# Wprowadzenie

## Cel

…

## Założenia

..

## Implementacja

..

## Wnioski

…

# Schemat bazy

# Tabele

## Conferences

**CREATE TABLE Conferences (**

**ConferenceID INT NOT NULL IDENTITY,**

**Name VARCHAR(100) NOT NULL,**

**Place VARCHAR(100) NOT NULL,**

**DiscountForStudents FLOAT NOT NULL DEFAULT 0,**

**Description VARCHAR(200) NOT NULL,**

**CONSTRAINT ProperDiscountForStudents CHECK (**

**DiscountForStudents >= 0**

**AND DiscountForStudents <= 100**

**),**

**CONSTRAINT ProperDescription CHECK (LEN(Description) > 5),**

**CONSTRAINT Conferences\_pk PRIMARY KEY (ConferenceID)**

**);**

## ConferenceDays

**CREATE TABLE ConferenceDays (**

**ConferenceDayID INT NOT NULL IDENTITY,**

**Conferences\_ConferenceID INT NOT NULL,**

**DATE DATE NOT NULL,**

**NumberOfParticipants INT NOT NULL,**

**CONSTRAINT PositiveNumberOfConferenceParticipants CHECK (NumberOfParticipants > 0),**

**CONSTRAINT ConferenceDays\_pk PRIMARY KEY (ConferenceDayID)**

**);**

**-- Reference: Conferences\_ConferenceDays (table: ConferenceDays)**

**ALTER TABLE ConferenceDays ADD CONSTRAINT Conferences\_ConferenceDays FOREIGN KEY (Conferences\_ConferenceID) REFERENCES Conferences (ConferenceID)**

## ConferenceCosts

**CREATE TABLE ConferenceCosts (**

**ConferenceCostID INT NOT NULL IDENTITY,**

**Conferences\_ConferenceID INT NOT NULL,**

**Cost DECIMAL(9, 2) NOT NULL,**

**DateFrom DATE NOT NULL,**

**DateTo DATE NOT NULL,**

**CONSTRAINT ProperDayDifferance CHECK (DateFrom <= DateTo),**

**CONSTRAINT NonnegativeCostConf CHECK (Cost >= 0),**

**CONSTRAINT ConferenceCosts\_pk PRIMARY KEY (ConferenceCostID)**

**);**

**-- Reference: Conferences\_ConferenceDay (table: ConferenceCosts)**

**ALTER TABLE ConferenceCosts ADD CONSTRAINT Conferences\_ConferenceDay FOREIGN KEY (Conferences\_ConferenceID) REFERENCES Conferences (ConferenceID)**

## Workshops

**CREATE TABLE Workshops (**

**WorkshopID INT NOT NULL IDENTITY,**

**ConferenceDays\_ConferenceDayID INT NOT NULL,**

**Name VARCHAR(100) NOT NULL,**

**StartTime DATE NOT NULL,**

**EndTime DATE NOT NULL,**

**Cost DECIMAL(9, 2) NOT NULL DEFAULT 0,**

**NumberOfParticipants INT NOT NULL DEFAULT 10,**

**CONSTRAINT NonnegativeCostWorkshop CHECK (Cost >= 0),**

**CONSTRAINT ProperTimeDifferance CHECK (StartTime <= EndTime),**

**CONSTRAINT PosiviteWorkshopParticipants CHECK (NumberOfParticipants > 0),**

**CONSTRAINT Workshops\_pk PRIMARY KEY (WorkshopID)**

**);**

**-- Reference: Workshops\_ConferenceDays (table: Workshops)**

**ALTER TABLE Workshops ADD CONSTRAINT Workshops\_ConferenceDays FOREIGN KEY (ConferenceDays\_ConferenceDayID) REFERENCES ConferenceDays (ConferenceDayID)**

## ConferenceBooking

**CREATE TABLE ConferenceBooking (**

**ConferenceBookingID INT NOT NULL IDENTITY,**

**Clients\_ClientID INT NOT NULL,**

**Conferences\_ConferenceID INT NOT NULL,**

**BookingDate DATE NOT NULL DEFAULT getdate(),**

**IsCanceled BIT NOT NULL DEFAULT 0,**

**CONSTRAINT ConferenceBooking\_pk PRIMARY KEY (ConferenceBookingID)**

**);**

**-- Reference: BookingConference\_Clients (table: ConferenceBooking)**

**ALTER TABLE ConferenceBooking ADD CONSTRAINT BookingConference\_Clients FOREIGN KEY (Clients\_ClientID) REFERENCES Clients (ClientID)**

**-- Reference: BookingConference\_Conferences (table: ConferenceBooking)**

**ALTER TABLE ConferenceBooking ADD CONSTRAINT BookingConference\_Conferences FOREIGN KEY (Conferences\_ConferenceID) REFERENCES Conferences (ConferenceID)**

## ConferenceDayBooking

**CREATE TABLE ConferenceDayBooking (**

**ConferenceDayBookingID INT NOT NULL IDENTITY,**

**ConferenceBooking\_ConferenceBookingID INT NOT NULL,**

**ConferenceDays\_ConferenceDayID INT NOT NULL,**

**BookingDate DATE NOT NULL DEFAULT getdate(),**

**IsCancelled BIT NOT NULL DEFAULT 0,**

**NumberOfParticipants INT NOT NULL,**

**NumberOfStudents INT NOT NULL DEFAULT 0,**

**CONSTRAINT PositiveNumberOfParticipants CHECK (NumberOfParticipants > 0),**

**CONSTRAINT NonnegativeNumberOfStudents CHECK (NumberOfStudents >= 0),**

**CONSTRAINT ProperNumberOfStudents CHECK (NumberOfParticipants >= NumberOfStudents),**

**CONSTRAINT ConferenceDayBooking\_pk PRIMARY KEY (ConferenceDayBookingID)**

**);**

**-- Reference: ConferenceDayBooking\_ConferenceBooking (table: ConferenceDayBooking)**

**ALTER TABLE ConferenceDayBooking ADD CONSTRAINT ConferenceDayBooking\_ConferenceBooking FOREIGN KEY (ConferenceBooking\_ConferenceBookingID) REFERENCES ConferenceBooking (ConferenceBookingID)**

**-- Reference: ConferenceDayBooking\_ConferenceDays (table: ConferenceDayBooking)**

**ALTER TABLE ConferenceDayBooking ADD CONSTRAINT ConferenceDayBooking\_ConferenceDays FOREIGN KEY (ConferenceDays\_ConferenceDayID) REFERENCES ConferenceDays (ConferenceDayID)**

## WorkshopBooking

**CREATE TABLE WorkshopBooking (**

**WorkshopBookingID INT NOT NULL IDENTITY,**

**Workshops\_WorkshopID INT NOT NULL,**

**ConferenceDayBooking\_ConferenceDayBookingID INT NOT NULL,**

**BookingDate DATE NOT NULL DEFAULT getdate(),**

**NumberOfParticipants INT NOT NULL,**

**CONSTRAINT PositiveWorkshopNumberOfParticipants CHECK (NumberOfParticipants > 0),**

**CONSTRAINT WorkshopBooking\_pk PRIMARY KEY (WorkshopBookingID)**

**);**

**-- Reference: WorkshopBooking\_ConferenceDayBooking (table: WorkshopBooking)**

**ALTER TABLE WorkshopBooking ADD CONSTRAINT WorkshopBooking\_ConferenceDayBooking FOREIGN KEY (ConferenceDayBooking\_ConferenceDayBookingID) REFERENCES ConferenceDayBooking (ConferenceDayBookingID)**

**-- Reference: WorkshopBooking\_Workshops (table: WorkshopBooking)**

**ALTER TABLE WorkshopBooking ADD CONSTRAINT WorkshopBooking\_Workshops FOREIGN KEY (Workshops\_WorkshopID) REFERENCES Workshops (WorkshopID)**

## Payments

**CREATE TABLE Payments (**

**PaymentID INT NOT NULL IDENTITY,**

**ConferenceBooking\_ConferenceBookingID INT NOT NULL,**

**Amount DECIMAL(9, 2) NOT NULL,**

**PayDate DATE NOT NULL DEFAULT getdate(),**

**CONSTRAINT PositiveValue CHECK (Amount > 0),**

**CONSTRAINT Payments\_pk PRIMARY KEY (PaymentID)**

**);**

**-- Reference: Payments\_ConferenceBooking (table: Payments)**

**ALTER TABLE Payments ADD CONSTRAINT Payments\_ConferenceBooking FOREIGN KEY (ConferenceBooking\_ConferenceBookingID) REFERENCES ConferenceBooking (ConferenceBookingID)**

## Participants

**CREATE TABLE Participants (**

**ParticipantID INT NOT NULL IDENTITY,**

**FirstName VARCHAR(40) NOT NULL,**

**LastName VARCHAR(40) NOT NULL,**

**Email VARCHAR(100) NOT NULL,**

**Street VARCHAR(40) NOT NULL,**

**City VARCHAR(40) NOT NULL,**

**PostalCode VARCHAR(10) NOT NULL,**

**County VARCHAR(40) NOT NULL,**

**CONSTRAINT Participants\_pk PRIMARY KEY (ParticipantID)**

**);**

## DayParticipants

**CREATE TABLE DayParticipants (**

**DayParticipantID INT NOT NULL IDENTITY,**

**ConferenceDayBooking\_ConferenceDayBookingID INT NOT NULL,**

**Participants\_ParticipantID INT NOT NULL,**

**StudentID VARCHAR(6) NULL DEFAULT NULL,**

**CONSTRAINT ProperStudentID CHECK (**

**StudentID LIKE '^\d{6}$'**

**OR StudentID IS NULL**

**),**

**CONSTRAINT DayParticipants\_pk PRIMARY KEY (DayParticipantID)**

**);**

**-- Reference: DayParticipants\_ConferenceDayBooking (table: DayParticipants)**

**ALTER TABLE DayParticipants ADD CONSTRAINT DayParticipants\_ConferenceDayBooking FOREIGN KEY (ConferenceDayBooking\_ConferenceDayBookingID) REFERENCES ConferenceDayBooking (ConferenceDayBookingID)**

**-- Reference: Participants\_DayParticipants (table: DayParticipants)**

**ALTER TABLE DayParticipants ADD CONSTRAINT Participants\_DayParticipants FOREIGN KEY (Participants\_ParticipantID) REFERENCES Participants (ParticipantID)**

## WorkshopParticipants

**CREATE TABLE WorkshopParticipants (**

**WorkshopParticipantID INT NOT NULL IDENTITY,**

**WorkshopBooking\_WorkshopBookingID INT NOT NULL,**

**DayParticipants\_DayParticipantID INT NOT NULL,**

**CONSTRAINT WorkshopParticipants\_pk PRIMARY KEY (WorkshopParticipantID)**

**);**

**-- Reference: DayParticipants\_WorkshopParticipants (table: WorkshopParticipants)**

**ALTER TABLE WorkshopParticipants ADD CONSTRAINT DayParticipants\_WorkshopParticipants FOREIGN KEY (DayParticipants\_DayParticipantID) REFERENCES DayParticipants (DayParticipantID)**

**-- Reference: WorkshopParticipants\_WorkshopBooking (table: WorkshopParticipants)**

**ALTER TABLE WorkshopParticipants ADD CONSTRAINT WorkshopParticipants\_WorkshopBooking FOREIGN KEY (WorkshopBooking\_WorkshopBookingID) REFERENCES WorkshopBooking (WorkshopBookingID)**

## Clients

**CREATE TABLE Clients (**

**ClientID INT NOT NULL IDENTITY,**

**IsCompany BIT NOT NULL,**

**Name VARCHAR(100) NOT NULL,**

**Surname VARCHAR(100) NULL,**

**Email VARCHAR(100) NOT NULL,**

**CONSTRAINT Clients\_pk PRIMARY KEY (ClientID)**

**);**

# Widoki

## MostPopularWorkshops

**CREATE VIEW view\_MostPopularWorkshops**

**AS**

**SELECT TOP 10 conf.name AS ConferenceName,**

**w.workshopid,**

**w.name,**

**isnull(sum(wp.workshopparticipantid), 0) AS popularnosc**

**FROM Workshops AS w**

**INNER JOIN conferencedays AS cd ON cd.conferencedayid = w.ConferenceDays\_ConferenceDayID**

**INNER JOIN conferences AS conf ON conf.conferenceid = cd.Conferences\_ConferenceID**

**INNER JOIN WorkshopBooking AS wb ON wb.Workshops\_WorkshopID = w.WorkshopID**

**INNER JOIN WorkshopParticipants AS wp ON wp.WorkshopBooking\_WorkshopBookingID = wb.workshopbookingid**

**GROUP BY w.workshopid,**

**w.name,**

**conf.name**

**ORDER BY popularnosc DESC**

## MostPopularConferences

**CREATE VIEW view\_MostPopularConferences**

**AS**

**SELECT TOP 10 c.ConferenceID,**

**c.Name,**

**isnull(sum(dp.DayParticipantID), 0) AS popularity**

**FROM Conferences AS c**

**INNER JOIN ConferenceDays AS cd ON cd.Conferences\_ConferenceID = c.ConferenceID**

**INNER JOIN ConferenceDayBooking AS cdb ON cdb.ConferenceDays\_ConferenceDayID = cd.ConferenceDayID**

**INNER JOIN DayParticipants AS dp ON dp.ConferenceDayBooking\_ConferenceDayBookingID = cdb.ConferenceDayBookingID**

**GROUP BY c.ConferenceID,**

**c.Name**

**ORDER BY popularity DESC**

## MostPopularConferencesByStudents

**CREATE VIEW view\_MostPopularConferencesByStudents**

**AS**

**SELECT TOP 10 c.ConferenceID,**

**c.Name,**

**isnull(sum(dp.DayParticipantID), 0) AS popularity**

**FROM Conferences AS c**

**INNER JOIN ConferenceDays AS cd ON cd.Conferences\_ConferenceID = c.ConferenceID**

**INNER JOIN ConferenceDayBooking AS cdb ON cdb.ConferenceDays\_ConferenceDayID = cd.ConferenceDayID**

**INNER JOIN DayParticipants AS dp ON dp.ConferenceDayBooking\_ConferenceDayBookingID = cdb.ConferenceDayBookingID**

**AND dp.StudentID IS NOT NULL**

**GROUP BY c.ConferenceID,**

**c.Name**

**ORDER BY popularity DESC**

## MostPopularWorkshopsByStudents

**CREATE VIEW view\_MostPopularWorkshopsByStudents**

**AS**

**SELECT TOP 10 conf.name AS ConferenceName,**

**w.workshopid,**

**w.name,**

**isnull(sum(wp.workshopparticipantid), 0) AS popularnosc**

**FROM Workshops AS w**

**INNER JOIN conferencedays AS cd ON cd.conferencedayid = w.ConferenceDays\_ConferenceDayID**

**INNER JOIN conferences AS conf ON conf.conferenceid = cd.Conferences\_ConferenceID**

**INNER JOIN WorkshopBooking AS wb ON wb.Workshops\_WorkshopID = w.WorkshopID**

**INNER JOIN WorkshopParticipants AS wp ON wp.WorkshopBooking\_WorkshopBookingID = wb.workshopbookingid**

**INNER JOIN DayParticipants AS dp ON dp.DayParticipantID = wp.DayParticipants\_DayParticipantID**

**AND dp.StudentID IS NOT NULL**

**GROUP BY w.workshopid,**

**w.name,**

**conf.name**

**ORDER BY popularnosc DESC**

## MostProfitableConference

**CREATE VIEW view\_MostProfitableConference**

**AS**

**SELECT TOP 10 c.ConferenceID,**

**c.Name,**

**cc.Cost + a.koszt AS cena**

**FROM Conferences AS c**

**INNER JOIN ConferenceCosts AS cc ON cc.Conferences\_ConferenceID = c.ConferenceID**

**INNER JOIN (**

**SELECT con.ConferenceID,**

**(**

**SELECT isnull(sum(w.Cost), 0) AS koszta**

**FROM conferencedays AS cd**

**INNER JOIN Workshops AS w ON w.ConferenceDays\_ConferenceDayID = cd.ConferenceDayID**

**WHERE cd.Conferences\_ConferenceID = con.ConferenceID**

**) AS koszt**

**FROM Conferences AS con**

**) AS a ON c.ConferenceID = a.ConferenceID**

**ORDER BY cena**

## MostProfitableWorkshops

**CREATE VIEW view\_MostProfitableWorkshops**

**AS**

**SELECT TOP 10 c.name AS NazwaKonferencji,**

**w.name,**

**w.cost**

**FROM Workshops AS w**

**INNER JOIN ConferenceDays AS cd ON cd.ConferenceDayID = w.ConferenceDays\_ConferenceDayID**

**INNER JOIN Conferences AS c ON c.ConferenceID = cd.Conferences\_ConferenceID**

**ORDER BY w.cost DESC**

## WorkshopsFreePlaces

**CREATE VIEW view\_WorkshopsFreePlaces**

**AS**

**SELECT c.name AS ConferenceName,**

**w.workshopid,**

**w.name,**

**w.numberofparticipants AS Miejsca,**

**isnull(SUM(wp.WorkshopParticipantID), 0) AS Zajete,**

**w.numberofparticipants - isnull(SUM(wb.NumberOfParticipants), 0) AS WolneMiejsca**

**FROM workshops AS w**

**INNER JOIN WorkshopBooking AS wb ON wb.workshops\_workshopid = w.workshopid**

**INNER JOIN WorkshopParticipants AS wp ON wp.WorkshopBooking\_WorkshopBookingID = wb.WorkshopBookingID**

**INNER JOIN conferencedays AS cd ON cd.conferencedayid = w.conferencedays\_conferencedayid**

**INNER JOIN conferences AS c ON c.conferenceid = cd.conferences\_conferenceid**

**GROUP BY w.workshopid,**

**w.name,**

**w.numberofparticipants,**

**c.name**

## ConferenceFreePlaces

**CREATE VIEW view\_ConferenceFreePlaces**

**AS**

**SELECT c.name,**

**cd.DATE,**

**isnull(sum(dp.DayParticipantID), 0) AS Zajete,**

**cdb.NumberOfParticipants AS Miejsca,**

**cdb.NumberOfParticipants - isnull(sum(dp.DayParticipantID), 0) AS WolneMiejsca**

**FROM Conferences AS c**

**INNER JOIN ConferenceDays AS cd ON cd.Conferences\_ConferenceID = c.ConferenceID**

**INNER JOIN ConferenceDayBooking AS cdb ON cdb.ConferenceDays\_ConferenceDayID = cd.ConferenceDayID**

**INNER JOIN DayParticipants AS dp ON dp.ConferenceDayBooking\_ConferenceDayBookingID = cdb.ConferenceDayBookingID**

**GROUP BY c.Name,**

**cd.DATE,**

**cdb.NumberOfParticipants**

## AvailableConferenceDays

**CREATE VIEW view\_AvailableConferenceDays**

**AS**

**SELECT cdp.\***

**FROM VIEW\_ConferenceFreePlaces cdp**

**WHERE WolneMiejsca > 0**

## AvailableWorkshops

**CREATE VIEW view\_AvailableWorkshops**

**AS**

**SELECT w.\***

**FROM view\_WorkshopsFreePlaces w**

**WHERE WolneMiejsca > 0**

## ClientsActivity

**CREATE VIEW view\_ClientsActivity**

**AS**

**SELECT c.\*,**

**isnull((**

**SELECT COUNT(\*)**

**FROM conferencebooking AS cb**

**WHERE cb.clients\_clientid = c.ClientID**

**), 0) AS Bookings,**

**isnull((**

**SELECT SUM(p.amount)**

**FROM conferencebooking AS cb**

**INNER JOIN payments AS p ON cb.ConferenceBookingID = p.ConferenceBooking\_ConferenceBookingID**

**WHERE cb.clients\_clientid = c.ClientID**

**), 0) AS Payments**

**FROM Clients AS c**

## MostProfitableClients

**CREATE VIEW view\_MostProfitableClients**

**AS**

**SELECT TOP 10 c.\***

**FROM view\_ClientsActivity c**

**ORDER BY Payments**

# Procedury

## Dodające

### AddConferenceDay

**CREATE PROCEDURE PROCEDURE\_AddConferenceDay (**

**@ConferenceID INT,**

**@Date DATE,**

**@NumberOfParticipants INT**

**)**

**AS**

**BEGIN**

**INSERT INTO ConferenceDays (**

**Conferences\_ConferenceID,**

**DATE,**

**NumberOfParticipants**

**)**

**VALUES (**

**@ConferenceID,**

**@Date,**

**@NumberOfParticipants**

**);**

**END**

### AddWorkshop

**CREATE PROCEDURE PROCEDURE\_AddWorkshop (**

**@ConferenceDayID INT,**

**@Name VARCHAR(100),**

**@StartTime DATE,**

**@EndTime DATE,**

**@Cost DECIMAL(9, 2),**

**@NumberOfParticipants INT**

**)**

**AS**

**BEGIN**

**INSERT INTO Workshops (**

**ConferenceDays\_ConferenceDayID,**

**Name,**

**StartTime,**

**EndTime,**

**Cost,**

**NumberOfParticipants**

**)**

**VALUES (**

**@ConferenceDayID,**

**@Name,**

**@StartTime,**

**@EndTime,**

**@Cost,**

**@NumberOfParticipants**

**);**

**END**

### AddConferenceCost

**CREATE PROCEDURE PROCEDURE\_AddConferenceCost (**

**@ConferenceID INT,**

**@Cost DECIMAL(9, 2),**

**@DateFrom DATE,**

**@DateTo DATE**

**)**

**AS**

**BEGIN**

**INSERT INTO ConferenceCosts (**

**Conferences\_ConferenceID,**

**Cost,**

**DateFrom,**

**DateTo**

**)**

**VALUES (**

**@ConferenceID,**

**@Cost,**

**@DateFrom,**

**@DateTo**

**);**

**END**

### AddConferenceBooking

**CREATE PROCEDURE PROCEDURE\_AddConferenceBooking (**

**@ConferenceID INT,**

**@ClientID INT,**

**@BookingDate DATE**

**)**

**AS**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM Conferences**

**WHERE @ConferenceID = ConferenceID**

**)**

**BEGIN**

**SELECT 'Nie znaleziono konferencji'**

**END**

**ELSE**

**BEGIN**

**IF @BookingDate IS NULL**

**BEGIN**

**INSERT INTO ConferenceBooking (**

**Conferences\_ConferenceID,**

**Clients\_ClientID**

**)**

**VALUES (**

**@ConferenceID,**

**@ClientID**

**);**

**END**

**ELSE**

**BEGIN**

**INSERT INTO ConferenceBooking (**

**Conferences\_ConferenceID,**

**BookingDate,**

**Clients\_ClientID**

**)**

**VALUES (**

**@ConferenceID,**

**@BookingDate,**

**@ClientID**

**);**

**END**

**END**

**END**

### AddConferenceDayBooking

**CREATE PROCEDURE PROCEDURE\_AddConferenceDayBooking (**

**@ConferenceDayID INT,**

**@ConferenceBookingID INT,**

**@NumberOfParticipants INT,**

**@NumberOfStudents INT**

**)**

**AS**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM ConferenceDays**

**WHERE @ConferenceDayID = ConferenceDayID**

**)**

**OR NOT EXISTS (**

**SELECT \***

**FROM ConferenceBooking**

**WHERE @ConferenceBookingID = ConferenceBookingID**

**)**

**BEGIN**

**SELECT 'Nie znaleziono ConferenceDayID lub ConferenceBookingID'**

**END**

**ELSE**

**BEGIN**

**IF @NumberOfStudents IS NULL**

**BEGIN**

**INSERT INTO ConferenceDayBooking (**

**ConferenceDays\_ConferenceDayID,**

**ConferenceBooking\_ConferenceBookingID,**

**NumberOfParticipants**

**)**

**VALUES (**

**@ConferenceDayID,**

**@ConferenceBookingID,**

**@NumberOfParticipants**

**);**

**END**

**ELSE**

**BEGIN**

**INSERT INTO ConferenceDayBooking (**

**ConferenceDays\_ConferenceDayID,**

**ConferenceBooking\_ConferenceBookingID,**

**NumberOfParticipants,**

**NumberOfStudents**

**)**

**VALUES (**

**@ConferenceDayID,**

**@ConferenceBookingID,**

**@NumberOfParticipants,**

**@NumberOfStudents**

**);**

**END**

**END**

**END**

### AddWorkshopBooking

**CREATE PROCEDURE PROCEDURE\_AddWorkshopBooking (**

**@WorkshopID INT,**

**@ConferenceDayBookingID INT,**

**@BookingDate DATE,**

**@NumberOfParticipants INT**

**)**

**AS**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM Workshops**

**WHERE @WorkshopID = WorkshopID**

**)**

**OR NOT EXISTS (**

**SELECT \***

**FROM ConferenceDayBooking**

**WHERE @ConferenceDayBookingID = ConferenceDayBookingID**

**)**

**BEGIN**

**SELECT 'Nie znaleziono WorkshopID lub ConferenceDayBookingID'**

**END**

**ELSE**

**BEGIN**

**IF @BookingDate IS NULL**

**BEGIN**

**INSERT INTO WorkshopBooking (**

**Workshops\_WorkshopID,**

**ConferenceDayBooking\_ConferenceDayBookingID,**

**NumberOfParticipants**

**)**

**VALUES (**

**@WorkshopID,**

**@ConferenceDayBookingID,**

**@NumberOfParticipants**

**);**

**END**

**ELSE**

**BEGIN**

**INSERT INTO WorkshopBooking (**

**Workshops\_WorkshopID,**

**ConferenceDayBooking\_ConferenceDayBookingID,**

**BookingDate,**

**NumberOfParticipants**

**)**

**VALUES (**

**@WorkshopID,**

**@ConferenceDayBookingID,**

**@BookingDate,**

**@NumberOfParticipants**

**);**

**END**

**END**

**END**

### AddDayParticipant

**CREATE PROCEDURE PROCEDURE\_AddDayParticipant (**

**@ConferenceDayBookingID INT,**

**@ParticipantID INT,**

**@StudentID VARCHAR(6)**

**)**

**AS**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM ConferenceDayBooking**

**WHERE @ConferenceDayBookingID = ConferenceDayBookingID**

**)**

**BEGIN**

**SELECT 'Nie znaleziono ConferenceDayBookingID'**

**END**

**ELSE**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM Participants**

**WHERE @ParticipantID = ParticipantID**

**)**

**BEGIN**

**SELECT 'Nie znaleziono ParticipantID'**

**END**

**ELSE**

**BEGIN**

**INSERT INTO DayParticipants (**

**ConferenceDayBooking\_ConferenceDayBookingID,**

**Participants\_ParticipantID,**

**StudentID**

**)**

**VALUES (**

**@ConferenceDayBookingID,**

**@ParticipantID,**

**@StudentID**

**);**

**END**

**END**

**END**

### AddWorkshopParticipant

**CREATE PROCEDURE PROCEDURE\_AddWorkshopParticipant (**

**@WorkshopBookingID INT,**

**@DayParticipantID INT**

**)**

**AS**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM WorkshopBooking**

**WHERE @WorkshopBookingID = WorkshopBookingID**

**)**

**BEGIN**

**SELECT 'Nie znaleziono WorkshopBookingID'**

**END**

**ELSE**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM Participants**

**WHERE @DayParticipantID = ParticipantID**

**)**

**BEGIN**

**SELECT 'Nie znaleziono ParticipantID'**

**END**

**ELSE**

**BEGIN**

**INSERT INTO WorkshopParticipants (**

**WorkshopBooking\_WorkshopBookingID,**

**DayParticipants\_DayParticipantID**

**)**

**VALUES (**

**@WorkshopBookingID,**

**@DayParticipantID**

**);**

**END**

**END**

**END**

### AddParticipant

**CREATE PROCEDURE PROCEDURE\_AddParticipant (**

**@FirstName VARCHAR(40),**

**@LastName VARCHAR(40),**

**@Email VARCHAR(100),**

**@Street VARCHAR(40),**

**@City VARCHAR(40),**

**@PostalCode VARCHAR(10),**

**@Country VARCHAR(40)**

**)**

**AS**

**BEGIN**

**INSERT INTO Participants (**

**FirstName,**

**LastName,**

**Email,**

**Street,**

**City,**

**PostalCode,**

**County**

**)**

**VALUES (**

**@FirstName,**

**@LastName,**

**@Email,**

**@Street,**

**@City,**

**@PostalCode,**

**@Country**

**);**

**END**

### AddClient

**CREATE PROCEDURE PROCEDURE\_AddClient (**

**@IsCompany BIT,**

**@Name VARCHAR(100),**

**@Surname VARCHAR(100),**

**@Email VARCHAR(100)**

**)**

**AS**

**BEGIN**

**INSERT INTO Clients (**

**IsCompany,**

**Name,**

**Surname,**

**Email**

**)**

**VALUES (**

**@IsCompany,**

**@Name,**

**@Surname,**

**@Email**

**);**

**END**

## Aktualizujące

### RemoveConference

**CREATE PROCEDURE PROCEDURE\_RemoveConference (@ConferenceID INT)**

**AS**

**IF NOT EXISTS (**

**SELECT \***

**FROM Conferences**

**WHERE @ConferenceID = ConferenceID**

**)**

**BEGIN**

**SELECT 'Nie znaleziono konferencji'**

**END**

**ELSE**

**BEGIN**

**DELETE**

**FROM Conferences**

**WHERE Conferences.ConferenceID = @ConferenceID;**

**END**

### UpdateConferenceDetails

**CREATE PROCEDURE PROCEDURE\_UpdateConferenceDetails (**

**@ConferenceID INT,**

**@Name VARCHAR(100),**

**@Place VARCHAR(100),**

**@DiscountForStudents FLOAT,**

**@Description VARCHAR(200)**

**)**

**AS**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM Conferences**

**WHERE @ConferenceID = ConferenceID**

**)**

**BEGIN**

**SELECT 'Nie znaleziono konferencji'**

**END**

**ELSE**

**BEGIN**

**IF @Name IS NOT NULL**

**BEGIN**

**UPDATE Conferences**

**SET Name = @Name**

**WHERE ConferenceID = @ConferenceID;**

**END**

**ELSE**

**BEGIN**

**SELECT 'ConferenceID is null'**

**END**

**IF @DiscountForStudents IS NOT NULL**

**BEGIN**

**UPDATE Conferences**

**SET DiscountForStudents = @DiscountForStudents**

**WHERE ConferenceID = @ConferenceID;**

**END**

**ELSE**

**BEGIN**

**SELECT 'DiscountForStudents is null'**

**END**

**IF @Place IS NOT NULL**

**BEGIN**

**UPDATE Conferences**

**SET Place = @Place**

**WHERE ConferenceID = @ConferenceID;**

**END**

**ELSE**

**BEGIN**

**SELECT 'Place is null'**

**END**

**IF @Description IS NOT NULL**

**BEGIN**

**UPDATE Conferences**

**SET Description = @Description**

**WHERE ConferenceID = @ConferenceID;**

**END**

**ELSE**

**BEGIN**

**SELECT 'Description is null'**

**END**

**END**

**END**

### UpdateWorkshopDetails

**CREATE PROCEDURE PROCEDURE\_UpdateWorkshopDetails (**

**@WorkshopID INT,**

**@Name VARCHAR(100),**

**@StartTime DATE,**

**@EndTime DATE,**

**@Cost DECIMAL(9, 2),**

**@NumberOfParticipants INT**

**)**

**AS**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM Workshops**

**WHERE @WorkshopID = WorkshopID**

**)**

**BEGIN**

**SELECT 'Nie znaleziono warsztatu'**

**END**

**ELSE**

**BEGIN**

**IF @Name IS NOT NULL**

**BEGIN**

**UPDATE Workshops**

**SET Name = @Name**

**WHERE WorkshopID = @WorkshopID;**

**END**

**ELSE**

**BEGIN**

**SELECT 'WorkshopID is null'**

**END**

**IF @StartTime IS NOT NULL**

**BEGIN**

**UPDATE Workshops**

**SET StartTime = @StartTime**

**WHERE WorkshopID = @WorkshopID;**

**END**

**ELSE**

**BEGIN**

**SELECT 'StartTime is null'**

**END**

**IF @EndTime IS NOT NULL**

**BEGIN**

**UPDATE Workshops**

**SET EndTime = @EndTime**

**WHERE WorkshopID = @WorkshopID;**

**END**

**ELSE**

**BEGIN**

**SELECT 'EndTime is null'**

**END**

**IF @Cost IS NOT NULL**

**BEGIN**

**UPDATE Workshops**

**SET Cost = @Cost**

**WHERE WorkshopID = @WorkshopID;**

**END**

**ELSE**

**BEGIN**

**SELECT 'Cost is null'**

**END**

**IF @NumberOfParticipants IS NOT NULL**

**BEGIN**

**UPDATE Workshops**

**SET NumberOfParticipants = @NumberOfParticipants**

**WHERE WorkshopID = @WorkshopID;**

**END**

**ELSE**

**BEGIN**

**SELECT 'NumberOfParticipants is null'**

**END**

**END**

**END**

### CancelConferenceBooking

**CREATE PROCEDURE PROCEDURE\_CancelConferenceBooking (@ConferenceBookingID INT)**

**AS**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM ConferenceBooking**

**WHERE @ConferenceBookingID = ConferenceBookingID**

**)**

**OR NOT (**

**SELECT IsCanceled**

**FROM ConferenceBooking**

**WHERE @ConferenceBookingID = ConferenceBookingID**

**) = 0**

**BEGIN**

**SELECT 'Nie znaleziono ConferenceBookingID lub rezerwacja została już wcześniej anulowana'**

**END**

**ELSE**

**BEGIN**

**UPDATE ConferenceBooking**

**SET IsCanceled = 1**

**WHERE @ConferenceBookingID = ConferenceBookingID;**

**END**

**END**

### CancelConferenceDayBooking

**CREATE PROCEDURE PROCEDURE\_CancelConferenceDayBooking (@ConferenceDayBookingID INT)**

**AS**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM ConferenceDayBooking**

**WHERE @ConferenceDayBookingID = ConferenceDayBookingID**

**)**

**OR NOT (**

**SELECT IsCancelled**

**FROM ConferenceDayBooking**

**WHERE @ConferenceDayBookingID = ConferenceDayBookingID**

**) = 0**

**BEGIN**

**SELECT 'Nie znaleziono ConferenceDayBookingID lub rezerwacja została już wcześniej anulowana'**

**END**

**ELSE**

**BEGIN**

**UPDATE ConferenceDayBooking**

**SET IsCancelled = 1**

**WHERE @ConferenceDayBookingID = ConferenceDayBookingID;**

**END**

**END**

### UpdateWorkShopNumberOfParticipants

**CREATE PROCEDURE PROCEDURE\_UpdateWorkShopNumberOfParticipants (**

**@WorkshopID INT,**

**@NumberOfParticipants INT**

**)**

**AS**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM Workshops**

**WHERE @WorkshopID = WorkshopID**

**)**

**BEGIN**

**SELECT 'Nie znaleziono WorkshopID'**

**END**

**ELSE**

**BEGIN**

**UPDATE Workshops**

**SET NumberOfParticipants = @NumberOfParticipants**

**WHERE @WorkshopID = WorkshopID;**

**END**

**END**

### UpdateConferenceDayNumberOfParticipants

**CREATE PROCEDURE PROCEDURE\_UpdateConferenceDayNumberOfParticipants (**

**@ConferenceDayID INT,**

**@NumberOfParticipants INT**

**)**

**AS**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM ConferenceDays**

**WHERE @ConferenceDayID = ConferenceDayID**

**)**

**BEGIN**

**SELECT 'Nie znaleziono ConferenceDayID'**

**END**

**ELSE**

**BEGIN**

**UPDATE ConferenceDays**

**SET NumberOfParticipants = @NumberOfParticipants**

**WHERE @ConferenceDayID = ConferenceDayID;**

**END**

**END**

### ShowConferenceDaysAmountOfParticipants

**CREATE PROCEDURE PROCEDURE\_ShowConferenceDaysAmountOfParticipants (@ConferenceID INT)**

**AS**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM Conferences**

**WHERE @ConferenceID = ConferenceID**

**)**

**BEGIN**

**SELECT 'Nie znaleziono ConferenceID'**

**END**

**ELSE**

**BEGIN**

**SELECT ConferenceDayID,**

**DATE,**

**SUM(ConferenceDayBooking.NumberOfParticipants) AS Participants,**

**SUM(NumberOfStudents) AS Students**

**FROM ConferenceDays**

**INNER JOIN ConferenceDayBooking ON Conferences\_ConferenceID = ConferenceDays\_ConferenceDayID**

**WHERE Conferences\_ConferenceID = @ConferenceID**

**AND IsCancelled = 0**

**GROUP BY ConferenceDayID,**

**DATE**

**END**

**END**

### ShowListOfEventsOfConference

**CREATE PROCEDURE PROCEDURE\_ShowListOfEventsOfConference (@ConferenceDayID INT)**

**AS**

**BEGIN**

**IF NOT EXISTS (**

**SELECT \***

**FROM ConferenceDays**

**WHERE @ConferenceDayID = ConferenceDayID**

**)**

**BEGIN**

**SELECT 'Nie znaleziono ConferenceDayID'**

**END**

**ELSE**

**BEGIN**

**SELECT WorkshopID,**

**Name,**

**StartTime,**

**EndTime,**

**Cost,**

**NumberOfParticipants**

**FROM Workshops**

**WHERE ConferenceDays\_ConferenceDayID = @ConferenceDayID**

**END**

**END**