

Tadeusz Wiśniewski

Projekt przedstawia bazę danych przewodników turystycznych. Baza danych przechowuje informacje o typach biletów danych klientów danych przewodników oraz rodzajach trasy i dokonanych zamówieniach.

Zdania SQL:

```
CREATE DATABASE ProjektTadeuszWisniewski
```

```
--Przechowuje informacje o możliwych typach biletów
```

```
CREATE TABLE TypyBiletow(  
ID CHAR(1) PRIMARY KEY,  
NazwaTypu NVARCHAR(50) NOT NULL,  
Cena MONEY DEFAULT 10,  
)
```

```
--Przechowuje dane turystów
```

```
CREATE TABLE Turysci(  
ID INT PRIMARY KEY IDENTITY,  
Nazwisko NVARCHAR(50) NOT NULL,  
Imie NVARCHAR(50) NOT NULL,  
Wiek INT NOT NULL,  
CHECK (Wiek BETWEEN 12 AND 99)  
)
```

```
--Przechowuje informacje o poziomach trudności tras (Potrzebne by oraniczyć poziomy do tylko kilku)
```

```
CREATE TABLE PoziomyTrudnosciTras(  
ID CHAR(2) PRIMARY KEY,  
NazwaPoziomu NVARCHAR(50) NOT NULL  
)
```

```
--Przechowuje informacje o dostępnych trasach wycieczek
```

```
CREATE TABLE Trasy(  
ID INT PRIMARY KEY IDENTITY,  
Nazwa NVARCHAR(50) NOT NULL,  
PoziomTrudnosci CHAR(2) REFERENCES PoziomyTrudnosciTras(ID)  
)
```

```
--Przechowuje dane przewodników
```

```
CREATE TABLE Przewodnicy(  
ID INT PRIMARY KEY IDENTITY,  
Nazwisko NVARCHAR(50) NOT NULL,  
Imie NVARCHAR(50) NOT NULL,  
)
```

```
--Przechowuje informację o wolnych przewodnikach (nie miałem pomysłu na trigger)
```

```
CREATE TABLE PrzewodnicyWolni(  
ID INT PRIMARY KEY IDENTITY,  
Przewodnik INT REFERENCES Przewodnicy(ID)  
)
```

```
--Przechowuje informacje o terminach wycieczek
```

```
CREATE TABLE TerminyWycieczek(  
ID INT PRIMARY KEY IDENTITY,  
TerminWycieczki DATE NOT NULL,  
GodzinaWycieczki TIME(0) DEFAULT '12:00:00',  
CHECK (TerminWycieczki BETWEEN GETDATE() AND '2024-01-01')  
)
```

```
--przechowuje dokonane zamowienia
```

```

CREATE TABLE Zamowienia(
Turysta INT REFERENCES Turysci(ID),
Trasa INT REFERENCES Trasy(ID),
Bilet CHAR(1) REFERENCES TypyBiletow(ID),
TerminWycieczki INT REFERENCES TerminyWycieczek(ID),
PRIMARY KEY (Turysta, TerminWycieczki)
)

--uzupelnia tabele WolniPrzewodnicy numerami id przewodnikow
CREATE OR ALTER TRIGGER Tr_WolniPrzewodnicy ON Przewodnicy
AFTER INSERT
AS
    INSERT INTO PrzewodnicyWolni(Przewodnik)
    SELECT I.ID
    FROM INSERTED AS I
GO

--uzupełnienie tabel przykładowymi wartościami
INSERT INTO TypyBiletow VALUES('U','Ulgowy', 10)
INSERT INTO TypyBiletow VALUES('N','Normalny', 20)

INSERT INTO Turysci VALUES('Jan', 'Kowalski', 20)
INSERT INTO Turysci VALUES('Jan2', 'Kowalski2', 22)
INSERT INTO Turysci VALUES('Jan3', 'Kowalski3', 23)

INSERT INTO PoziomyTrudnosciTras VALUES('L','Latwa')
INSERT INTO PoziomyTrudnosciTras VALUES('S','Srednia')
INSERT INTO PoziomyTrudnosciTras VALUES('T','Trudna')

INSERT INTO Trasy VALUES ('Pierwsza', 'L')
INSERT INTO Trasy VALUES ('Druga', 'S')
INSERT INTO Trasy VALUES ('Trzecia', 'T')

INSERT INTO Przewodnicy VALUES('Karol', 'Nowak')
INSERT INTO Przewodnicy VALUES('Karol1', 'Nowak2')
INSERT INTO Przewodnicy VALUES('Karol1', 'Nowak2')

INSERT INTO TerminyWycieczek(TerminWycieczki) VALUES('2023-06-01')
INSERT INTO TerminyWycieczek(TerminWycieczki) VALUES('2023-06-02')
INSERT INTO TerminyWycieczek(TerminWycieczki) VALUES('2023-06-03')

INSERT INTO Zamowienia VALUES(1, 1, 'U', 1)
INSERT INTO Zamowienia VALUES(2, 2, 'N', 2)
INSERT INTO Zamowienia VALUES(3, 3, 'N', 3)

--widok wyświetlający pełne dane zamówień
CREATE OR ALTER VIEW PelneDaneZamowienia
AS
SELECT T.Imie AS ImieKlienta, T.Nazwisko AS NazwiskoKlienta, Tr.Nazwa AS NazwaTrasy,
TW.TerminWycieczki AS TerminWycieczki, TW.GodzinaWycieczki AS GodzinaWycieczki, Z.Bilet
FROM Zamowienia AS Z JOIN Turysci AS T ON Z.Turysta=T.ID JOIN Trasy AS Tr ON Z.Trasa=Tr.ID
JOIN TerminyWycieczek AS TW ON Z.TerminWycieczki=TW.ID

--wiok wyświetlający zamwienia z biletem ulgowym
CREATE OR ALTER VIEW ZamowieniaZBiletemUlgowym
AS
SELECT COUNT(*) AS IloscZamowienZBiletemUlgowym
FROM Zamowienia
WHERE Bilet LIKE 'U'
GROUP BY Bilet

--Widok wyświetlający dane przewodnikow wolnych

```

```
CREATE OR ALTER VIEW PrzewodnicyWolniDane
AS
SELECT PW.ID, P.Imie, P.Nazwisko
FROM PrzewodnicyWolni AS PW JOIN Przewodnicy AS P ON PW.Przewodnik=P.ID
```

```
SELECT * FROM TypyBiletow
SELECT * FROM Turysci
SELECT * FROM PoziomyTrudnosciTras
SELECT * FROM Trasy
SELECT * FROM Przewodnicy
SELECT * FROM PrzewodnicyWolni
SELECT * FROM PrzewodnicyWolniDane
SELECT * FROM TerminyWycieczek
SELECT * FROM PelneDaneZamowienia
SELECT * FROM ZamowieniaZBiletemUlgowym
SELECT * FROM Zamowienia
```

--procedura wyswietlajaca z widoku PelneDaneZamowieniaia tylko te, ktore sa zgodne z argumentem procedury

```
CREATE OR ALTER PROC Proc1 (@DataWycieczki DATE)
AS
SELECT *
FROM PelneDaneZamowienia
WHERE TerminWycieczki = @DataWycieczki
GO
```

```
EXEC Proc1 '2023-06-01'
```

--funkcja wyswietlajaca z widoku PelneDaneZamowieniaia tylko te, ktore sa zgodne z argumentem funkcji

```
CREATE OR ALTER FUNCTION Fun1(@GodzinaWycieczki TIME)
RETURNS TABLE
AS
RETURN(SELECT * FROM PelneDaneZamowienia WHERE GodzinaWycieczki=@GodzinaWycieczki)

SELECT * FROM Fun1('12:00:00')
```