Warunki integralnościowe

Piotr Karamon, Kyrylo Iakymenko, Joanna Konieczny

Contents

1 Roles

```
CREATE TABLE Roles (
    RoleID int  NOT NULL,
    RoleName nvarchar(200)  NOT NULL,
    CONSTRAINT RoleName UNIQUE (RoleName),
    CONSTRAINT employeeType PRIMARY KEY  (RoleID)
)
```

2 EmployeeRoles

```
CREATE TABLE EmployeeRoles (
    EmployeeID int  NOT NULL,
    RoleID int  NOT NULL,
    CONSTRAINT EmployeeRoles_pk PRIMARY KEY  (EmployeeID,RoleID)
)
```

3 Webinars

```
CREATE TABLE Webinars (
    WebinarID int  NOT NULL,
    ReleaseDate datetime  NOT NULL,
    RecordingLink nvarchar(max)  NULL,
    WebinarLink nvarchar(max)  NOT NULL,
    LecturerID int  NOT NULL,
    TranslatorID int  NULL,
    LanguageID int  NOT NULL,
    RecordingReleaseDate date  NULL,
    CONSTRAINT Webinars_pk PRIMARY KEY  (WebinarID)
)
```

Warunki integralnościowe:

- `Webinars_RecodingReleaseDateValid`

  RecodingReleaseData musi być po ReleaseDate

  ```
  CONSTRAINT Webinars_RecodingReleaseDateValid CHECK
  (RecordingReleaseDate >= ReleaseDate)
  ```

- `Webinars_RecodingLinkRelationWithRecordingReleaseDate`

  Jeżeli jest nagranie to RecordingReleaseDate nie może być nullem, jeżeli nie ma to RecordingReleaseDate musi być nullem

  ```
  CONSTRAINT Webinars_RecodingLinkRelationWithRecordingReleaseDate CHECK
  ((RecordingReleaseDate IS NULL AND RecordingLink IS NULL)
  OR
  (RecordingReleaseDate IS NOT NULL AND RecordingLink IS NOT NULL))
  ```

4 Languages

```
CREATE TABLE Languages (
    LanguageID int  NOT NULL,
    LanguageName nvarchar(200)  NOT NULL,
    CONSTRAINT LanguageName UNIQUE (LanguageName),
    CONSTRAINT Languages_pk PRIMARY KEY  (LanguageID)
)
```

5 Courses

```
CREATE TABLE Courses (
    CourseID int  NOT NULL,
    CourseName nvarchar(max)  NOT NULL,
    Description nvarchar(max)  NOT NULL,
    StartDate datetime  NOT NULL,
    EndDate datetime  NOT NULL,
    CoordinatorID int  NOT NULL,
    MaxStudents int  NULL,
    CONSTRAINT Courses_pk PRIMARY KEY  (CourseID)
)
```

Warunki integralnościowe:

- `Course_MaxStudents`

  MaxStudents może być NULL, jeżeli np. jest to kurs wyłącznie online, w przeciwnym wypadku musi być > 0

  ```
  CONSTRAINT Course_MaxStudents CHECK
  (MaxStudents is NULL OR
  (MaxStudents > 0) )
  ```

- `Course_DateIntervalIsValid`

  EndDate musi być po StartDate

  ```
  CONSTRAINT Course_DateIntervalIsValid CHECK
  (StartDate < EndDate)
  ```

6 Modules

```
CREATE TABLE Modules (
    ModuleID int  NOT NULL,
    CourseID int  NOT NULL,
    ModuleName nvarchar(max)  NOT NULL,
    ModuleDescription nvarchar(max)  NOT NULL,
    CONSTRAINT Modules_pk PRIMARY KEY  (ModuleID)
)
```

7 Studies

```
CREATE TABLE Studies (
    StudiesID int  NOT NULL,
    Name int  NOT NULL,
    Description nvarchar(max)  NOT NULL,
    CoordinatorID int  NOT NULL,
    StartDate Date  NOT NULL,
    EndDate Date  NOT NULL,
    MaxStudents int  NOT NULL,
    LanguageID int  NOT NULL,
    Semester int  NOT NULL,
    FieldOfStudiesID int  NOT NULL,
    SemesterNumber int  NOT NULL,
    CONSTRAINT Studies_pk PRIMARY KEY  (StudiesID)
)
```

Warunki integralnościowe:

- `Studies_DateIntervalIsValid`

  EndDate musi być po StartDate

  ```
  CONSTRAINT Studies_DateIntervalIsValid CHECK
  (StartDate < EndDate)
  ```

- `Studies_MaxStudentsIsValid`

  Maksymalna liczba studentów musi być większa od 0

  ```
  CONSTRAINT Studies_MaxStudentsIsValid CHECK
  (MaxStudents > 0)
  ```

- `Studies_SemesterIsValid`

  Numery semestrów zaczynają się od 1

  ```
  CONSTRAINT Studies_SemesterIsValid CHECK
  (SemesterNumber >= 1)
  ```

8 ExamsGrades

```
CREATE TABLE ExamsGrades (
    StudentID int  NOT NULL,
    ExamID int  NOT NULL,
    FinalGrade decimal(2,1)  NOT NULL,
    CONSTRAINT ExamsGrades_pk PRIMARY KEY  (StudentID,ExamID)
)
```

Warunki integralnościowe:

- `FinalExams_FinalGradeIsValid`

  Przyjmujemy skalę ocen jak na publicznej uczelni wyższej

  ```
  CONSTRAINT FinalExams_FinalGradeIsValid CHECK
  (FinalGrade IN (2.0, 3.0, 3.5, 4.0, 4.5, 5.0))
  ```

9 Internships

```
CREATE TABLE Internships (
    InternshipID int  NOT NULL,
    StudiesID int  NOT NULL,
    Description nvarchar(max)  NOT NULL,
    StartDate date  NOT NULL,
    EndDate date  NOT NULL,
    CONSTRAINT Internships_pk PRIMARY KEY  (InternshipID)
)
```

Warunki integralnościowe:

- `Internships_DateIntervalIsValid`

  EndDate musi być po StartDate

  ```
  CONSTRAINT Internships_DateIntervalIsValid CHECK
  (StartDate < EndDate)
  ```

10 InternshipDetails

```
CREATE TABLE InternshipDetails (
    StudentID int  NOT NULL,
    IntershipID int  NOT NULL,
    CompletedAt date  NULL,
    Completed bit  NOT NULL,
    CompanyName nvarchar(500)  NOT NULL,
    City nvarchar(500)  NOT NULL,
    Country nvarchar(500)  NOT NULL,
    PostalCode nvarchar(500)  NOT NULL,
    Address nvarchar(500)  NOT NULL,
    CONSTRAINT InternshipDetails_pk PRIMARY KEY  (IntershipID,StudentID)
)
```

Warunki integralnościowe:

- `InternshipDetails_CompletedAtIsValid`

  CompletedAt nie może być w przyszłości

  ```
  CONSTRAINT InternshipDetails_CompletedAtIsValid CHECK
  (CompletedAt <= GetDate())
  ```

- `InternshipDetails_PostalCodeIsValid`

  Kod pocztowy ma być w prawidłowym formacie

  ```
  CONSTRAINT InternshipDetails_PostalCodeIsValid CHECK
  (PostalCode LIKE '[0-9][0-9]-[0-9][0-9][0-9]'
  OR PostalCode LIKE '[0-9][0-9][0-9][0-9][0-9]'
  OR PostalCode LIKE '[0-9][0-9][0-9][0-9][0-9][0-9]')
  ```

11 Products

```
CREATE TABLE Products (
    ProductID int  NOT NULL,
    Price money  NOT NULL,
    AdvancePayment money  NULL,
    ProductType nvarchar(max)  NOT NULL,
    AddedAt datetime  NOT NULL,
    Closed bit  NOT NULL,
    ClosedAt datetime  NOT NULL,
    CONSTRAINT Products_pk PRIMARY KEY  (ProductID)
)
```

Warunki integralnościowe:

- `Products_PriceIsValid`

  Cena musi być większa lub równa 0

  ```
  CONSTRAINT Products_PriceIsValid CHECK
  (Price >= 0)
  ```

- `Products_AdvancePaymentIsValid`

  Zaliczka musi być większa od 0 i nie może być większa od ceny całkowitej, lub być NULL

  ```
  CONSTRAINT Products_AdvancePaymentIsValid CHECK
  ((AdvancePayment > 0 AND AdvancePayment < Price)
  OR (AdvancePayment IS NULL))
  ```

- `Products_ProductTypeIsValid`

  Produkt może być webinarem, kursem, studia mi, albo pojedyńczym postkaniem studyjnym

  ```
  CONSTRAINT Products_ProductTypeIsValid CHECK
  (ProductType IN ('studies', 'course','webinar', 'public study session'))
  ```

- `Products_AddedAtIsValid`

  AddedAt nie może być w przyszłości

  ```
  CONSTRAINT Products_AddedAtIsValid CHECK
  (AddedAt <= GetDate())
  ```

- `Products_ClosedAtIsValid`

  ClosedAt musi być po AddedAt

  ```
  CONSTRAINT Products_ClosedAtIsValid CHECK
  (ClosedAt <= GetDate() AND ClosedAt >= AddedAt)
  ```

12 Payments

```
CREATE TABLE Payments (
    PaymentID int  NOT NULL,
    UserID int  NOT NULL,
    ProductID int  NOT NULL,
    Price money  NOT NULL,
```

```
    Date date  NOT NULL,
    Status nvarchar(300)  NOT NULL,
    CONSTRAINT Payments_pk PRIMARY KEY  (PaymentID)
)
```

Warunki integralnościowe:

- `Payments_Price`

  Kwota musi być >= 0

  ```
  CONSTRAINT Payments_Price CHECK
  (Price >= 0)
  ```

- `Payments_Status`

  Możliwe wartości dla statusu płatności: "Successful", "Failed"

  ```
  CONSTRAINT Payments_Status CHECK
  (Status in ('Successful', 'Failed'))
  ```

- `Payments_Date`

  Data płatności nie może być z przyszłości

  ```
  CONSTRAINT Payments_Date CHECK
  (Date <= GetDate())
  ```

13 Carts

```
CREATE TABLE Carts (
    UserID int  NOT NULL,
    ProductID int  NOT NULL,
    AddedAt datetime  NOT NULL,
    CONSTRAINT Carts_pk PRIMARY KEY  (UserID,ProductID)
)
```

Warunki integralnościowe:

- `Carts_AddedAtIsValid`

  AddedAt nie może być w przyszłości

  ```
  CONSTRAINT Carts_AddedAtIsValid CHECK
  (AddedAt <= GetDate())
  ```

14 WebinarsAttendence

```
CREATE TABLE WebinarsAttendence (
    WebinarID int  NOT NULL,
    UserID int  NOT NULL,
    CONSTRAINT WebinarsAttendence_pk PRIMARY KEY  (WebinarID,UserID)
)
```

15 StationaryStudiesSessions

```
CREATE TABLE StationaryStudiesSessions (
    StationaryStudiesSessionID int  NOT NULL,
    Address nvarchar(500)  NOT NULL,
    City nvarchar(500)  NOT NULL,
    Country nvarchar(500)  NOT NULL,
    PostalCode nvarchar(20)  NOT NULL,
    ClassroomNumber nvarchar(30)  NOT NULL,
    CONSTRAINT StationaryStudiesSessions_pk PRIMARY KEY  (StationaryStudiesSessionID)
)
```

Warunki integralnościowe:

- `StationaryStudiesSessions_PostalCodeIsValid`

  Kod pocztowy jest w poprawny formacie

  ```
  CONSTRAINT StationaryStudiesSessions_PostalCodeIsValid CHECK
  (PostalCode LIKE '[0-9][0-9]-[0-9][0-9][0-9]'
  OR PostalCode LIKE '[0-9][0-9][0-9][0-9][0-9]'
  OR PostalCode LIKE '[0-9][0-9][0-9][0-9][0-9][0-9]')
  ```

16 OnlineStudiesSessions

```
CREATE TABLE OnlineStudiesSessions (
    OnlineStudiesSessionID int  NOT NULL,
    WebinarLink nvarchar(max)  NOT NULL,
    RecordingLink int  NULL,
    TranslatorID int  NULL,
    CONSTRAINT OnlineStudiesSessions_pk PRIMARY KEY  (OnlineStudiesSessionID)
)
```

17 StudiesSessions

```
CREATE TABLE StudiesSessions (
    StudiesSessionID int  NOT NULL,
    SubjectID int  NOT NULL,
    StartDate datetime  NOT NULL,
    EndDate datetime  NOT NULL,
    LecturerID int  NOT NULL,
    MaxStudents int  NOT NULL,
    TranslatorID int  NULL,
    LanguageID int  NOT NULL,
    CONSTRAINT StudiesSessions_pk PRIMARY KEY  (StudiesSessionID)
)
```

Warunki integralnościowe:

- `StudiesSessions_DateIntervalIsValid`

  EndDate musi być po StartDate

  ```
  CONSTRAINT StudiesSessions_DateIntervalIsValid CHECK
  (StartDate < EndDate)
  ```

- `MaxStudentsVerification`

```
CONSTRAINT MaxStudentsVerification CHECK
(MaxStudents > 0)
```

18 StudiesSessionsAttendence

```
CREATE TABLE StudiesSessionsAttendence (
    SessionID int  NOT NULL,
    StudentID int  NOT NULL,
    Completed bit  NOT NULL,
    CONSTRAINT StudiesSessionsAttendence_pk PRIMARY KEY  (SessionID,StudentID)
)
```

19 CourseOfflineSessions

```
CREATE TABLE CourseOfflineSessions (
    CourseOfflineSessionID int  NOT NULL,
    Link nvarchar(max)  NOT NULL,
    Description nvarchar(max)  NOT NULL,
    UploadedAt datetime  NOT NULL,
    CONSTRAINT CourseOfflineSessions_pk PRIMARY KEY  (CourseOfflineSessionID)
)
```

Warunki integralnościowe:

- `CourseOfflineSessions_UploadedAtIsValid`

  UploadedAt nie może być w przyszłości

```
CONSTRAINT CourseOfflineSessions_UploadedAtIsValid CHECK
(UploadedAt <= GETDATE() )
```

20 CourseStationarySessions

```
CREATE TABLE CourseStationarySessions (
    CourseStationarySessionID int  NOT NULL,
    StartDate datetime  NOT NULL,
    EndDate datetime  NOT NULL,
    Address nvarchar(500)  NOT NULL,
    City nvarchar(500)  NOT NULL,
    Country nvarchar(500)  NOT NULL,
    PostalCode nvarchar(20)  NOT NULL,
    ClassroomNumber nvarchar(30)  NOT NULL,
    MaxStudents int  NOT NULL,
    CONSTRAINT CourseStationarySessions_pk PRIMARY KEY  (CourseStationarySessionID)
)
```

Warunki integralnościowe:

- `CourseStationarySessions_DateIntervalIsValid`

  EndDate musi być po StartDate

```
CONSTRAINT CourseStationarySessions_DateIntervalIsValid CHECK
(StartDate < EndDate)
```

- `CourseStationarySessions_PostalCodeIsValid`

  Kod pocztowy ma być w poprawnym formacie

```
CONSTRAINT CourseStationarySessions_PostalCodeIsValid CHECK
(PostalCode LIKE '[0-9][0-9]-[0-9][0-9][0-9]'
OR PostalCode LIKE '[0-9][0-9][0-9][0-9][0-9]'
OR PostalCode LIKE '[0-9][0-9][0-9][0-9][0-9][0-9]')
```

- `CourseStationarySessions_MaxStudentsIValid`

  MaxStudents musi być większy od zera

```
CONSTRAINT CourseStationarySessions_MaxStudentsIValid CHECK
(MaxStudents > 0)
```

21 CourseOnlineSessions

```
CREATE TABLE CourseOnlineSessions (
    CourseOnlineSessionID int  NOT NULL,
    StartDate datetime  NOT NULL,
    EndDate datetime  NOT NULL,
    WebinarLink text  NOT NULL,
    RecordingLink text  NULL,
    CONSTRAINT CourseOnlineSessions_pk PRIMARY KEY  (CourseOnlineSessionID)
)
```

Warunki integralnościowe:

- `CourseOnlineSessions_DateIntervalCheck`

  EndDate musi być po StartDate

```
CONSTRAINT CourseOnlineSessions_DateIntervalCheck CHECK
(StartDate < EndDate)
```

22 CoursesSessions

```
CREATE TABLE CoursesSessions (
    CourseSessionID int  NOT NULL,
    LanguageID int  NOT NULL,
    ModuleID int  NOT NULL,
    LecturerID int  NOT NULL,
    TranslatorID int  NULL,
    CONSTRAINT CoursesSessions_pk PRIMARY KEY  (CourseSessionID)
)
```

23 CourseSessionsAttendance

```
CREATE TABLE CourseSessionsAttendance (
    CourseParticipantID int  NOT NULL,
    CourseSessionID int  NOT NULL,
    Completed bit  NOT NULL,
    CONSTRAINT CourseSessionsAttendance_pk PRIMARY KEY  (CourseSessionID,CourseParticipantID)
)
```

24 PublicStudySessions

```
CREATE TABLE PublicStudySessions (
    StudiesSessionID int  NOT NULL,
    ProductID int  NOT NULL,
    CONSTRAINT PublicStudySessions_pk PRIMARY KEY  (ProductID,StudiesSessionID)
)
```

25 Students

```
CREATE TABLE Students (
    StudentID int  NOT NULL,
    UserID int  NOT NULL,
    StudiesID int  NOT NULL,
    StudiesPrice money  NOT NULL,
    EntryFee money  NOT NULL,
    DuePostponedPayment datetime  NULL,
    EntryFeePaymentID int  NULL,
    RemainingPaymentID int  NULL,
    CONSTRAINT Students_pk PRIMARY KEY  (StudentID)
)
```

Warunki integralnościowe:

- `Students_PriceIsValid`

  Cena za studia musi być większa od zera

  ```
  CONSTRAINT Students_PriceIsValid CHECK
  (StudiesPrice > 0)
  ```

- `Students_EntryFeeIsValid`

  Zaliczka musi być większa od zera i mniejsza od całkowitej ceny studiów.

  ```
  CONSTRAINT Students_EntryFeeIsValid CHECK
  (EntryFee > 0 AND EntryFee < StudiesPrice)
  ```

26 CourseParticipants

```
CREATE TABLE CourseParticipants (
    CourseParticipantID int  NOT NULL,
    UserID int  NOT NULL,
    CourseID int  NOT NULL,
    CoursePrice money  NOT NULL,
```

```
    EntryFee money  NOT NULL,
    EntryFeePaymentID int  NULL,
    RemainingPaymentID int  NULL,
    FullPricePaymentID int  NULL,
    DuePostponedPayment datetime  NULL,
    CONSTRAINT CourseParticipants_pk PRIMARY KEY  (CourseParticipantID)
)
```

Warunki integralnościowe:

- `CourseParticipants_PriceCheck`

  Cena musi być >=0

  ```
  CONSTRAINT CourseParticipants_PriceCheck CHECK
  (CoursePrice >= 0)
  ```

- `CourseParticpants_EntryFeeCheck`

  Zaliczka nie może być ujemna oraz nie może być większa od całkowitej ceny

  ```
  CONSTRAINT CourseParticpants_EntryFeeCheck CHECK
  (EntryFee >= 0 and EntryFee <= CoursePrice)
  ```

27 RecordingAccessTime

```
CREATE TABLE RecordingAccessTime (
    RecordingAcessTimeID int  NOT NULL,
    StartDate datetime  NOT NULL,
    EndDate datetime  NULL,
    NumberOfDays int  NOT NULL,
    WebinarID int  NULL,
    CONSTRAINT RecordingAccessTime_pk PRIMARY KEY  (RecordingAcessTimeID)
)
```

Warunki integralnościowe:

- `RecordingAccessTime_DateIntervalIsValid`

  EndDate musi być po StartDate

  ```
  CONSTRAINT RecordingAccessTime_DateIntervalIsValid CHECK
  (StartDate < EndDate)
  ```

- `RecordingAccessTime_NumberOfDaysIsValid`

  Liczba dni na którą udostępniamy nagrani musi być większa lub równa 0

  ```
  CONSTRAINT RecordingAccessTime_NumberOfDaysIsValid CHECK
  (NumberOfDays >= 0)
  ```

28 MinAttendancePercentageToPassInternship

```
CREATE TABLE MinAttendancePercentageToPassInternship (
    MinAttendancePercentageToPassInternshipID int  NOT NULL,
    StartDate datetime  NOT NULL,
    EndDate datetime  NULL,
    AttendancePercentage decimal(6,4)  NOT NULL,
    InternshipID int  NULL,
    CONSTRAINT MinAttendancePercentageToPassInternship_pk PRIMARY KEY
    ↪  (MinAttendancePercentageToPassInternshipID)
)
```

Warunki integralnościowe:

- `MinAttendancePercentageToPassInternship_DateIntervalIsValid`

  EndDate musi być po StartDate

  ```
  CONSTRAINT MinAttendancePercentageToPassInternship_DateIntervalIsValid CHECK
  (StartDate < EndDate)
  ```

- `MinAttendancePercentageToPassInternship_PercentageIsValid`

  Procent obecności musi być w przedziale od 0 do 1 włącznie

  ```
  CONSTRAINT MinAttendancePercentageToPassInternship_PercentageIsValid CHECK
  (AttendancePercentage BETWEEN 0 AND 1.0)
  ```

29 MaxDaysForPaymentBeforeStudiesStart

```
CREATE TABLE MaxDaysForPaymentBeforeStudiesStart (
    MaxDaysForPaymentBeforeStudiesStartID int  NOT NULL,
    StartDate datetime  NOT NULL,
    EndDate datetime  NULL,
    NumberOfDays int  NOT NULL,
    StudiesID int  NOT NULL,
    CONSTRAINT MaxDaysForPaymentBeforeStudiesStart_pk PRIMARY KEY
    ↪  (MaxDaysForPaymentBeforeStudiesStartID)
)
```

Warunki integralnościowe:

- `MaxDaysForPaymentBeforeStudiesStart_DateIntervalIsValid`

  EndDate musi być po StartDate

  ```
  CONSTRAINT MaxDaysForPaymentBeforeStudiesStart_DateIntervalIsValid CHECK
  (EndDate > StartDate)
  ```

- `MaxDaysForPaymentBeforeStudiesStart_NumberOfDaysIsValid`

  Liczba dni przed rozpoczęciem musi być większa od 0

  ```
  CONSTRAINT MaxDaysForPaymentBeforeStudiesStart_NumberOfDaysIsValid CHECK
  (NumberOfDays > 0)
  ```

30 MinAttendancePercentageToPassCourse

```
CREATE TABLE MinAttendancePercentageToPassCourse (
    MinAttendancePercentageToPassStudiesID int  NOT NULL,
    StartDate datetime  NOT NULL,
    EndDate datetime  NULL,
    AttendancePercentage decimal(6,4)  NOT NULL,
    CourseID int  NULL,
    CONSTRAINT MinAttendancePercentageToPassCourse_pk PRIMARY KEY
    ↪  (MinAttendancePercentageToPassStudiesID)
)
```

Warunki integralnościowe:

- `MinAttendancePercentageToPassCourse_DateIntervalIsValid`

  EndDate musi być po StartDate

  ```
  CONSTRAINT MinAttendancePercentageToPassCourse_DateIntervalIsValid CHECK
  ((StartDate < EndDate))
  ```

- `MinAttendancePercentageToPassCourse_AttendencePercentageIsValid`

  Procent obecności musi być w przedziale od 0 do 1 włącznie

  ```
  CONSTRAINT MinAttendancePercentageToPassCourse_AttendencePercentageIsValid CHECK
  ((AttendencePercentage >= 0) and (AttendencePercentage <= 1))
  ```

31 DaysInInternship

```
CREATE TABLE DaysInInternship (
    MinAttendancePercentageToPassStudiesID int  NOT NULL,
    StartDate datetime  NOT NULL,
    EndDate datetime  NULL,
    NumberOfDays int  NOT NULL,
    InternshipID int  NULL,
    CONSTRAINT DaysInInternship_pk PRIMARY KEY  (MinAttendancePercentageToPassStudiesID)
)
```

Warunki integralnościowe:

- `DaysInInternship_DateIntervalIsValid`

  EndDate musi być po StartDate

  ```
  CONSTRAINT DaysInInternship_DateIntervalIsValid CHECK
  (StartDate < EndDate)
  ```

- `DaysInInternship_NumberOfDaysIsValid`

  Liczba dni stażu musi być większa od zera

  ```
  CONSTRAINT DaysInInternship_NumberOfDaysIsValid CHECK
  (NumberOfDays > 0)
  ```

32 MaxDaysForPaymentBeforeCourseStart

```
CREATE TABLE MaxDaysForPaymentBeforeCourseStart (
    MaxDaysForPaymentBeforeCourseStartID int  NOT NULL,
    StartDate datetime  NOT NULL,
    EndDate datetime  NULL,
    NumberOfDays int  NOT NULL,
    CourseID int  NULL,
    CONSTRAINT MaxDaysForPaymentBeforeCourseStart_pk PRIMARY KEY
    ↪  (MaxDaysForPaymentBeforeCourseStartID)
)
```

Warunki integralnościowe:

- `MaxDaysForPaymentBeforeCourseStart_DateIntervalIsValid`

  EndDate musi być po StartDate

  ```
  CONSTRAINT MaxDaysForPaymentBeforeCourseStart_DateIntervalIsValid CHECK
  (StartDate < EndDate)
  ```

- `MaxDaysForPaymentBeforeCourseStart_NumberOfDaysIValid`

  Liczba dni przed rozpoczęciem musi być większa od 0

  ```
  CONSTRAINT MaxDaysForPaymentBeforeCourseStart_NumberOfDaysIValid CHECK
  (NumberOfDays > 0)
  ```

33 MinAttendancePercentageToPassStudies

```
CREATE TABLE MinAttendancePercentageToPassStudies (
    MinAttendancePercentageToPassStudiesID int  NOT NULL,
    StartDate datetime  NOT NULL,
    EndDate datetime  NULL,
    AttendancePercentage decimal(6,4)  NOT NULL,
    StudiesID int  NULL,
    CONSTRAINT MinAttendancePercentageToPassStudies_pk PRIMARY KEY
    ↪  (MinAttendancePercentageToPassStudiesID)
)
```

Warunki integralnościowe:

- `MinAttendancePercentageToPassStudies_DateIntervalIsValid`

  EndDate musi być po StartDate

  ```
  CONSTRAINT MinAttendancePercentageToPassStudies_DateIntervalIsValid CHECK
  (StartDate < EndDate)
  ```

- `MinAttendancePercentageToPassStudies_PercentageIsValid`

  Procent obecności musi być w przedziale od 0 do 1 włącznie

  ```
  CONSTRAINT MinAttendancePercentageToPassStudies_PercentageIsValid CHECK
  (AttendancePercentage BETWEEN 0 AND 1)
  ```

34 People

```
CREATE TABLE People (
    PersonID int  NOT NULL,
    FirstName nvarchar(max)  NOT NULL,
    LastName nvarchar(500)  NOT NULL,
    BirthDate date  NOT NULL,
    Address nvarchar(500)  NOT NULL,
    City nvarchar(500)  NOT NULL,
    Region nvarchar(500)  NOT NULL,
    PostalCode nvarchar(20)  NOT NULL,
    Country nvarchar(500)  NOT NULL,
    Phone nvarchar(20)  NOT NULL,
    Email nvarchar(500)  NOT NULL,
    CONSTRAINT id PRIMARY KEY  (PersonID)
)
```

Warunki integralnościowe:

- `People_EmailValid`

  Adres email musi zawierać znak '@'

  ```
  CONSTRAINT People_EmailValid CHECK
  (Email LIKE '%@%')
  ```

- `People_BirthDateValid`

  Data urodzenia nie może być z przyszłości

  ```
  CONSTRAINT People_BirthDateValid CHECK
  (BirithDate <= GetDate())
  ```

- `People_PhoneIsValid`

  Numer telefonu składa się z cyfr

  ```
  CONSTRAINT People_PhoneIsValid CHECK
  ((ISNUMERIC([Phone])=(1)))
  ```

- `People_PostalCodeIsValid`

  Kod pocztowy musi być w poprawnym formacie.

  ```
  CONSTRAINT People_PostalCodeIsValid CHECK
  (PostalCode LIKE '[0-9][0-9]-[0-9][0-9][0-9]'
  OR PostalCode LIKE '[0-9][0-9][0-9][0-9][0-9]'
  OR PostalCode LIKE '[0-9][0-9][0-9][0-9][0-9][0-9]')
  ```

35 PeopleDataChangeHistory

```
CREATE TABLE PeopleDataChangeHistory (
    PersonDataChangeHistoryID int  NOT NULL,
    PersonID int  NOT NULL,
    ChangedAt datetime  NOT NULL,
    New_FirstName nvarchar(max)  NOT NULL,
    Old_FirstName nvarchar(max)  NOT NULL,
    New_LastName nvarchar(500)  NOT NULL,
```

```
    Old_LastName nvarchar(500)  NOT NULL,
    New_BirthDate date  NOT NULL,
    Old_BirthDate date  NOT NULL,
    New_Address nvarchar(500)  NOT NULL,
    Old_Address nvarchar(500)  NOT NULL,
    New_City nvarchar(500)  NOT NULL,
    Old_City nvarchar(500)  NOT NULL,
    New_Region nvarchar(500)  NOT NULL,
    Old_Region nvarchar(500)  NOT NULL,
    New_PostalCode nvarchar(20)  NOT NULL,
    Old_PostalCode nvarchar(500)  NOT NULL,
    New_Country nvarchar(500)  NOT NULL,
    Old_Country nvarchar(500)  NOT NULL,
    New_Email nvarchar(500)  NOT NULL,
    Old_Email nvarchar(500)  NOT NULL,
    New_Phone nvarchar(20)  NOT NULL,
    Old_Phone nvarchar(500)  NOT NULL,
    CONSTRAINT id PRIMARY KEY  (PersonDataChangeHistoryID)
)
```

Warunki integralnościowe:

- `PeopleDataChangeHistory_ChangedAtIsValid`

  Data zmiany nie może być w przyszłości

  ```
  CONSTRAINT PeopleDataChangeHistory_ChangedAtIsValid CHECK
  (ChangedAt <= GetDate())
  ```

- `PeopleDataChangeHistory_NewPostalCodeIsValid`

  Kod pocztowy musi być w poprawnym formacie

  ```
  CONSTRAINT PeopleDataChangeHistory_NewPostalCodeIsValid CHECK
  (New_PostalCode LIKE '[0-9][0-9]-[0-9][0-9][0-9]'
  OR New_PostalCode LIKE '[0-9][0-9][0-9][0-9][0-9]'
  OR New_PostalCode LIKE '[0-9][0-9][0-9][0-9][0-9][0-9]')
  ```

- `PeopleDataChangeHistory_New_EmailValid`

  Nowy adres email musi zawierać znak '@'

  ```
  CONSTRAINT PeopleDataChangeHistory_New_EmailValid CHECK
  (New_Email LIKE '%@%')
  ```

- `PeopleDataChangeHistory_New_BirthDate`

  Data urodzenia nie może być w przyszłości

  ```
  CONSTRAINT PeopleDataChangeHistory_New_BirthDate CHECK
  (New_BirthDate() <= GetDate())
  ```

- `PeopleDataChangeHistory_New_Phone`

  Nowy numer telefonu składa się z cyfr

  ```
  CONSTRAINT PeopleDataChangeHistory_New_Phone CHECK
  (ISNUMERIC([Phone])=(1))
  ```

36 WebinarParticipants

```
CREATE TABLE WebinarParticipants (
    CourseParticipant int  NOT NULL,
    UserID int  NOT NULL,
    WebinarID int  NOT NULL,
    WebinarPrice money  NOT NULL,
    DuePostponedPayment datetime  NULL,
    FullPricePaymentID int  NOT NULL,
    CONSTRAINT WebinarParticipants_pk PRIMARY KEY  (CourseParticipant)
)
```

Warunki integralnościowe:

- `WebinarParticipants_WebinarPrice`

  Cena za webinar musi być większa lub równa zero

  ```
  CONSTRAINT WebinarParticipants_WebinarPrice CHECK
  (WebinarPrice >= 0)
  ```

37 PublicStudySessionParticipants

```
CREATE TABLE PublicStudySessionParticipants (
    PublicStudySessionParticipantID int  NOT NULL,
    UserID int  NOT NULL,
    PublicStudySessionID int  NOT NULL,
    SessionPrice money  NOT NULL,
    DuePostponedPayment datetime  NULL,
    FullPricePaymentID int  NULL,
    CONSTRAINT PublicStudySessionParticipants_pk PRIMARY KEY  (PublicStudySessionParticipantID)
)
```

Warunki integralnościowe:

- `PublicStudySessionParticipants_SessionPriceIsValid`

  ```
  CONSTRAINT PublicStudySessionParticipants_SessionPriceIsValid CHECK
  (SessionPrice > 0)
  ```

38 PublicStudySessionsAttendanceForOutsiders

```
CREATE TABLE PublicStudySessionsAttendanceForOutsiders (
    PublicStudySessionID int  NOT NULL,
    PublicStudySessions_ProductID int  NOT NULL,
    Completed bit  NOT NULL,
    PublicStudySessionParticipantID int  NOT NULL,
    CONSTRAINT PublicStudySessionsAttendanceForOutsiders_pk PRIMARY KEY  (PublicStudySessionID)
)
```

39 SubjectMakeUpPossibilities

```
CREATE TABLE SubjectMakeUpPossibilities (
    SubjectID int  NOT NULL,
    ProductID int  NOT NULL,
    CONSTRAINT SubjectMakeUpPossibilities_pk PRIMARY KEY  (SubjectID,ProductID)
)
```

40 MadeUpAttendance

```
CREATE TABLE MadeUpAttendance (
    SubjectID int  NOT NULL,
    ProductID int  NOT NULL,
    StudentID int  NOT NULL,
    AttendanceValue int  NOT NULL,
    CONSTRAINT MadeUpAttendance_pk PRIMARY KEY  (SubjectID,ProductID)
)
```

Warunki integralnościowe:

- `MadeUpAttendance_AttendanceValue`

  Jest to liczba odrobionych zajęć z przedmiotu, zatem musi być większa od zera

  ```
  CONSTRAINT MadeUpAttendance_AttendanceValue CHECK
  (AttendanceValue > 0)
  ```

41 ProductPriceChangeHistory

```
CREATE TABLE ProductPriceChangeHistory (
    ProductPriceChangeHistoryID int  NOT NULL,
    ProductID int  NOT NULL,
    Old_Price money  NOT NULL,
    New_Price money  NULL,
    Old_AdvancePayment money  NULL,
    New_AdvancePayment money  NULL,
    ChangedAt datetime  NOT NULL,
    CONSTRAINT ProductPriceChangeHistory_pk PRIMARY KEY  (ProductPriceChangeHistoryID)
)
```

Warunki integralnościowe:

- `ProductHistory_ChangedAtIsValid`

  ChangedAt musi być po AddedAt

  ```
  CONSTRAINT ProductHistory_ChangedAtIsValid CHECK
  (ChangedAt <= GetDate() AND ChangedAt >= AddedAt)
  ```

- `ProductHIstory_NewPriceIsValid`

  Nowa cenu musi być nieujemna

  ```
  CONSTRAINT ProductHIstory_NewPriceIsValid CHECK
  (New_price >= 0)
  ```

- `ProductHistory_NewAdvancePaymentIsValid`

  Jeżeli wpisano zaliczkę to musi być ona większa od zera

```
CONSTRAINT ProductHistory_NewAdvancePaymentIsValid CHECK
(New_AdvancePayment > 0)
```

42 CartHistory

```
CREATE TABLE CartHistory (
    CartHistoryID int  NOT NULL,
    UserID int  NOT NULL,
    ProductID int  NOT NULL,
    AddedAt datetime  NOT NULL,
    RemovedAt datetime  NOT NULL,
    CONSTRAINT CartHistory_pk PRIMARY KEY  (CartHistoryID)
)
```

Warunki integralnościowe:

- `CartHistory_AddedAt`

  AddedAt nie może być w przyszłości

```
CONSTRAINT CartHistory_AddedAt CHECK
(AddedAt <= GetDate())
```

- `CartHIstory_RemovedAt`

  RemovedAt musi być po AddedAt

```
CONSTRAINT CartHIstory_RemovedAt CHECK
(RemovedAt >= AddedAt AND RemovedAt <= GetDate())
```

43 Exams

```
CREATE TABLE Exams (
    ExamID int  NOT NULL,
    SubjectID int  NOT NULL,
    StartDate datetime  NOT NULL,
    EndDate datetime  NOT NULL,
    Country nvarchar(500)  NOT NULL,
    City nvarchar(500)  NOT NULL,
    PostalCode nvarchar(500)  NOT NULL,
    Address nvarchar(500)  NOT NULL,
    CONSTRAINT Exams_pk PRIMARY KEY  (ExamID)
)
```

Warunki integralnościowe:

- `Exams_DateInteralIsValid`

  EndDate musi być po StartDate

```
CONSTRAINT Exams_DateInteralIsValid CHECK
(StartDate < EndDate)
```

- `Exams_PostalCodeIsValid`

  Kod pocztowy ma być w poprawnym formacie

```
CONSTRAINT Exams_PostalCodeIsValid CHECK
(PostalCode LIKE '[0-9][0-9]-[0-9][0-9][0-9]'
OR PostalCode LIKE '[0-9][0-9][0-9][0-9][0-9]'
OR PostalCode LIKE '[0-9][0-9][0-9][0-9][0-9][0-9]')
```

44 Subjects

```
CREATE TABLE Subjects (
    SubjectID int  NOT NULL,
    StudiesID int  NOT NULL,
    Description nvarchar(max)  NOT NULL,
    CoordinatorID int  NOT NULL,
    CONSTRAINT SubjectID PRIMARY KEY  (SubjectID)
)
```

45 FieldsOfStudies

```
CREATE TABLE FieldsOfStudies (
    FieldOfStudiesID int  NOT NULL,
    Name nvarchar(max)  NOT NULL,
    Description nvarchar(max)  NOT NULL,
    CONSTRAINT FieldsOfStudies_pk PRIMARY KEY  (FieldOfStudiesID)
)
```

46 Users

```
CREATE TABLE Users (
    UserID int  NOT NULL,
    CONSTRAINT Users_pk PRIMARY KEY  (UserID)
)
```

47 Employees

```
CREATE TABLE Employees (
    EmployeeID int  NOT NULL,
    CONSTRAINT id PRIMARY KEY  (EmployeeID)
)
```