyfikator studenta
- **StudiesID** [int] - część klucza głównego, identyfikator studiów
- **PracticeID** [int] - część klucza głównego, identyfikator praktyk
- **Present** [bit] - informacja o zdaniu praktyk

```

CREATE TABLE Users_Practices_Attendance (
    UserID int NOT NULL,
    StudiesID int NOT NULL,
    PracticeID int NOT NULL,
    Present bit NULL,
    CONSTRAINT Users_Practices_Attendance_pk PRIMARY KEY (UserID,StudiesID,PracticeID)
);

```

## Tabela Users\_Studies

Zawiera informacje studentach przypisanych do danych studiów:

- **UserID** [int] - część klucza głównego, identyfikator studenta
- **StudiesID** [int] - część klucza głównego, identyfikator studiów
- **Grade** [int] - wartość oceny studenta na koniec studiów
  - warunki: Grade >= 2 AND Grade <= 5

```

CREATE TABLE Users_Studies (
    UserID int NOT NULL,
    StudiesID int NOT NULL,
    Grade int NULL CHECK (Grade >= 2 AND Grade <= 5),
    CONSTRAINT Users_Studies_pk PRIMARY KEY (UserID,StudiesID)
);

```

## Tabela Studies\_Reunion

Zawiera ona informacje o zjazdach występujących na danych studiach:

- **ReunionID** [int] - klucz główny, identyfikator zjazdu
- **ProductID** [int] - klucz obcy, informacja o ID jaki ma dany zjazd w produktach (potrzebny do określenia czy dany użytkownik zapłacił za dany zjazd czy nie)
- **StudiesID** [int] - klucz poboczny, identyfikator studiów
- **StartDate** [date] - data startu danego zjazdu
- **EndDate** [date] - data końca danego zjazdu

```
CREATE TABLE Studies_Reunion (  
  ReunionID int NOT NULL IDENTITY(1, 1),  
  ProductID int NOT NULL,  
  StudiesID int NOT NULL,  
  StartDate date NOT NULL,  
  EndDate date NOT NULL,  
  CONSTRAINT Studies_Reunion_pk PRIMARY KEY (ReunionID)  
);
```