Alan Miziej

Opis projekt bazy danych - uproszczony system bankowy

Celem projektu jest zaprojektowanie bazy danych dla uproszczonego systemu bankowego, który będzie przechowywał i zarządzał informacjami o klientach, kontach bankowych, transakcjach, pracownikach, oddziałach banku oraz kartach płatniczych. Baza danych będzie wykorzystywana do śledzenia i zarządzania operacjami bankowymi, takimi jak otwieranie kont, przeprowadzanie transakcji, wydawanie kart płatniczych, a także monitorowanie danych klientów i pracowników.

CREATE DATABASE BankDB2;

GO

USE BankDB2;

GO

-- Tabela Klienci: przechowywanie danych osobowych klientów

CREATE TABLE Klienci (

KlientID INT PRIMARY KEY IDENTITY,

Imie NVARCHAR(50) NOT NULL,

Nazwisko NVARCHAR(50) NOT NULL,

DataUrodzenia DATE NOT NULL,

Adres NVARCHAR(255) NOT NULL,

Email NVARCHAR(100),

Telefon NVARCHAR(15),

CHECK (DATEDIFF(YEAR, DataUrodzenia, GETDATE()) >= 18)

);

GO

-- Tabela Oddziały: przechowuje informacje na temat oddziałów, gdzie można je znaleźć i jak się skontaktować

CREATE TABLE Oddzialy (

OddzialID INT PRIMARY KEY IDENTITY,

Nazwa NVARCHAR(100) NOT NULL,

Adres NVARCHAR(255) NOT NULL,

Telefon NVARCHAR(15)

);

GO

-- Tabela Pracownicy: dane osobowe pracowników

CREATE TABLE Pracownicy (

PracownikID INT PRIMARY KEY IDENTITY,

Imie NVARCHAR(50) NOT NULL,

Nazwisko NVARCHAR(50) NOT NULL,

OddzialID INT FOREIGN KEY REFERENCES Oddzialy(OddzialID),

Stanowisko NVARCHAR(50) NOT NULL,

DataZatrudnienia DATE NOT NULL

);

GO

-- Tabela Konta: przechowuje informacje o kontach bankowych, celem jest przypisanie konta do klienta

CREATE TABLE Konta (

KontoID INT PRIMARY KEY IDENTITY,

KlientID INT FOREIGN KEY REFERENCES Klienci(KlientID),

NumerKonta NVARCHAR(20) NOT NULL UNIQUE,

Saldo DECIMAL(18, 2) NOT NULL,

TypKonta NVARCHAR(50) NOT NULL,

CHECK (Saldo >= 0)

);

GO

-- Tabela Transakcje: informacje na temat transakcji wykonywanych na danym koncie

CREATE TABLE Transakcje (

TransakcjaID INT PRIMARY KEY IDENTITY,

KontoID INT FOREIGN KEY REFERENCES Konta(KontoID),

DataTransakcji DATETIME NOT NULL,

Kwota DECIMAL(18, 2) NOT NULL,

TypTransakcji NVARCHAR(50) NOT NULL,

Opis NVARCHAR(255)

);

GO

-- Tabela Karty: przechowuje informacje o kartach przypisanych do danego konta

CREATE TABLE Karty (

KartaID INT PRIMARY KEY IDENTITY,

KontoID INT FOREIGN KEY REFERENCES Konta(KontoID),

NumerKarty NVARCHAR(20) NOT NULL UNIQUE,

DataWaznosci DATE NOT NULL,

TypKarty NVARCHAR(50) NOT NULL,

CVV NVARCHAR(4) NOT NULL

);

GO

-- Widok 1: Klienci z saldem powyżej określonej wartości

CREATE VIEW KlienciZDuzymsaldem AS

SELECT k.Imie, k.Nazwisko, k.Email, k.Telefon, ko.Saldo

FROM Klienci k

JOIN Konta ko ON k.KlientID = ko.KlientID

WHERE ko.Saldo > 10000;

GO

-- Widok 2: Sumy transakcji na kontach

CREATE VIEW SumyTransakcji AS

SELECT k.NumerKonta, SUM(t.Kwota) AS SumaTransakcji

FROM Konta k

JOIN Transakcje t ON k.KontoID = t.KontoID

GROUP BY k.NumerKonta;

GO

-- Funkcja: Zwraca wszystkie transakcje powyżej zadanej kwoty

CREATE FUNCTION dbo.TransakcjePowyzejKwoty (

@Kwota DECIMAL(18, 2)

)

RETURNS TABLE

AS

RETURN (

SELECT TransakcjaID, KontoID, DataTransakcji, Kwota, TypTransakcji, Opis

FROM Transakcje

WHERE Kwota > @Kwota

);

GO

-- Procedura składowana: Przeniesienie środków między kontami

CREATE PROCEDURE dbo.PrzeniesSrodki

@KontoZ INT,

@KontoDo INT,

@Kwota DECIMAL(18, 2)

AS

BEGIN

DECLARE @SaldoZ DECIMAL(18, 2);

DECLARE @SaldoDo DECIMAL(18, 2);

-- Sprawdzenie salda konta, z którego są przenoszone środki

SELECT @SaldoZ = Saldo FROM Konta WHERE KontoID = @KontoZ;

SELECT @SaldoDo = Saldo FROM Konta WHERE KontoID = @KontoDo;

IF @SaldoZ >= @Kwota

BEGIN

-- Aktualizacja salda konta z którego są przenoszone środki

UPDATE Konta

SET Saldo = Saldo - @Kwota

WHERE KontoID = @KontoZ;

-- Aktualizacja salda konta, na które są przenoszone środki

UPDATE Konta

SET Saldo = Saldo + @Kwota

WHERE KontoID = @KontoDo;

-- Wstawienie danych transakcji

INSERT INTO Transakcje (KontoID, DataTransakcji, Kwota, TypTransakcji, Opis)

VALUES (@KontoZ, GETDATE(), -@Kwota, 'Przelew', 'Przelew na konto ' + CAST(@KontoDo AS NVARCHAR(20)));

INSERT INTO Transakcje (KontoID, DataTransakcji, Kwota, TypTransakcji, Opis)

VALUES (@KontoDo, GETDATE(), @Kwota, 'Przelew', 'Przelew z konta ' + CAST(@KontoZ AS NVARCHAR(20)));

END

ELSE

BEGIN

RAISERROR ('Niewystarczające środki na koncie.', 16, 1);

END

END;

GO

-- Wyzwalacz: aktualizacja salda konta po dodaniu transakci

CREATE TRIGGER trg\_AktualizacjaSaldPoTransakcji

ON Transakcje

AFTER INSERT

AS

BEGIN

DECLARE @KontoID INT;

DECLARE @Kwota DECIMAL(18, 2);

SELECT @KontoID = i.KontoID, @Kwota = i.Kwota

FROM inserted i;

UPDATE Konta

SET Saldo = Saldo + @Kwota

WHERE KontoID = @KontoID;

END;

GO

Opis widoków:

1. KlienciZDuzymsaldem

Widok KlienciZDuzymsaldem wyświetla informacje o klientach którzy mają saldo na koncie powyżej 10,000.

2. SumyTransakcji

Widok SumyTransakcji grupuje transakcje według numeru konta i oblicza sumę transakcji dla każdego konta.

Diagram bazy danych:

Obraz zawierający tekst, zrzut ekranu, diagram, numer

Opis wygenerowany automatycznie