

---

# Podstawy baz danych

---

**Dzień i godzina zajęć:** Środa 15:00

**Nr zespołu:** 2

**Autorzy:** Dariusz Marecik, Filip Węgrzyn, Paweł Fornagiel

**Link do repozytorium git:** <https://github.com/pFornagiel/bazy-danych-2025>

---

Założenia dotyczące projektu:

- W zakres studiów wchodzi pojedyncze przedmioty (studium), które mają przypisane spotkania
- 

## 1. Wymagania i funkcje systemu

---

### Opis Funkcjonalności Systemu

---

#### Funkcje Systemu

- Weryfikacja limitu zapisanych osób i blokowanie jego przekroczenia
- Blokowanie zapisu / dostępu do treści po upływie terminu ważności
- Blokowanie możliwości zapisania się na te same zajęcia wiele razy

#### Użytkownicy

- Studenci (użytkownicy zalogowani)
- Goście (użytkownicy niezalogowani)
- Prowadzący zajęcia
- Dyrektor Szkoły
- Administrator zasobów
- Dziekanat
- Tłumacz

#### Funkcje poszczególnych użytkowników

Studenci (użytkownicy zalogowani, rozszerzenie możliwości gości)

- możliwość zapisania się na kurs
- zapis na praktyki
- usunięcie konta
- dodanie i usunięcie adresu korespondencyjnego
- wyświetlenie wykazu zajęć w których brał udział / obecności
- wyświetlenie frekwencji / stopnia zaliczenia dla poszczególnych zajęć

- wyświetlenie dostępnych kursów / webinarów / studiów
- wyświetlanie linków dostępu do udostępnionych zasobów
- dodanie, usunięcie i przegląd elementów w koszyku
- stworzenie zamówienia
- opłacenie zamówienia

### Goście (użytkownicy niezalogowani)

- dostęp do wybranych webinarów
- przegląd dostępnych webinarów / studiów / kursów
- założenie konta

### Prowadzący zajęcia

- modyfikacja terminu zajęć
- modyfikacja udostępnionych zasobów
- sprawdzanie obecności dla każdych zajęć
- wyświetlenie wykazu prowadzonych zajęć

### Administrator zasobów

- dodawanie / usuwanie webinarów, kursów i studiów
- dodawanie / usuwanie materiałów

### Dyrektor

- dodawanie / usuwanie pracowników
- modyfikacja dostępu do kursu
- modyfikacja opłat za kurs
- modyfikacja czasu na dokonanie płatności dla danej osoby
- przegląd wszelkich danych dotyczących realizowanych zajęć

### Dziekanat

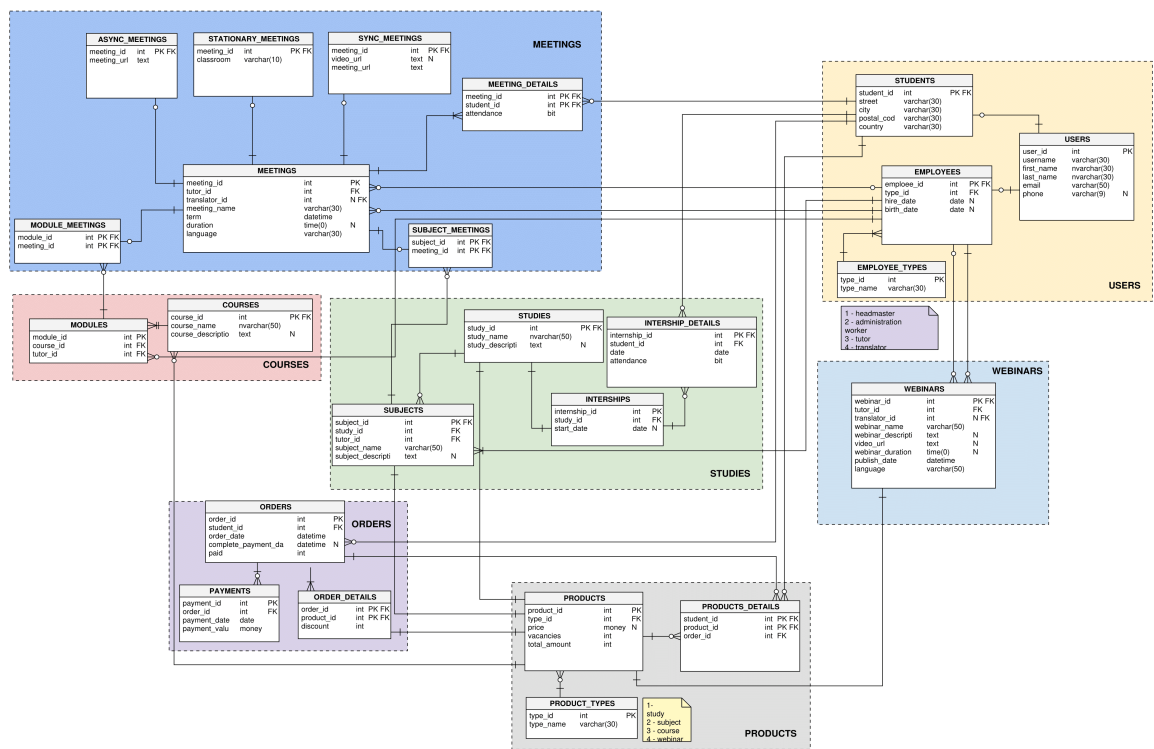
- tworzenie dyplomów potwierdzających ukończenie kursu / studium
- dodawanie / modyfikacja praktyk
- modyfikacja webinarów / kursów / studiów / przedmiotów
- dodawanie webinarów / kursów / studiów / przedmiotów
- dodawanie / usuwanie tłumacza do wybranych przedmiotów
- tworzenie sylabusu
- generowanie harmonogramu
- generowanie danych dotyczących realizowanych zajęć
- wyświetlenie zatrudnionych pracowników
- wyświetlenie studentów przypisanych do danego zasobu wraz z limitami zasobu
- wyświetlenie danych dotyczących wybranych form zajęć
- wykrywanie i wyświetlanie kolizji czasowych studentów
- Raportowanie:
  - Tworzenie raportu liczby zapisanych osób na przyszłe wydarzenia wraz z informacjami o wydarzeniach

- Tworzenie raportu dotyczącego frekwencji na zakończonych wydarzeniach
- Tworzenie raportu dotyczącego osób, które skorzystały z usług, ale nie uiściły opłat
- Tworzenie raportów finansowych
- Tworzenie list obecności dla poszczególnych form zajęć
- Tworzenie list kolizji czasowych wśród użytkowników

Tłumacz

- Dostęp do zasobów poszczególnych kursów / studiów i webinarów
- Dodawanie przetłumaczonych zasobów do kursów /studiów / webinarów

Schemat bazy danych



Opis tabel

Kategoria Users

Tabela Users

Column Name	Data Type	Properties
user_id	int	Primary Key

Column Name	Data Type	Properties
username	varchar(30)	
first_name	nvarchar(30)	
last_name	nvarchar(30)	
email	varchar(50)	
phone	varchar(9)	
CONSTRAINT	unique_email	
CONSTRAINT	unique_phone	

Zawiera podstawowe informacje o każdym użytkowniku bazy.

- *user\_id* int - klucz główny, identyfikuje użytkownika
- username varchar(30) - nazwa użytkownika w bazie danych
- first\_name nvarchar(30) - imię użytkownika
- last\_name nvarchar(30) - nazwisko użytkownika
- email varchar(50) - email użytkownika
  - warunek: (mail LIKE '%\_@%.%')
- phone varchar(9) nullable - numer telefonu użytkownika
  - warunek: LEN(Phone) = 15 AND ISNUMERIC(Phone) = 1

```
-- Table: USERS
CREATE TABLE USERS (
  user_id int NOT NULL IDENTITY,
  username varchar(30) NOT NULL,
  first_name nvarchar(30) NOT NULL,
  last_name nvarchar(30) NOT NULL,
  email varchar(50) NOT NULL CHECK (mail LIKE '%_@%.%'),
  phone varchar(9) NULL CHECK (LEN(Phone) = 9 AND ISNUMERIC(Phone) = 1),
  CONSTRAINT unique_email UNIQUE (email),
  CONSTRAINT unique_phone UNIQUE (phone),
  CONSTRAINT USERS_pk PRIMARY KEY (user_id)
);
```

Tabela Students

Column Name	Data Type	Properties
student_id	int	Primary Key Foreign Key

Column Name	Data Type	Properties
street	varchar(30)	
city	varchar(30)	
postal_code	varchar(30)	
country	varchar(30)	

Zawiera infromacje specyficzne dla studenta

- *student\_id* int - klucz główny, klucz obcy, identyfikuje studenta
- *street* varchar(30) - ulica, na której mieszka studenta
- *city* varchar(30) - miasto, w którym mieszka studenta
- *postal\_code* varchar(30) - kod pocztowy studenta
- *country* varchar(30) - kraj pochodzenia studenta

```
-- Table: STUDENTS
CREATE TABLE STUDENTS (
  student_id int NOT NULL,
  street varchar(30) NOT NULL,
  city varchar(30) NOT NULL,
  postal_code varchar(30) NOT NULL,
  country varchar(30) NOT NULL,
  CONSTRAINT STUDENTS_pk PRIMARY KEY (student_id)
);
```

Tabela EMPLOYEES

Column Name	Data Type	Properties
emploee_id	int	Primary Key Foreign Key
type_id	int	
hire_date	date	
birth_date	date	

Zawiera szczególne informacje dla pracowników (dyrektora, pracownika dziekanatu, nauczyciela, tłumacza)

- *emploee\_id* int - klucz główny, klucz obcy, identyfikator pracownika
- *type\_id* int - sklucz obcy, typ pracownika (opisany poniżej)
- *hire\_date* date nullable - data zatrudnienia

- DEFAULT current\_date
- birth\_date date nullable - data urodzin pracownika
  - DEFAULT current\_date

```
-- Table: EMPLOYEES
CREATE TABLE EMPLOYEES (
  employee_id int NOT NULL,
  type_id int NOT NULL,
  hire_date date NULL DEFAULT current_date,
  birth_date date NULL DEFAULT current_date,
  CONSTRAINT EMPLOYEES_pk PRIMARY KEY (employee_id)
);
```

Tabela EMPLOYEES\_TYPE

Column Name	Data Type	Properties
type_id	int	Primary Key
type_name	varchar(30)	

Zawiera opis typów pracowników

- type\_id int - klucz główny, typ pracownika
  - 1 - headmaster
  - 2 - administration worker
  - 3 - tutor
  - 4 - translator
- type\_name varchar(30) - nazwa pełnionej funkcji

```
-- Table: EMPLOYEE_TYPES
CREATE TABLE EMPLOYEE_TYPES (
  type_id int NOT NULL IDENTITY,
  type_name varchar(30) NOT NULL,
  CONSTRAINT EMPLOYEE_TYPES_pk PRIMARY KEY (type_id)
);
```

Kategoria Products

Tabela Products

Column Name	Data Type	Properties
product_id	int	Primary Key Foreign Key

Column Name	Data Type	Properties
type_id	int	
price	money	
vacancies	int	
total_amount	int	

Zawiera informacje o każdym produkcie w ofercie. Produkt jest rozumiany jako każda z form przeprowadzania zajęć.

- *product\_id* int - klucz główny, identyfikuje produkt
- *type\_id* int - klucz obcy, numer kategorii produktu
- *price* money nullable - cena za produkt
  - warunek: *price* >= 0
  - DEFAULT 1000
- *vacancies* int - ilość wolnych miejsc możliwych do zakupu na dane zajęcia
  - warunek: *vacancies* >= 0
- *total\_amount* int - liczba wszystkich dostępnych miejsc na dane zajęcia

```
-- Table: PRODUCTS
CREATE TABLE PRODUCTS (
  product_id int NOT NULL IDENTITY,
  type_id int NOT NULL,
  price money NULL DEFAULT 1000 CHECK (price>=0),
  vacancies int NOT NULL CHECK (vacancies>=0),
  total_amount int NOT NULL DEFAULT 30 CHECK (total_amount>0),
  CONSTRAINT product_id PRIMARY KEY (product_id)
);
```

Tabela PRODUCTS\_DETAILS

Column Name	Data Type	Properties
student_id	int	Primary Key Foreign Key
product_id	int	Primary Key Foreign Key
order_id	int	

Zawiera informacje o studentach zapisanych na dane zajęcia oraz o numerze zamówienia z jakiego został kupiony dostęp do zajęć

- student\_id int - wchodzi w skład klucza głównego, klucz obcy, identyfikuje studenta
- product\_id int - wchodzi w skład klucza głównego, klucz obcy, identyfikuje produkt
- order\_id int - klucz obcy, identyfikuje zamówienie z jakiego został kupiony dostęp do zajęć

```
-- Table: PRODUCTS_DETAILS
CREATE TABLE PRODUCTS_DETAILS (
  student_id int NOT NULL,
  product_id int NOT NULL,
  order_id int NOT NULL,
  CONSTRAINT PRODUCTS_DETAILS_pk PRIMARY KEY (student_id,product_id)
);
```

Tabela PRODUCT\_TYPES

Column Name	Data Type	Properties
type_id	int	Primary Key
type_name	varchar(30)	

Zawiera informacje o typach produktów

- type\_id int - klucz główny, identyfikuje typ:
  - 1- study,
  - 2 - subject,
  - 3 - course,
  - 4 - webinar
- type\_name varchar(30) - nazwa typu

```
-- Table: PRODUCT_TYPES
CREATE TABLE PRODUCT_TYPES (
  type_id int NOT NULL IDENTITY,
  type_name varchar(30) NOT NULL,
  CONSTRAINT PRODUCT_TYPES_pk PRIMARY KEY (type_id)
);
```

Kategoria Orders

Tabela ORDERS

Column Name	Data Type	Properties
order_id	int	Primary Key Foreign Key
student_id	int	
order_date	datetime	



Column Name	Data Type	Properties
complete_payment_date	datetime	
paid	int	

Zawiera informacje na temat zamówienia pod danym identyfikatorem

- order\_id int - klucz główny, identyfikator zamówienia
- student\_id int - kluczo obcy, identyfikator studenta
- order\_date datetime nullable - data złożenia zamówienia
- complete\_payment\_date datetime nullable - data opłacenia pełnej kwoty zamówienia
- paid int - kwota, którą zapłacił student w ramach zamówienia (nie musi być to pełna kwota zamówienia)

```
-- Table: ORDERS
CREATE TABLE ORDERS (
  order_id int NOT NULL IDENTITY,
  student_id int NOT NULL,
  order_date datetime NOT NULL DEFAULT actual_date,
  complete_payment_date datetime NULL,
  paid int NOT NULL DEFAULT 0 CHECK (paid>=0),
  CONSTRAINT ORDERS_pk PRIMARY KEY (order_id)
);
```

Tabela ORDER\_DETAILS

Column Name	Data Type	Properties
order_id	int	Primary Key Foreign Key
product_id	int	Primary Key Foreign Key
discount	int	

Tabela zawiera informacje o produktach wchodzących w skład danego zamówienia i rabacie udzielonym dla danego zamówienia

- order\_id int - wchodzi w skład klucza głównego, klucz obcy, identyfikator zamówienia
- product\_id int - wchodzi w skład klucza głównego, klucz obcy, identyfikator produktu
- discount int - rabat udzielony dla zamówienia
  - Warunki: discount >= 0 and discount <= 1
  - DEFAULT 0

```
-- Table: ORDER_DETAILS
CREATE TABLE ORDER_DETAILS (
  order_id int NOT NULL,
  product_id int NOT NULL,
  discount int NOT NULL DEFAULT 0 CHECK (discount >= 0 and discount <= 1),
  CONSTRAINT ORDER_DETAILS_pk PRIMARY KEY (order_id,product_id)
);
```

Tabela PAYMENTS

Column Name	Data Type	Properties
payment_id	int	Primary Key Foreign Key
order_id	int	
payment_date	date	
payment_value	money	

Zawiera informacje o płatności dołączonej do danego zamówienia

- payment\_id int - klucz główny, identyfikator płatności
- order\_id int - klucz obcy, identyfikator zamówienia
- payment\_date date - data dokonania płatności
- payment\_value money - opłacona płatności
  - warunek: payment\_value >= 0

```
-- Table: PAYMENTS
CREATE TABLE PAYMENTS (
  payment_id int NOT NULL IDENTITY,
  order_id int NOT NULL,
  payment_date date NOT NULL DEFAULT actual_date,
  payment_value money NOT NULL CHECK (payment_value>=0),
  CONSTRAINT PAYMENTS_pk PRIMARY KEY (payment_id)
);
```

Kategoria Webinars

Tabela Webinars

Column Name	Data Type	Properties
webinar_id	int	Primary Key Foreign Key

Column Name	Data Type	Properties
tutor_id	int	
translator_id	int	
webinar_name	varchar(50)	
webinar_description	text	
video_url	text	
webinar_duration	time(0)	
publish_date	datetime	
language	varchar(50)	

Zawiera informacje specyficzne dla każdego produktu będącego webinarem

- webinar\_id int - klucz główny, klucz obcy, identyfikator webinaru
- tutor\_id int - klucz obcy, identyfikator nauczyciela
- translator\_id int nullable - klucz obcy, identyfikator tłumacza
- webinar\_name varchar(50) - nazwa webinaru
- webinar\_description text nullable - opis webinaru
- video\_url text nullable - link do zapisu webinaru
- webinar\_duration time(0) - czas trwania webinaru
  - warunek: DurationTime > '00:00:00'
  - DEFAULT 01:30:00
- publish\_date datetime - data przeprowadzenia i udostępnienia materiałów video
- language varchar(50) - język w jakim był prowadzony webinar
  - DEFAULT 'POLISH'

```
-- Table: WEBINARS
CREATE TABLE WEBINARS (
  webinar_id int NOT NULL,
  tutor_id int NOT NULL,
  translator_id int NULL,
  webinar_name varchar(50) NOT NULL,
  webinar_description text NULL,
  video_url text NULL,
  webinar_duration time(0) NULL DEFAULT 01:30:00 CHECK (DurationTime >
'00:00:00'),
  publish_date datetime NOT NULL,
```

```
language varchar(50) NOT NULL DEFAULT 'POLISH',
CONSTRAINT WEBINARS_pk PRIMARY KEY (webinar_id)
);
```

Kategoria COURSES

Tabela COURSES

Column Name	Data Type	Properties
course_id	int	Primary Key Foreign Key
course_name	nvarchar(50)	
course_description	text	

Zawiera informacje o produktach, które są kursami

- course\_id int - klucz główny, klucz obcy, identyfikator kursu
- course\_name nvarchar(50) - nazwa kursu
- course\_description text nullable - opis kursu

```
-- Table: COURSES
CREATE TABLE COURSES (
  course_id int NOT NULL,
  course_name nvarchar(50) NOT NULL,
  course_description text NULL,
  CONSTRAINT COURSES_pk PRIMARY KEY (course_id)
);
```

Tabela MODULES

Column Name	Data Type	Properties
module_id	int	Primary Key Foreign Key
course_id	int	
tutor_id	int	

Zawiera szczegółowe informacje dla każdego modułu kursu

- module\_id int - klucz główny, identyfikator modułu
- course\_id int - klucz obcy, identyfikator kursu, z którego pochodzi
- tutor\_id int - klucz obcy, identyfikator nauczyciela, który prowadzi dany moduł

```
-- Table: MODULES
CREATE TABLE MODULES (
  module_id int NOT NULL IDENTITY,
  course_id int NOT NULL,
  tutor_id int NOT NULL,
  CONSTRAINT MODULES_pk PRIMARY KEY (module_id)
);
```

Kategoria STUDIES

Tabela STUDIES

Column Name	Data Type	Properties
study_id	int	Primary Key Foreign Key
study_name	nvarchar(50)	
study_description	text	

Zawiera ogólne informacje o danych studiach

- study\_id int - klucz główny, klucz obcy, identyfikator studium
- study\_name nvarchar(50) - nazwa studium
- study\_description text nullable - opis studium

```
-- Table: STUDIES
CREATE TABLE STUDIES (
  study_id int NOT NULL,
  study_name nvarchar(50) NOT NULL,
  study_description text NULL,
  CONSTRAINT STUDIES_pk PRIMARY KEY (study_id)
);
```

Tabela SUBJECTS

Column Name	Data Type	Properties
subject_id	int	Primary Key Foreign Key
study_id	int	
tutor_id	int	
subject_name	varchar(50)	

Column Name	Data Type	Properties
subject_description	text	

Zawiera informacje szczegółowe informacje dotyczące przedmiotow

- subject\_id int - klucz główny, klucz obcy, identyfikator przedmiotu
- subject\_name varchar(50) - nazwa przedmiotu
- subject\_description text nullable - opis przedmiotu
- study\_id int - klucz obcy, identyfikator studiów, z których pochodzi przedmiot
- tutor\_id int - klucz obcy, identyfikator nauczyciela, który uczy dany przedmiot

```
-- Table: SUBJECTS
CREATE TABLE SUBJECTS (
  subject_id int NOT NULL,
  study_id int NOT NULL,
  tutor_id int NOT NULL,
  subject_name varchar(50) NOT NULL,
  subject_description text NULL,
  CONSTRAINT SUBJECTS_pk PRIMARY KEY (subject_id)
);
```

Tabela INTERSHIPS

Column Name	Data Type	Properties
internship_id	int	Primary Key Foreign Key
study_id	int	
start_date	date	

Zawiera informacje o praktykach prowadzonych na danych studiach

- internship\_id - klucz główny, identyfikator praktyk
- study\_id int - klucz obcy, identyfikator studiów
- start\_date date nullable - data rozpoczęcia praktyk

```
-- Table: INTERSHIPS
CREATE TABLE INTERSHIPS (
  internship_id int NOT NULL IDENTITY,
  study_id int NOT NULL,
  start_date date NULL,
```

```
        CONSTRAINT INTERSHIPS_pk PRIMARY KEY (internship_id)
    );
```

Tabela INTERSHIPS\_DETAILS

Column Name	Data Type	Properties
internship_id	int	Primary Key Foreign Key
student_id	int	Primary Key Foreign Key
date	date	
attendance	bit	

Zawiera szczegółowe informacje na temat danych praktyk

- internship\_id int - klucz główny, klucz obcy, identyfikator praktyk
- student\_id int - klucz główny, klucz obcy, identyfikator studenta biorącego udział w praktykach
- date date - data dnia praktyk
- attendance bit - obecność,  
1 - student uczestniczył w praktykach danego dnia,  
0 - student nie uczestniczył w praktykach danego dnia

```
-- Table: INTERSHIP_DETAILS
-- Table: INTERSHIP_DETAILS
CREATE TABLE INTERSHIP_DETAILS (
    internship_id int NOT NULL,
    student_id int NOT NULL,
    date date NOT NULL,
    attendance bit NOT NULL,
    CONSTRAINT INTERSHIP_DETAILS_pk PRIMARY KEY (internship_id)
);
```

Kategoria MEETINGS

Tabela MEETINGS

Column Name	Data Type	Properties
meeting_id	int	Primary Key Foreign Key
tutor_id	int	

Column Name	Data Type	Properties
translator_id	int	
meeting_name	varchar(30)	
term	datetime	
duration	time(0)	
language	varchar(30)	

Zawiera ogólne informacje na temat spotkania

- meeting\_id int - klucz główny, identyfikator spotkania
- tutor\_id int - klucz obcy, identyfikator nauczyciela prowadzącego spotkanie
- translator\_id int nullable nullable - klucz obcy, identyfikator tłumacza tłumaczącego spotkanie
- meeting\_name varchar(30) - nazwa spotkania
- term datetime - data i godzina spotkania
- duration time(0) nullable - czas trwania spotkania
  - Warunek: duration > '00:00:00'
  - DEFAULT 01:30:00
- language varchar(30) - język w jakim przeprowadza się spotkanie
  - DEFAULT 'POLISH'

```
-- Table: MEETINGS
CREATE TABLE MEETINGS (
  meeting_id int NOT NULL IDENTITY,
  tutor_id int NOT NULL,
  translator_id int NULL,
  meeting_name varchar(30) NOT NULL,
  term datetime NOT NULL,
  duration time(0) NULL DEFAULT 01:30:00 CHECK (duration>'00:00:00'),
  language varchar(30) NOT NULL DEFAULT 'POLISH',
  CONSTRAINT MEETINGS_pk PRIMARY KEY (meeting_id)
);
```

Tabela MEETING\_DETAILS

Column Name	Data Type	Properties
meeting_id	int	Primary Key Foreign Key



Column Name	Data Type	Properties
student_id	int	Primary Key Foreign Key
attendance	bit	

Zawiera szczegółowe informacje na temat osób biorących udział w spotkaniu

- meeting\_id int - klucz główny, identyfikator spotkania
- student\_id int - identyfikator studenta, zapisanego na spotkanie
- attendance bit - obecność,  
1 - student uczestniczył w spotkaniu,  
0 - student nie uczestniczył w spotkaniu

```
-- Table: MEETING_DETAILS
CREATE TABLE MEETING_DETAILS (
  meeting_id int NOT NULL,
  student_id int NOT NULL,
  attendance bit NOT NULL,
  CONSTRAINT MEETING_DETAILS_pk PRIMARY KEY (meeting_id,student_id)
);
```

Tabela SUBJECT\_MEETINGS

Column Name	Data Type	Properties
subject_id	int	Primary Key Foreign Key
meeting_id	int	Primary Key Foreign Key

Tabela posiada identyfikatory spotkań, które odbywają się w ramach danego przedmiotu wraz z jego identyfikatorem

- subject\_id int - wchodzi w skład klucza głównego, klucz obcy, identyfikator przedmiotu
- meeting\_id int - wchodzi w skład klucza głównego, klucz obcy, identyfikator spotkania

```
-- Table: SUBJECT_MEETINGS
CREATE TABLE SUBJECT_MEETINGS (
  subject_id int NOT NULL,
  meeting_id int NOT NULL,
  CONSTRAINT SUBJECT_MEETINGS_pk PRIMARY KEY (subject_id,meeting_id)
);
```

Tabela MODULE\_MEETINGS

Column Name	Data Type	Properties
module_id	int	Primary Key Foreign Key
meeting_id	int	Primary Key Foreign Key

Tabela posiada identyfikatory spotkań, które odbywają się w ramach danego modułu kursu wraz z jego identyfikatorem

- module\_id int - wchodzi w skład klucza głównego, klucz obcy, identyfikator modułu
- meeting\_id int - wchodzi w skład klucza głównego, klucz obcy, identyfikator spotkania

```
-- Table: MODULE_MEETINGS
CREATE TABLE MODULE_MEETINGS (
  module_id int NOT NULL,
  meeting_id int NOT NULL,
  CONSTRAINT MODULE_MEETINGS_pk PRIMARY KEY (module_id,meeting_id)
);
```

Tabela ASYNC\_MEETINGS

Column Name	Data Type	Properties
meeting_id	int	Primary Key Foreign Key
meeting_url	text	

Zawiera dane dotyczące spotkań internetowych, które nie są na żywo

- meeting\_id int - klucz główny, klucz obcy, identyfikator spotkania
- meeting\_url text - link do spotkania

```
-- Table: ASYNC_MEETINGS
CREATE TABLE ASYNC_MEETINGS (
  meeting_id int NOT NULL,
  meeting_url text NOT NULL,
  CONSTRAINT ASYNC_MEETINGS_pk PRIMARY KEY (meeting_id)
);
```

Tabela SYNC\_MEETINGS

Column Name	Data Type	Properties
meeting_id	int	Primary Key Foreign Key
video_url	text	
meeting_url	text	

Zawiera dane dotyczące spotkań internetowych, które są na żywo

- meeting\_id int - klucz główny, klucz obcy, identyfikator spotkania
- video\_url text nullable - link do zapisu video spotkania
- meeting\_url text - link do spotkania

```
-- Table: SYNC_MEETINGS
CREATE TABLE SYNC_MEETINGS (
  meeting_id int NOT NULL,
  video_url text NULL,
  meeting_url text NOT NULL,
  CONSTRAINT SYNC_MEETINGS_pk PRIMARY KEY (meeting_id)
);
```

STATIONARY\_MEETINGS

Column Name	Data Type	Properties
meeting_id	int	Primary Key Foreign Key
classroom	varchar(10)	

Zawiera dane dotyczące spotkań internetowych, które są stacjonarnie

- meeting\_id int - klucz główny, klucz obcy, identyfikator spotkania
- classroom varchar(10) - numer pokoju, w którym przeprowadzane jest spotkanie

```
-- Table: STATIONARY_MEETINGS
CREATE TABLE STATIONARY_MEETINGS (
  meeting_id int NOT NULL,
  classroom varchar(10) NOT NULL,
  CONSTRAINT STATIONARY_MEETINGS_pk PRIMARY KEY (meeting_id)
);
```

# Dokumentacja kluczy obcych

Table Name	FK Column	Referenced Table	Referenced Column
ASYNC_MEETINGS	meeting_id	MEETINGS	meeting_id
COURSES	course_id	PRODUCTS	product_id
EMPLOYEES	type_id	EMPLOYEE_TYPES	type_id
EMPLOYEES	emploee_id	USERS	user_id
WEBINARS	tutor_id	EMPLOYEES	emploee_id
INTERSHIPS	study_id	STUDIES	study_id
INTERSHIP_DETAILS	internship_id	INTERSHIPS	internship_id
INTERSHIP_DETAILS	student_id	STUDENTS	student_id
MEETING_DETAILS	meeting_id	MEETINGS	meeting_id
MEETING_DETAILS	student_id	STUDENTS	student_id
MEETINGS	tutor_id	EMPLOYEES	emploee_id
MODULE_MEETINGS	meeting_id	MEETINGS	meeting_id
MEETINGS	translator_id	EMPLOYEES	emploee_id
MODULES	course_id	COURSES	course_id
MODULES	tutor_id	EMPLOYEES	emploee_id
MODULE_MEETINGS	module_id	MODULES	module_id
ORDER_DETAILS	order_id	ORDERS	order_id
PAYMENTS	order_id	ORDERS	order_id
PRODUCTS_DETAILS	order_id	ORDERS	order_id
PRODUCTS_DETAILS	product_id	PRODUCTS	product_id
PRODUCTS_DETAILS	student_id	STUDENTS	student_id
PRODUCTS	type_id	PRODUCT_TYPES	type_id
ORDER_DETAILS	product_id	PRODUCTS	product_id
SUBJECTS	subject_id	PRODUCTS	product_id
STATIONARY_MEETINGS	meeting_id	MEETINGS	meeting_id
ORDERS	student_id	STUDENTS	student_id
STUDIES	study_id	PRODUCTS	product_id
SUBJECTS	tutor_id	EMPLOYEES	emploee_id
SUBJECT_MEETINGS	meeting_id	MEETINGS	meeting_id
SUBJECT_MEETINGS	subject_id	SUBJECTS	subject_id

Table Name	FK Column	Referenced Table	Referenced Column
SUBJECTS	study_id	STUDIES	study_id
SYNC_MEETINGS	meeting_id	MEETINGS	meeting_id
STUDENTS	student_id	USERS	user_id
WEBINARS	translator_id	EMPLOYEES	employee_id
WEBINARS	webinar_id	PRODUCTS	product_id