

Contents

| | |
|--|----------|
| Raport 3 | 1 |
| Opis funkcji systemu dla firmy oferującej kursy i szkolenia | 1 |
| Role Użytkowników i Funkcje Systemu | 1 |
| Diagram bazy danych | 3 |
| Kod do generowania bazy danych: | 3 |

Raport 3

Zespół 1

Skład zespołu :

Paweł Czajczyk

Julia Demitraszek

Szymon Rybski

Opis funkcji systemu dla firmy oferującej kursy i szkolenia

Role Użytkowników i Funkcje Systemu

System zarządzający kursami i szkoleniami obsługuje różnorodne formy kształcenia, takie jak webinary, kursy, i studia. Poniżej znajduje się opis funkcji realizowanych przez system oraz zakres uprawnień poszczególnych użytkowników.

1. Role Użytkowników

Administrator

- Zarządzanie wszystkimi danymi w systemie, w tym tworzenie, edycja i usuwanie kursów, webinarów oraz studiów.
- Zarządzanie wszystkimi użytkownikami, w tym dodawanie użytkowników i ich usuwanie.
- Przeglądanie i generowanie wszystkich dostępnych raportów.
- Ustawianie wyjątków płatności, np. zgody na płatności odroczone(dla stałych użytkowników) - wyjątek musi być zaakceptowany przez dyrektora.

Wykładowca

- Zarządzanie harmonogramem oraz treściami kursów, webinarów oraz studiów, które prowadzi.
- Przeglądanie listy uczestników swoich zajęć i zarządzanie ich obecnością.
- Przeglądanie raportów związanych z prowadzonymi zajęciami, np. frekwencja uczestników.
- Generowanie list obecności oraz raportów dotyczących frekwencji.
- Przygotowanie sylabusu oraz harmonogramu zajęć dla studiów.
- Możliwość oznaczenia odrobionych przez studenta zajęć

Uczestnik (Student)

- Rejestracja na kursy, webinary oraz studia.
- Przeglądanie harmonogramów zapisanych kursów, webinarów oraz studiów.
- Rejestracja na płatne pojedyncze spotkania w ramach studiów
- Opłacanie kursów, webinarów oraz studiów poprzez zintegrowany system płatności.
- Uczestnictwo w kursach online, oglądanie nagrań z kursów asynchronicznych, uczestnictwo w kursach stacjonarnych.
- Przeglądanie ocen oraz dyplomów uzyskanych po zakończeniu kursów i studiów.
- Możliwość zgłoszenia odrobienia nieobecności - do zatwierdzenia przez wykładowcę
- Dodawanie wydarzeń, kursów, studiów i webinarów do koszyka i zarządzanie przedmiotami w koszyku
- Dostęp do nagrań z webinarów przez 30 dni z możliwością wykupienia nagrań do zakończonych
- Możliwość wystąpienia do dyrektora o zniżkę dla stałych klientów
- Przeglądanie dotyczących użytkownika raportów bilokacji
- Dostęp o informacji o swojej frekwencji i zaległych płatności

Tłumacz

- Przeglądanie harmonogramów zajęć, do których został przypisany.
- Udział w tłumaczeniu na żywo webinarów oraz kursów.
- Wgląd do harmonogramu

Dyrektor Szkoły

- Zarządzanie wyjątkami płatnościowymi (np. odroczenie płatności).
- Przeglądanie wszystkich raportów, w tym finansowych oraz frekwencji.
- Zarządzanie dostępnością kursów i webinarów, decydowanie o limitach miejsc.
- Tworzenie i edytowanie kursów, webinarów oraz studiów
- Możliwość dodawania użytkowników i ich usuwania
- Możliwość edytowania sylabusu
- Możliwość spoglądania na listę dłużników
- Generowanie raportu

2. Funkcje Systemu i Uprawnienia

Funkcje ogólne

- Rejestracja i logowanie użytkowników - Każdy użytkownik może założyć konto w systemie, logować się i edytować swoje dane profilowe.
- Przeglądanie harmonogramu - Uczestnicy mogą przeglądać harmonogram swoich zajęć i dostosowywać dostępność do kursów.
- Generowanie dyplomów - Po ukończeniu kursu lub studiów, system automatycznie generuje certyfikat/dyplom, który jest wysyłany Poczta Polska na adres korespondencyjny.
- Dodawanie kursów, webinarów i kierunków studiów
- Usuwanie możliwości dostępu do webinarów po 30 dniach

Płatności

- Obsługa płatności - System integruje się z zewnętrznym systemem płatności, umożliwiając płatności online za kursy, webinary oraz studia.
- Generowanie koszyka - Uczestnik może dodawać produkty do koszyka, a następnie generowany jest link do płatności.
- Śledzenie płatności - System śledzi statusy płatności (udane/nieudane) i aktualizuje dostęp uczestników do kursów i webinarów.

Raportowanie i analiza

- Generowanie raportów finansowych - Administrator i Dyrektor Szkoły mają dostęp do raportów finansowych dla webinarów, kursów oraz studiów.
- Raporty dotyczące frekwencji - Wykładowca może generować raporty z frekwencji dla prowadzonych zajęć.
- Lista dłużników - Administrator ma dostęp do listy uczestników, którzy nie dokonali wymaganych płatności.
- Raport bilokacji - System automatycznie wykrywa uczestników zapisanych na kolidujące ze sobą szkolenia.

Uczestnictwo i zarządzanie

- Rejestracja na kursy - Uczestnicy mogą się rejestrować na kursy, webinary oraz studia, przy czym system zarządza dostępnością miejsc.
- Możliwość wykreślenia się z listy kursantów/ studentów
- Przydzielanie wykładowców i tłumaczy - Administrator przydziela wykładowców i tłumaczy do poszczególnych wydarzeń.
- Ustawienia dostępności nagrań - Administrator i wykładowca mogą decydować, czy nagranie jest dostępne (płatne/darmowe) i przez jaki okres.
- Ustawianie limitów miejsc dla kursów hybrydowych i stacjonarnych, czy studiów
- Zmienianie limitów i możliwość dodawania osób z zewnątrz
- Frekwencja i zaliczenia - System automatycznie monitoruje obecność uczestników i sprawdza, czy spełniają wymagania (np. 80% frekwencji).

Zarządzanie treścią

- Zarządzanie modułami kursów - Wykładowcy mają możliwość tworzenia, edytowania i usuwania modułów kursów.
- Dodawanie nagrań i materiałów edukacyjnych - Wykładowcy mogą dodawać materiały edukacyjne i nagrania do swoich kursów.
- Monitorowanie postępów - System śledzi postępy uczestników w zaliczaniu modułów, zarówno online, jak i stacjonarnych.

Diagram bazy danych

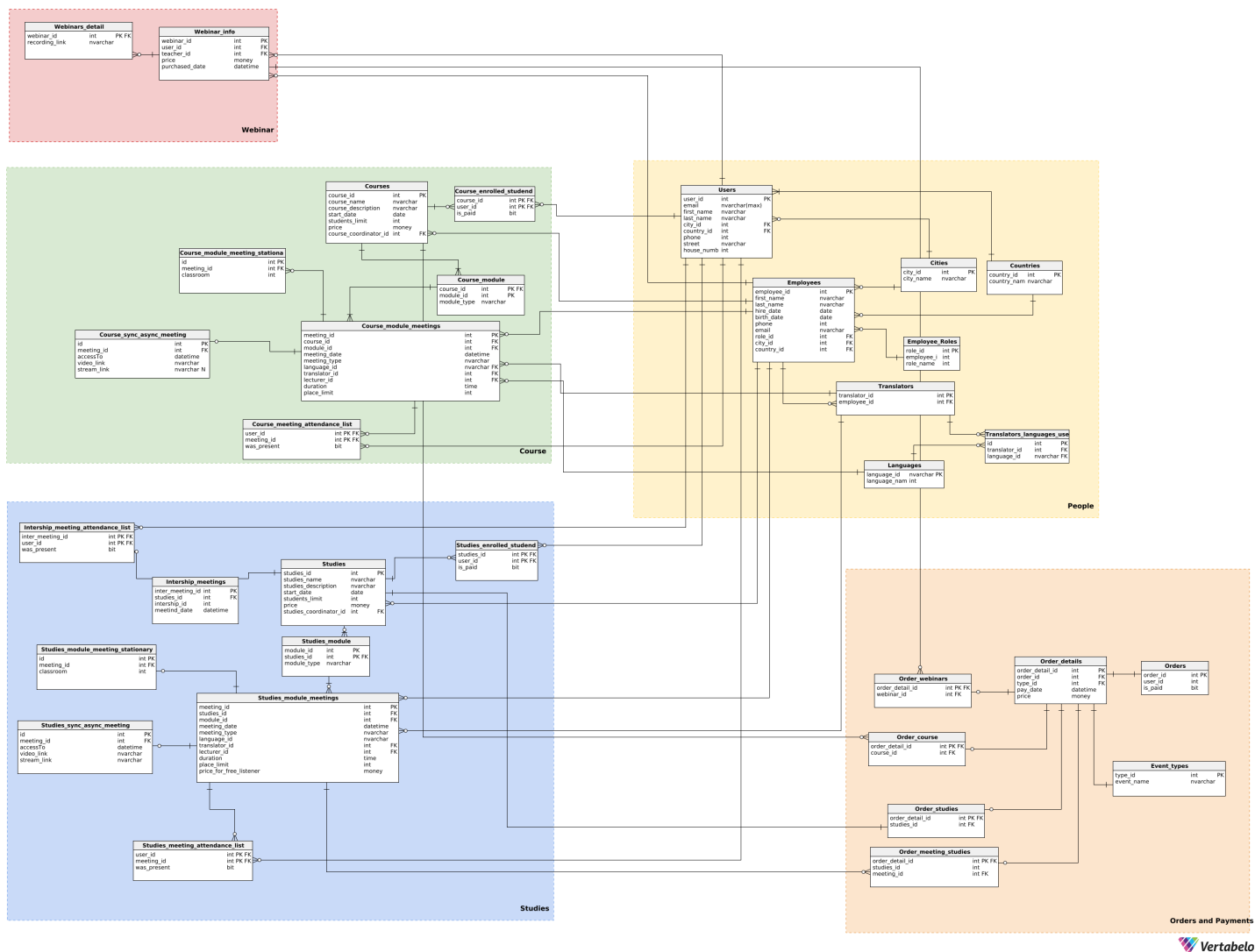


Figure 1: diagram

diagram w wersji svg

Kod do generowania bazy danych:

```
-- Created by Vertabelo (http://vertabelo.com)
-- Last modification date: 2024-12-09 14:33:15.629

-- tables
-- Table: Cities
CREATE TABLE Cities (
  city_id int NOT NULL,
  city_name nvarchar NOT NULL,
  CONSTRAINT Cities_pk PRIMARY KEY (city_id)
);

-- Table: Countries
CREATE TABLE Countries (
  country_id int NOT NULL,
  country_name nvarchar NOT NULL,
  CONSTRAINT Countries_pk PRIMARY KEY (country_id)
);

-- Table: Course_enrolled_students
CREATE TABLE Course_enrolled_students (
```

```

course_id int NOT NULL,
user_id int NOT NULL,
is_paid bit NOT NULL,
order_id int NOT NULL,
CONSTRAINT Course_enrolled_students_pk PRIMARY KEY (course_id,user_id)
);

-- Table: Course_meeting_attendance_list
CREATE TABLE Course_meeting_attendance_list (
    user_id int NOT NULL,
    course_id int NOT NULL,
    meeting_id int NOT NULL,
    was_present bit NOT NULL,
    CONSTRAINT Course_meeting_attendance_list_pk PRIMARY KEY (user_id,meeting_id,course_id)
);

-- Table: Course_module
CREATE TABLE Course_module (
    course_id int NOT NULL,
    meeting_id int NOT NULL,
    module_type nvarchar NOT NULL,
    CONSTRAINT Course_module_pk PRIMARY KEY (course_id,meeting_id)
);

-- Table: Course_module_meeting_stationary
CREATE TABLE Course_module_meeting_stationary (
    id int NOT NULL,
    course_id int NOT NULL,
    meeting_id int NOT NULL,
    classroom int NOT NULL,
    CONSTRAINT Course_module_meeting_stationary_pk PRIMARY KEY (id)
);

-- Table: Course_module_meetings
CREATE TABLE Course_module_meetings (
    course_id int NOT NULL,
    meeting_id int NOT NULL,
    meeting_date datetime NOT NULL,
    meeting_type nvarchar NOT NULL,
    language_id nvarchar NOT NULL,
    translator_id int NOT NULL,
    lecturer_id int NOT NULL,
    duration time NOT NULL,
    place_limit int NOT NULL,
    CONSTRAINT Course_module_meetings_pk PRIMARY KEY (meeting_id,course_id)
);

-- Table: Course_sync_async_meeting
CREATE TABLE Course_sync_async_meeting (
    id int NOT NULL,
    course_id int NOT NULL,
    meeting_id int NOT NULL,
    accessTo datetime NOT NULL,
    video_link nvarchar NOT NULL,
    stream_link nvarchar NULL,
    CONSTRAINT Course_sync_async_meeting_pk PRIMARY KEY (id)
);

-- Table: Courses
CREATE TABLE Courses (
    course_id int NOT NULL,
    course_name nvarchar NOT NULL,
    course_description nvarchar NOT NULL,
    start_date date NOT NULL,
    students_limit int NOT NULL,

```

```

    price money NOT NULL,
    course_coordinator_id int NOT NULL,
    CONSTRAINT Courses_pk PRIMARY KEY (course_id)
);

-- Table: Employee_Roles
CREATE TABLE Employee_Roles (
    role_id int NOT NULL,
    employee_id int NOT NULL,
    role_name int NOT NULL,
    CONSTRAINT Employee_Roles_pk PRIMARY KEY (role_id)
);

-- Table: Employees
CREATE TABLE Employees (
    employee_id int NOT NULL,
    first_name nvarchar NOT NULL,
    last_name nvarchar NOT NULL,
    hire_date date NOT NULL,
    birth_date date NOT NULL,
    phone int NOT NULL,
    email nvarchar NOT NULL,
    role_id int NOT NULL,
    city_id int NOT NULL,
    country_id int NOT NULL,
    CONSTRAINT Employees_pk PRIMARY KEY (employee_id)
);

-- Table: Event_types
CREATE TABLE Event_types (
    type_id int NOT NULL,
    event_name nvarchar NOT NULL,
    CONSTRAINT Event_types_pk PRIMARY KEY (type_id)
);

-- Table: Internship_meeting_attendance_list
CREATE TABLE Internship_meeting_attendance_list (
    inter_meeting_id int NOT NULL,
    user_id int NOT NULL,
    was_present bit NOT NULL,
    CONSTRAINT Internship_meeting_attendance_list_pk PRIMARY KEY (inter_meeting_id,user_id)
);

-- Table: Internship_meetings
CREATE TABLE Internship_meetings (
    inter_meeting_id int NOT NULL,
    studies_id int NOT NULL,
    internship_id int NOT NULL,
    meetind_date datetime NOT NULL,
    CONSTRAINT Internship_meetings_pk PRIMARY KEY (inter_meeting_id)
);

-- Table: Languages
CREATE TABLE Languages (
    language_id nvarchar NOT NULL,
    language_name int NOT NULL,
    CONSTRAINT Languages_pk PRIMARY KEY (language_id)
);

-- Table: Order_course
CREATE TABLE Order_course (
    order_detail_id int NOT NULL,
    course_id int NOT NULL,
    CONSTRAINT Order_course_pk PRIMARY KEY (order_detail_id)
);

```

```

-- Table: Order_details
CREATE TABLE Order_details (
    order_detail_id int NOT NULL,
    order_id int NOT NULL,
    type_id int NOT NULL,
    CONSTRAINT Order_details_pk PRIMARY KEY (order_detail_id)
);

-- Table: Order_meeting_studies
CREATE TABLE Order_meeting_studies (
    order_detail_id int NOT NULL,
    studies_id int NOT NULL,
    meeting_id int NOT NULL,
    CONSTRAINT Order_meeting_studies_pk PRIMARY KEY (order_detail_id)
);

-- Table: Order_studies
CREATE TABLE Order_studies (
    order_detail_id int NOT NULL,
    studies_id int NOT NULL,
    CONSTRAINT Order_studies_pk PRIMARY KEY (order_detail_id)
);

-- Table: Order_webinars
CREATE TABLE Order_webinars (
    order_detail_id int NOT NULL,
    webinar_id int NOT NULL,
    CONSTRAINT Order_webinars_pk PRIMARY KEY (order_detail_id)
);

-- Table: Orders
CREATE TABLE Orders (
    order_id int NOT NULL,
    user_id int NOT NULL,
    is_paid bit NOT NULL,
    price money NOT NULL,
    CONSTRAINT Orders_pk PRIMARY KEY (order_id)
);

-- Table: Studies
CREATE TABLE Studies (
    studies_id int NOT NULL,
    studies_name nvarchar NOT NULL,
    studies_description nvarchar NOT NULL,
    start_date date NOT NULL,
    students_limit int NOT NULL,
    price money NOT NULL,
    studies_coordinator_id int NOT NULL,
    CONSTRAINT Studies_pk PRIMARY KEY (studies_id)
);

-- Table: Studies_enrolled_students
CREATE TABLE Studies_enrolled_students (
    studies_id int NOT NULL,
    user_id int NOT NULL,
    only_meeting bit NOT NULL,
    order_id int NOT NULL,
    CONSTRAINT Studies_enrolled_students_pk PRIMARY KEY (studies_id,user_id)
);

-- Table: Studies_meeting_attendance_list
CREATE TABLE Studies_meeting_attendance_list (
    user_id int NOT NULL,
    studies_id int NOT NULL,

```

```

meeting_id int NOT NULL,
was_present bit NOT NULL,
CONSTRAINT Studies_meeting_attendance_list_pk PRIMARY KEY (user_id,meeting_id,studies_id)
);

-- Table: Studies_module
CREATE TABLE Studies_module (
    studies_id int NOT NULL,
    meeting_id int NOT NULL,
    module_type nvarchar NOT NULL,
    CONSTRAINT Studies_module_pk PRIMARY KEY (studies_id,meeting_id)
);

-- Table: Studies_module_meeting_stationary
CREATE TABLE Studies_module_meeting_stationary (
    id int NOT NULL,
    studies_id int NOT NULL,
    meeting_id int NOT NULL,
    classroom int NOT NULL,
    CONSTRAINT Studies_module_meeting_stationary_pk PRIMARY KEY (id)
);

-- Table: Studies_module_meetings
CREATE TABLE Studies_module_meetings (
    meeting_id int NOT NULL,
    studies_id int NOT NULL,
    meeting_date datetime NOT NULL,
    meeting_type nvarchar NOT NULL,
    language_id nvarchar NOT NULL,
    translator_id int NOT NULL,
    lecturer_id int NOT NULL,
    duration time NOT NULL,
    place_limit int NOT NULL,
    price_for_free_listener money NOT NULL,
    CONSTRAINT Studies_module_meetings_pk PRIMARY KEY (meeting_id,studies_id)
);

-- Table: Studies_sync_async_meeting
CREATE TABLE Studies_sync_async_meeting (
    id int NOT NULL,
    studies_id int NOT NULL,
    meeting_id int NOT NULL,
    accessTo datetime NOT NULL,
    video_link nvarchar NOT NULL,
    stream_link nvarchar NOT NULL,
    CONSTRAINT Studies_sync_async_meeting_pk PRIMARY KEY (id)
);

-- Table: Translators
CREATE TABLE Translators (
    translator_id int NOT NULL,
    employee_id int NOT NULL,
    CONSTRAINT Translators_pk PRIMARY KEY (translator_id)
);

-- Table: Translators_languages_used
CREATE TABLE Translators_languages_used (
    id int NOT NULL,
    translator_id int NOT NULL,
    language_id nvarchar NOT NULL,
    CONSTRAINT Translators_languages_used_pk PRIMARY KEY (id)
);

-- Table: Users
CREATE TABLE Users (

```

```

user_id int NOT NULL,
email nvarchar(max) NOT NULL,
first_name nvarchar NOT NULL,
last_name nvarchar NOT NULL,
city_id int NOT NULL,
country_id int NOT NULL,
phone int NOT NULL,
street nvarchar NOT NULL,
house_number int NOT NULL,
CONSTRAINT Users_pk PRIMARY KEY (user_id)
);

-- Table: Webinar_enrolled_students
CREATE TABLE Webinar_enrolled_students (
    webinar_id int NOT NULL,
    user_id int NOT NULL,
    order_id int NOT NULL,
    purashed_date money NOT NULL,
    CONSTRAINT Webinar_enrolled_students_pk PRIMARY KEY (webinar_id,user_id)
);

-- Table: Webinar_info
CREATE TABLE Webinar_info (
    webinar_id int NOT NULL,
    user_id int NOT NULL,
    teacher_id int NOT NULL,
    price money NOT NULL,
    CONSTRAINT Webinar_info_pk PRIMARY KEY (webinar_id)
);

-- Table: Webinars_detail
CREATE TABLE Webinars_detail (
    webinar_id int NOT NULL,
    recording_link nvarchar NOT NULL,
    CONSTRAINT Webinars_detail_pk PRIMARY KEY (webinar_id)
);

-- Table: enrolled_for_meeting
CREATE TABLE enrolled_for_meeting (
    studies_id int NOT NULL,
    meeting_id int NOT NULL,
    user_id int NOT NULL,
    CONSTRAINT enrolled_for_meeting_pk PRIMARY KEY (studies_id,meeting_id,user_id)
);

-- foreign keys
-- Reference: Cities_Users (table: Users)
ALTER TABLE Users ADD CONSTRAINT Cities_Users
    FOREIGN KEY (city_id)
    REFERENCES Cities (city_id);

-- Reference: Countries_Users (table: Users)
ALTER TABLE Users ADD CONSTRAINT Countries_Users
    FOREIGN KEY (country_id)
    REFERENCES Countries (country_id);

-- Reference: Course_enrolled_studends_Courses (table: Course_enrolled_studends)
ALTER TABLE Course_enrolled_studends ADD CONSTRAINT Course_enrolled_studends_Courses
    FOREIGN KEY (course_id)
    REFERENCES Courses (course_id);

-- Reference: Course_enrolled_studends_Orders (table: Course_enrolled_studends)
ALTER TABLE Course_enrolled_studends ADD CONSTRAINT Course_enrolled_studends_Orders
    FOREIGN KEY (order_id)
    REFERENCES Orders (order_id);

```



```

-- Reference: Course_enrolled_students_Users (table: Course_enrolled_students)
ALTER TABLE Course_enrolled_students ADD CONSTRAINT Course_enrolled_students_Users
    FOREIGN KEY (user_id)
    REFERENCES Users (user_id);

-- Reference: Course_meeting_attendance_list_Course_module_meetings (table: Course_meeting_attendance_list)
ALTER TABLE Course_meeting_attendance_list ADD CONSTRAINT Course_meeting_attendance_list_Course_module_meetings
    FOREIGN KEY (meeting_id,course_id)
    REFERENCES Course_module_meetings (meeting_id,course_id);

-- Reference: Course_meeting_attendance_list_Users (table: Course_meeting_attendance_list)
ALTER TABLE Course_meeting_attendance_list ADD CONSTRAINT Course_meeting_attendance_list_Users
    FOREIGN KEY (user_id)
    REFERENCES Users (user_id);

-- Reference: Course_module_Course_module_meetings (table: Course_module)
ALTER TABLE Course_module ADD CONSTRAINT Course_module_Course_module_meetings
    FOREIGN KEY (meeting_id,course_id)
    REFERENCES Course_module_meetings (meeting_id,course_id);

-- Reference: Course_module_meetings_Course_module_meeting_stationary (table: Course_module_meeting_stationary)
ALTER TABLE Course_module_meeting_stationary ADD CONSTRAINT Course_module_meetings_Course_module_meeting_stationary
    FOREIGN KEY (meeting_id,course_id)
    REFERENCES Course_module_meetings (meeting_id,course_id);

-- Reference: Course_module_meetings_Courses (table: Course_module_meetings)
ALTER TABLE Course_module_meetings ADD CONSTRAINT Course_module_meetings_Courses
    FOREIGN KEY (course_id)
    REFERENCES Courses (course_id);

-- Reference: Course_module_meetings_Employees (table: Course_module_meetings)
ALTER TABLE Course_module_meetings ADD CONSTRAINT Course_module_meetings_Employees
    FOREIGN KEY (lecturer_id)
    REFERENCES Employees (employee_id);

-- Reference: Course_module_meetings_Languages (table: Course_module_meetings)
ALTER TABLE Course_module_meetings ADD CONSTRAINT Course_module_meetings_Languages
    FOREIGN KEY (language_id)
    REFERENCES Languages (language_id);

-- Reference: Course_video_access_Course_module_meetings (table: Course_sync_async_meeting)
ALTER TABLE Course_sync_async_meeting ADD CONSTRAINT Course_video_access_Course_module_meetings
    FOREIGN KEY (meeting_id,course_id)
    REFERENCES Course_module_meetings (meeting_id,course_id);

-- Reference: Courses_Employees (table: Courses)
ALTER TABLE Courses ADD CONSTRAINT Courses_Employees
    FOREIGN KEY (course_coordinator_id)
    REFERENCES Employees (employee_id);

-- Reference: Employees_Cities (table: Employees)
ALTER TABLE Employees ADD CONSTRAINT Employees_Cities
    FOREIGN KEY (city_id)
    REFERENCES Cities (city_id);

-- Reference: Employees_Countries (table: Employees)
ALTER TABLE Employees ADD CONSTRAINT Employees_Countries
    FOREIGN KEY (country_id)
    REFERENCES Countries (country_id);

-- Reference: Employees_Employee_Roles (table: Employees)
ALTER TABLE Employees ADD CONSTRAINT Employees_Employee_Roles
    FOREIGN KEY (role_id)
    REFERENCES Employee_Roles (role_id);

```

```

-- Reference: Internship_meeting_attendance_list_Internship_meetings (table: Internship_meeting_attendance_list)
ALTER TABLE Internship_meeting_attendance_list ADD CONSTRAINT Internship_meeting_attendance_list_Internship_meetings
    FOREIGN KEY (inter_meeting_id)
    REFERENCES Internship_meetings (inter_meeting_id);

-- Reference: Internship_meeting_attendance_list_Users (table: Internship_meeting_attendance_list)
ALTER TABLE Internship_meeting_attendance_list ADD CONSTRAINT Internship_meeting_attendance_list_Users
    FOREIGN KEY (user_id)
    REFERENCES Users (user_id);

-- Reference: Internship_meetings_Studies (table: Internship_meetings)
ALTER TABLE Internship_meetings ADD CONSTRAINT Internship_meetings_Studies
    FOREIGN KEY (studies_id)
    REFERENCES Studies (studies_id);

-- Reference: Order_course_Courses (table: Order_course)
ALTER TABLE Order_course ADD CONSTRAINT Order_course_Courses
    FOREIGN KEY (course_id)
    REFERENCES Courses (course_id);

-- Reference: Order_course_Order_details (table: Order_course)
ALTER TABLE Order_course ADD CONSTRAINT Order_course_Order_details
    FOREIGN KEY (order_detail_id)
    REFERENCES Order_details (order_detail_id);

-- Reference: Order_details_Event_types (table: Order_details)
ALTER TABLE Order_details ADD CONSTRAINT Order_details_Event_types
    FOREIGN KEY (type_id)
    REFERENCES Event_types (type_id);

-- Reference: Order_details_Orders (table: Order_details)
ALTER TABLE Order_details ADD CONSTRAINT Order_details_Orders
    FOREIGN KEY (order_id)
    REFERENCES Orders (order_id);

-- Reference: Order_meeting_studies_Order_details (table: Order_meeting_studies)
ALTER TABLE Order_meeting_studies ADD CONSTRAINT Order_meeting_studies_Order_details
    FOREIGN KEY (order_detail_id)
    REFERENCES Order_details (order_detail_id);

-- Reference: Order_meeting_studies_Studies_module_meetings (table: Order_meeting_studies)
ALTER TABLE Order_meeting_studies ADD CONSTRAINT Order_meeting_studies_Studies_module_meetings
    FOREIGN KEY (meeting_id,studies_id)
    REFERENCES Studies_module_meetings (meeting_id,studies_id);

-- Reference: Order_studies_Order_details (table: Order_studies)
ALTER TABLE Order_studies ADD CONSTRAINT Order_studies_Order_details
    FOREIGN KEY (order_detail_id)
    REFERENCES Order_details (order_detail_id);

-- Reference: Order_studies_Studies (table: Order_studies)
ALTER TABLE Order_studies ADD CONSTRAINT Order_studies_Studies
    FOREIGN KEY (studies_id)
    REFERENCES Studies (studies_id);

-- Reference: Order_webinars_Order_details (table: Order_webinars)
ALTER TABLE Order_webinars ADD CONSTRAINT Order_webinars_Order_details
    FOREIGN KEY (order_detail_id)
    REFERENCES Order_details (order_detail_id);

-- Reference: Order_webinars_Webinar_info (table: Order_webinars)
ALTER TABLE Order_webinars ADD CONSTRAINT Order_webinars_Webinar_info
    FOREIGN KEY (webinar_id)
    REFERENCES Webinar_info (webinar_id);

```

```

-- Reference: Studies_Employees (table: Studies)
ALTER TABLE Studies ADD CONSTRAINT Studies_Employees
    FOREIGN KEY (studies_coordinator_id)
    REFERENCES Employees (employee_id);

-- Reference: Studies_enrolled_students_Orders (table: Studies_enrolled_students)
ALTER TABLE Studies_enrolled_students ADD CONSTRAINT Studies_enrolled_students_Orders
    FOREIGN KEY (order_id)
    REFERENCES Orders (order_id);

-- Reference: Studies_enrolled_students_Studies (table: Studies_enrolled_students)
ALTER TABLE Studies_enrolled_students ADD CONSTRAINT Studies_enrolled_students_Studies
    FOREIGN KEY (studies_id)
    REFERENCES Studies (studies_id);

-- Reference: Studies_enrolled_students_Users (table: Studies_enrolled_students)
ALTER TABLE Studies_enrolled_students ADD CONSTRAINT Studies_enrolled_students_Users
    FOREIGN KEY (user_id)
    REFERENCES Users (user_id);

-- Reference: Studies_meeting_attendance_list_Studies_module_meetings (table: Studies_meeting_attendance_list)
ALTER TABLE Studies_meeting_attendance_list ADD CONSTRAINT Studies_meeting_attendance_list_Studies_module_meetings
    FOREIGN KEY (meeting_id,studies_id)
    REFERENCES Studies_module_meetings (meeting_id,studies_id);

-- Reference: Studies_meeting_attendance_list_Users (table: Studies_meeting_attendance_list)
ALTER TABLE Studies_meeting_attendance_list ADD CONSTRAINT Studies_meeting_attendance_list_Users
    FOREIGN KEY (user_id)
    REFERENCES Users (user_id);

-- Reference: Studies_module_Studies_module_meetings (table: Studies_module)
ALTER TABLE Studies_module ADD CONSTRAINT Studies_module_Studies_module_meetings
    FOREIGN KEY (meeting_id,studies_id)
    REFERENCES Studies_module_meetings (meeting_id,studies_id);

-- Reference: Studies_module_meeting_stationary_Studies_module_meetings (table: Studies_module_meeting_stationary)
ALTER TABLE Studies_module_meeting_stationary ADD CONSTRAINT Studies_module_meeting_stationary_Studies_module_meetings
    FOREIGN KEY (meeting_id,studies_id)
    REFERENCES Studies_module_meetings (meeting_id,studies_id);

-- Reference: Studies_module_meetings_Employees (table: Studies_module_meetings)
ALTER TABLE Studies_module_meetings ADD CONSTRAINT Studies_module_meetings_Employees
    FOREIGN KEY (lecturer_id)
    REFERENCES Employees (employee_id);

-- Reference: Studies_module_meetings_Studies (table: Studies_module_meetings)
ALTER TABLE Studies_module_meetings ADD CONSTRAINT Studies_module_meetings_Studies
    FOREIGN KEY (studies_id)
    REFERENCES Studies (studies_id);

-- Reference: Studies_module_meetings_Translators (table: Studies_module_meetings)
ALTER TABLE Studies_module_meetings ADD CONSTRAINT Studies_module_meetings_Translators
    FOREIGN KEY (translator_id)
    REFERENCES Translators (translator_id);

-- Reference: Studies_sync_async_meeting_Studies_module_meetings (table: Studies_sync_async_meeting)
ALTER TABLE Studies_sync_async_meeting ADD CONSTRAINT Studies_sync_async_meeting_Studies_module_meetings
    FOREIGN KEY (meeting_id,studies_id)
    REFERENCES Studies_module_meetings (meeting_id,studies_id);

-- Reference: Translators_Course_module_meetings (table: Course_module_meetings)
ALTER TABLE Course_module_meetings ADD CONSTRAINT Translators_Course_module_meetings
    FOREIGN KEY (translator_id)
    REFERENCES Translators (translator_id);

```

```

-- Reference: Translators_Employees (table: Translators)
ALTER TABLE Translators ADD CONSTRAINT Translators_Employees
    FOREIGN KEY (employee_id)
    REFERENCES Employees (employee_id);

-- Reference: Translators_languages_used_Languages (table: Translators_languages_used)
ALTER TABLE Translators_languages_used ADD CONSTRAINT Translators_languages_used_Languages
    FOREIGN KEY (language_id)
    REFERENCES Languages (language_id);

-- Reference: Translators_languages_used_Translators (table: Translators_languages_used)
ALTER TABLE Translators_languages_used ADD CONSTRAINT Translators_languages_used_Translators
    FOREIGN KEY (translator_id)
    REFERENCES Translators (translator_id);

-- Reference: Webinar_acces_Users (table: Webinar_info)
ALTER TABLE Webinar_info ADD CONSTRAINT Webinar_acces_Users
    FOREIGN KEY (user_id)
    REFERENCES Users (user_id);

-- Reference: Webinar_enrolled_students_Orders (table: Webinar_enrolled_students)
ALTER TABLE Webinar_enrolled_students ADD CONSTRAINT Webinar_enrolled_students_Orders
    FOREIGN KEY (order_id)
    REFERENCES Orders (order_id);

-- Reference: Webinar_enrolled_students_Webinar_info (table: Webinar_enrolled_students)
ALTER TABLE Webinar_enrolled_students ADD CONSTRAINT Webinar_enrolled_students_Webinar_info
    FOREIGN KEY (webinar_id)
    REFERENCES Webinar_info (webinar_id);

-- Reference: Webinar_info_Employees (table: Webinar_info)
ALTER TABLE Webinar_info ADD CONSTRAINT Webinar_info_Employees
    FOREIGN KEY (teacher_id)
    REFERENCES Employees (employee_id);

-- Reference: Webinars_detail_Webinar_info (table: Webinars_detail)
ALTER TABLE Webinars_detail ADD CONSTRAINT Webinars_detail_Webinar_info
    FOREIGN KEY (webinar_id)
    REFERENCES Webinar_info (webinar_id);

-- Reference: enrolled_for_meeting_Studies_enrolled_students (table: enrolled_for_meeting)
ALTER TABLE enrolled_for_meeting ADD CONSTRAINT enrolled_for_meeting_Studies_enrolled_students
    FOREIGN KEY (studies_id,user_id)
    REFERENCES Studies_enrolled_students (studies_id,user_id);

-- End of file.

```