



AKADEMIA GÓRNICZO HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE

PODSTAWY BAZ DANYCH

PROJEKT:
KURSY I SZKOLENIA

AUTORZY:

JANECZKO TOMASZ

SMYDA TOMASZ

ZIELIŃSKI PRZEMYSŁAW

PROWADZĄCY:

DR INŻ. ROBERT MARCJAN

16 GRUDNIA 2023

Spis treści

1	Wprowadzenie	2
2	Funkcje realizowane przez system	2
2.1	Funkcje użytkowników	2
2.1.1	Administrator	2
2.1.2	Dyrektor	2
2.1.3	Menadżer	2
2.1.4	Nauczyciel	2
2.1.5	Klient	2
2.1.6	Niezarejestrowany użytkownik	3
3	Diagram	3
4	Tabele	4
4.1	Tabela attendance	4
4.2	Tabela employees	4
4.3	Tabela event_lecturers	5
4.4	Tabela event_students	5
4.5	Tabela events	5
4.6	Tabela exams	6
4.7	Tabela languages	6
4.8	Tabela lecturers	7
4.9	Tabela modules	7
4.10	Tabela order_event_details	8
4.11	Tabela order_module_details	8
4.12	Tabela orders	8
4.13	Tabela passing_exams	9
4.14	Tabela passing_practices	9
4.15	Tabela payment_statuses	9
4.16	Tabela practices	10
4.17	Tabela rooms	10
4.18	Tabela single_module_students	10
4.19	Tabela students	11
4.20	Tabela translator_languages	11
4.21	Tabela translators	12
4.22	Tabela types	12

1 Wprowadzenie

Celem projektu było zaplanowanie systemu bazodanowego dla firmy oferującej różnego rodzaju kursy i szkolenia. Początkowo oferowane usługi były świadczone wyłącznie stacjonarnie, ale ze względu na pandemię COVID-19 usługi zostały w różnym stopniu zdigitalizowane. Obecnie model świadczenia usług jest hybrydowy, ale bardzo niejednorodny dla różnych usług. Oferowane usługi dzielą się na webinary, kursy oraz szkolenia.

2 Funkcje realizowane przez system

2.1 Funkcje użytkowników

2.1.1 Administrator

- Dodawanie pracowników
- Obsługa platformy chmurowej (w tym usuwanie nagrań webinarów i kursów)

2.1.2 Dyrektor

- Wydanie zgody na płatność odroczoną w czasie

2.1.3 Menadżer

- Generowanie i przetwarzanie informacji o płatnościach
- Wprowadzenie cen dla płatnych webinarów, kursów oraz studiów
- Tworzenie programu studiów
- Wprowadzanie informacji o kolejnych dniach webinarium
- Wprowadzanie informacji o salach (kursy oraz studia stacjonarne)
- Tworzenie oraz zmienianie harmonogramu zajęć studiów
- Dostęp do raportów bilokacji – listy osób zapisanych na kolidujące się zajęcia
- Dodawanie nauczycieli
- Generowanie raportów finansowych – informacje o płatnościach klientów, zestawienie przychodów dla każdej oferowanej formy szkolenia

2.1.4 Nauczyciel

- Prowadzenie webinarów, kursów oraz studiów
- Dostęp do listy klientów i ich obecności na poszczególnych modułach zajęć

2.1.5 Klient

- Dostęp do darmowego webinarium
- Wykupienie dostępu do płatnego webinarium lub kursu
- Sprawdzanie przypisanych sal (kursy oraz studia stacjonarne)
- Sprawdzenie dotychczasowej frekwencji własnej na studiach
- Dostęp do nagrań (kursy online asynchroniczne)

- Założenie konta

4 Tabele

4.1 Tabela employees

Przechowuje informacje na temat zatrudnionych pracowników.

EmployeeID - Identyfikator pracownika

FirstName - Imię pracownika

LastName - Nazwisko pracownika

Email - Adres email pracownika

Role - Rola pracownika

BirthDate - Data urodzenia pracownika

HireDate - Data zatrudnienia pracownika

Address - Adres pracownika

City - Miasto zamieszkania pracownika

Region - Region zamieszkania pracownika

PostalCode - Kod pocztowy pracownika

Country - Kraj zamieszkania pracownika

Phone - Numer telefonu pracownika

```
01 | CREATE TABLE employees (  
02 |     EmployeeID int NOT NULL,  
03 |     FirstName varchar(30) NOT NULL,  
04 |     LastName varchar(30) NOT NULL,  
05 |     Email varchar(30) NOT NULL,  
06 |     Role varchar(30) NOT NULL,  
07 |     BirthDate date NOT NULL,  
08 |     HireDate date NOT NULL,  
09 |     Address varchar(30) NOT NULL,  
10 |     City varchar(30) NOT NULL,  
11 |     Region varchar(30) NULL,  
12 |     PostalCode varchar(30) NOT NULL,  
13 |     Country varchar(30) NOT NULL,  
14 |     Phone varchar(30) NOT NULL,  
15 |     CONSTRAINT employees_pk PRIMARY KEY (EmployeeID)  
16 | );
```

4.2 Tabela attendance

Przechowuje informacje o obecnościach studentów w danym module. Dzięki tej tabeli jesteśmy w stanie dowiedzieć się czy dany student był obecny na danym module zajęć.

AttendanceID - Identyfikator obecności na module

EventStudentID - Identyfikator studenta (klucz obcy)

ModuleID - Identyfikator modułu (klucz obcy)

IsPresent - Informacja czy student był obecny na zajęciach

```

01 | CREATE TABLE attendance (
02 |     AttendanceID int NOT NULL,
03 |     EventStudentID int NOT NULL,
04 |     ModuleID int NOT NULL,
05 |     IsPresent bit NULL,
06 |     CONSTRAINT attendance_pk PRIMARY KEY (AttendanceID)
07 | );

```

4.3 Tabela event_lecturers

Łączy dane wydarzenie z prowadzącym je wykładowcą.

EventID - Identyfikator wydarzenia

LecturerID - Identyfikator wykładowcy

Description - Opis obowiązków prowadzącego

```

01 | CREATE TABLE event_lecturers (
02 |     EventID int NOT NULL,
03 |     LecturerID int NOT NULL,
04 |     Description varchar(30) NOT NULL,
05 |     CONSTRAINT event_lecturers_pk PRIMARY KEY (EventID, LecturerID)
06 | );

```

4.4 Tabela event_students

Łączy dane wydarzenie z uczestniczącym w nim studentem.

EventStudentID - Identyfikator wydarzenia

StudentID - Identyfikator studenta

EventID - Identyfikator wydarzenia

IsPaidAdvance - Informacja czy zaliczka została wpłacona

IsPaidFull - Informacja czy została zapłacona całkowita kwota

```

01 | CREATE TABLE event_students (
02 |     EventStudentID int NOT NULL,
03 |     StudentID int NOT NULL,
04 |     EventID int NOT NULL,
05 |     IsPaidAdvance bit NOT NULL,
06 |     IsPaidFull bit NOT NULL,
07 |     CONSTRAINT event_students_pk PRIMARY KEY (EventStudentID)
08 | );

```

4.5 Tabela events

Zawiera informacje na temat wydarzeń

EventID - Identyfikator wydarzenia

Title - Tytuł wydarzenia

TypeID - Rodzaj wydarzenia

BeginDate - Data początkowa wydarzenia

EndDate - Data końcowa wydarzenia

Capacity - Limit osób, które mogą uczestniczyć w wydarzeniu

EventPrice - Cena wydarzenia

Description - Opis wydarzenia

```
01 | CREATE TABLE events (  
02 |     EventID int NOT NULL,  
03 |     Title varchar(30) NOT NULL,  
04 |     TypeID int NOT NULL,  
05 |     BeginDate date NOT NULL,  
06 |     EndDate date NOT NULL,  
07 |     Capacity int NOT NULL,  
08 |     EventPrice decimal(10,2) NOT NULL,  
09 |     Description varchar(30) NOT NULL,  
10 |     CONSTRAINT events_pk PRIMARY KEY (EventID)  
11 | );
```

4.6 Tabela exams

Zawiera informacje na temat egzaminów.

ExamID - Identyfikator egzaminu

EventID - Identyfikator wydarzenia, pod który podlega egzamin

BeginDate - Data początkowa egzaminu

EndDate - Data końcowa egzaminu

Description - Opis

```
01 | CREATE TABLE exams (  
02 |     ExamID int NOT NULL,  
03 |     EventID int NOT NULL,  
04 |     BeginDate datetime NOT NULL,  
05 |     EndDate datetime NOT NULL,  
06 |     Description varchar(30) NOT NULL,  
07 |     CONSTRAINT exams_pk PRIMARY KEY (ExamID)  
08 | );
```

4.7 Tabela languages

LanguageID - Identyfikator języka

LanguageName - Nazwa języka

```
01 | CREATE TABLE languages (  
02 |     LanguageID int NOT NULL,  
03 |     LanguageName varchar(30) NOT NULL,  
04 |     CONSTRAINT languages_pk PRIMARY KEY (LanguageID)  
05 | );
```

4.8 Tabela lecturers

LecturerID - Identyfikator wykładowcy

FirstName - Imię wykładowcy

LastName - Nazwisko wykładowcy

Email - Adres email wykładowcy

BirthDate - Data urodzenia wykładowcy

HireDate - Data zatrudnienia wykładowcy

Address - Adres zamieszkania wykładowcy

City - Miasto zamieszkania wykładowcy

Region - Region zamieszkania wykładowcy

PostalCode - Kod pocztowy wykładowcy

Country - Kraj zamieszkania wykładowcy

Phone - Numer telefonu wykładowcy

LanguageID - ID języka wykładowcy, w którym prowadzi wykład

```
01 | CREATE TABLE lecturers (  
02 |     LecturerID int NOT NULL,  
03 |     FirstName varchar(30) NOT NULL,  
04 |     LastName varchar(30) NOT NULL,  
05 |     Email varchar(30) NOT NULL,  
06 |     BirthDate date NOT NULL,  
07 |     HireDate date NOT NULL,  
08 |     Address varchar(30) NOT NULL,  
09 |     City varchar(30) NOT NULL,  
10 |     Region varchar(30) NULL,  
11 |     PostalCode varchar(30) NOT NULL,  
12 |     Country varchar(30) NOT NULL,  
13 |     Phone varchar(30) NOT NULL,  
14 |     LanguageID int NOT NULL,  
15 |     CONSTRAINT lecturers_pk PRIMARY KEY (LecturerID)  
16 | );
```

4.9 Tabela modules

ModuleID - Identyfikator modułu

EventID - Identyfikator wydarzenia, pod który podlega moduł

ModuleNo - Numer modułu

LecturerID - ID wykładowcy prowadzącego moduł

LanguageID - ID języka, w którym moduł jest prowadzony

TranslatorID - ID tłumacza

Description - Opis

BeginDate - Data początkowa modułu

EndDate - Data końcowa modułu

Capacity - Maksymalna liczba osób mogących uczestniczyć w module

RoomID - ID sali, w której odbywa się moduł

ModulePrice - Cena zakupu modułu

```
01 | CREATE TABLE modules (  
02 |     ModuleID int NOT NULL,  
03 |     EventID int NOT NULL,  
04 |     ModuleNo int NOT NULL,  
05 |     LecturerID int NOT NULL,  
06 |     LanguageID int NOT NULL,  
07 |     TranslatorID int NULL,  
08 |     Description varchar(30) NOT NULL,  
09 |     BeginDate datetime NOT NULL,  
10 |     EndDate datetime NOT NULL,  
11 |     Capacity int NOT NULL,  
12 |     RoomID int NOT NULL,  
13 |     ModulePrice decimal(10,2) NOT NULL,  
14 |     CONSTRAINT modules_pk PRIMARY KEY (ModuleID)  
15 | );
```

4.10 Tabela order_event_details

OrderID - Identyfikator zamówienia

EventID - Identyfikator wydarzenia

Price - Cena zakupu wydarzenia

```
01 | CREATE TABLE order_event_details (  
02 |     OrderID int NOT NULL,  
03 |     EventID int NOT NULL,  
04 |     Price decimal(10,2) NOT NULL,  
05 |     CONSTRAINT order_event_details_pk PRIMARY KEY (OrderID,EventID)  
06 | );
```

4.11 Tabela order_module_details

OrderID - Identyfikator zamówienia

ModuleID - Identyfikator modułu

Price - Cena zakupu modułu

```
01 | CREATE TABLE order_module_details (  
02 |     OrderID int NOT NULL,  
03 |     ModuleID int NOT NULL,  
04 |     Price decimal(10,2) NOT NULL,  
05 |     CONSTRAINT order_module_details_pk PRIMARY KEY (OrderID,ModuleID)  
06 | );
```

4.12 Tabela orders

OrderID - Identyfikator zamówienia

StudentID - Identyfikator studenta, który dokonał zamówienia

OrderDate - Data złożenia zamówienia

PaymentStatusID - Identyfikator statusu płatności

PaymentToken - Unikalny token płatności

```
01 | CREATE TABLE orders (  
02 |     OrderID int NOT NULL,  
03 |     StudentID int NOT NULL,  
04 |     OrderDate datetime NOT NULL,  
05 |     PaymentStatusID int NOT NULL,  
06 |     PaymentToken varchar(30) NULL,  
07 |     CONSTRAINT orders_pk PRIMARY KEY (OrderID)  
08 | );
```

4.13 Tabela passing_exams

ExamID - Identyfikator egzaminu

EventStudentID - Identyfikator studenta, który podchodzi do egzaminu

IsPassed - Informacja czy student zdał egzamin

```
01 | CREATE TABLE passing_exams (  
02 |     ExamID int NOT NULL,  
03 |     EventStudentID int NOT NULL,  
04 |     IsPassed int NULL,  
05 |     CONSTRAINT passing_exams_pk PRIMARY KEY (ExamID, EventStudentID)  
06 | );
```

4.14 Tabela passing_practices

PracticeID - Identyfikator praktyk

EventStudentID - Identyfikator studenta, który uczestniczy w praktykach

IsPresent - Informacja czy student był na praktykach

IsPassed - Informacja czy student zaliczył praktyki

```
01 | CREATE TABLE passing_practices (  
02 |     PracticeID int NOT NULL,  
03 |     EventStudentID int NOT NULL,  
04 |     IsPresent bit NULL,  
05 |     IsPassed bit NULL,  
06 |     CONSTRAINT passing_practices_pk PRIMARY KEY (PracticeID, EventStudentID)  
07 | );
```

4.15 Tabela payment_statuses

PaymentStatusID - Identyfikator statusu płatności

PaymentStatusName - Nazwa statusu płatności

```
01 | CREATE TABLE payment_statuses (  
02 |     PaymentStatusID int NOT NULL,  
03 |     PaymentStatusName varchar(30) NOT NULL,  
04 |     CONSTRAINT payment_statuses_pk PRIMARY KEY (PaymentStatusID)  
05 | );
```

4.16 Tabela practices

PracticeID - Identyfikator praktyk

EventID - Identyfikator wydarzenia, pod który podlegają praktyki

PracticeNo - Numer praktyk

Description - Opis

BeginDate - Data rozpoczęcia praktyk

EndDate - Data zakończenia praktyk

RoomID - Identyfikator sali, w której odbywają się praktyki

```
01 | CREATE TABLE practices (  
02 |     PracticeID int NOT NULL,  
03 |     EventID int NOT NULL,  
04 |     PracticeNo int NOT NULL,  
05 |     Description varchar(30) NOT NULL,  
06 |     BeginDate datetime NOT NULL,  
07 |     EndDate datetime NOT NULL,  
08 |     RoomID int NOT NULL,  
09 |     CONSTRAINT practices_pk PRIMARY KEY (PracticeID)  
10 | );
```

4.17 Tabela rooms

RoomID - Identyfikator sali

Name - Nazwa sali

Capacity - Maksymalna liczba osób mogących przebywać w sali

Availability - Dostępność sali

Equipment - Opis wyposażenia sali

```
01 | CREATE TABLE rooms (  
02 |     RoomID int NOT NULL,  
03 |     Name varchar(30) NOT NULL,  
04 |     Capacity int NOT NULL,  
05 |     Availability bit NOT NULL,  
06 |     Equipment varchar(30) NOT NULL,  
07 |     CONSTRAINT rooms_pk PRIMARY KEY (RoomID)  
08 | );
```

4.18 Tabela single_module_students

SingleModuleStudentID - Identyfikator połączenia studenta, który wykupił pojedynczy moduł z tym modulem

ModuleID - Identyfikator modułu

StudentID - Identyfikator studenta

IsPaid - Informacja, czy student opłacił moduł

IsPresent - Informacja, czy student był na module

```

01 | CREATE TABLE single_module_students (
02 |     SingleModuleStudentID int NOT NULL,
03 |     ModuleID int NOT NULL,
04 |     StudentID int NOT NULL,
05 |     IsPaid bit NOT NULL,
06 |     IsPresent bit NULL,
07 |     CONSTRAINT single_module_students_pk PRIMARY KEY (SingleModuleStudentID)
08 | );

```

4.19 Tabela students

StudentID - Identyfikator studenta

FirstName - Imię studenta

LastName - Nazwisko studenta

Email - Adres email studenta

BirthDate - Data urodzenia studenta

Address - Adres zamieszkania studenta

City - Miasto zamieszkania studenta

Region - Region zamieszkania studenta

PostalCode - Kod pocztowy studenta

Country - Kraj zamieszkania studenta

Phone - Numer telefonu studenta

```

01 | CREATE TABLE students (
02 |     StudentID int NOT NULL,
03 |     FirstName varchar(30) NOT NULL,
04 |     LastName varchar(30) NOT NULL,
05 |     Email varchar(30) NOT NULL,
06 |     BirthDate date NOT NULL,
07 |     Address varchar(30) NOT NULL,
08 |     City varchar(30) NOT NULL,
09 |     Region varchar(30) NOT NULL,
10 |     PostalCode varchar(30) NOT NULL,
11 |     Country varchar(30) NOT NULL,
12 |     Phone varchar(30) NOT NULL,
13 |     CONSTRAINT students_pk PRIMARY KEY (StudentID)
14 | );

```

4.20 Tabela translator_languages

```

01 | CREATE TABLE translator_languages (
02 |     TranslatorID int NOT NULL,
03 |     LanguageID int NOT NULL,
04 |     CONSTRAINT translator_languages_pk PRIMARY KEY (TranslatorID,LanguageID)
05 | );

```

4.21 Tabela translators

TranslatorID - Identyfikator tłumacza

FirstName - Imię tłumacza

LastName - Nazwisko tłumacza

Email - Adres email tłumacza

```
01 | CREATE TABLE translators (  
02 |     TranslatorID int NOT NULL,  
03 |     FirstName varchar(30) NOT NULL,  
04 |     LastName varchar(30) NOT NULL,  
05 |     Email varchar(30) NOT NULL,  
06 |     CONSTRAINT translators_pk PRIMARY KEY (TranslatorID)  
07 | );
```

4.22 Tabela types

TypeID - Identyfikator typu

TypeName - Nazwa typu

```
01 | CREATE TABLE types (  
02 |     TypeID int NOT NULL,  
03 |     TypeName varchar(30) NOT NULL,  
04 |     CONSTRAINT types_pk PRIMARY KEY (TypeID)  
05 | );
```