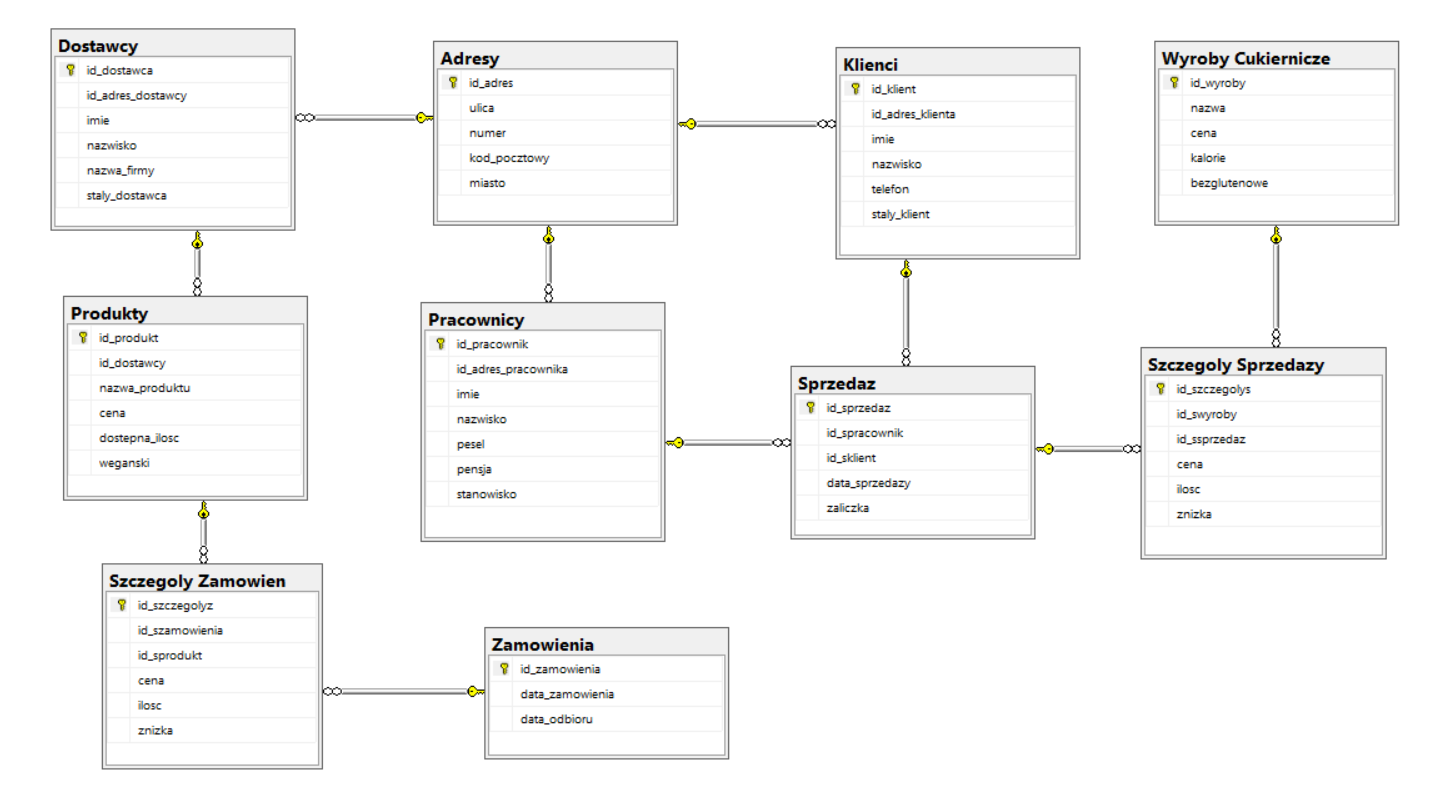
Autor: Inga Nowak nr indeksu 253958

PROJEKT BAZY DANYCH

Schemat bazy danych cukiernia



Opis bazy danych

Projekt przedstawia bazę danych cukierni. Zawiera on informacje dotyczące pracowników cukierni, dostawców zaopatrujących cukiernię w produkty, klientów oraz szczegóły dotyczące sprzedaży i zamówień.

**Omówienie tabel:**

* Adresy – tabela zawiera informacje dotyczące adresów klientów, pracowników oraz dostawców
* Dostawcy – dane osobowe dostawców, nazwa firmy
* Pracownicy – dane osobowe pracowników, kontakt, stanowisko, pensja
* Klienci – dane osobowe klientów, kontakt
* Produkty – nazwa produktów, cena, dostępna ilość, informacja czy są wegański
* Zamówienia – daty zamówienia i odbioru
* Szczegóły Zamówień – cena zamówienia, ilość oraz wysokość zniżki
* Wyroby Cukiernicze – nazwa wyrobów, cena, ilość kalorii, informacja czy są bezglutenowe
* Sprzedaż – data sprzedaży, informacja czy została wpłacona zaliczka
* Szczegóły sprzedaży – cena, ilość sprzedanych produktów, wysokość zniżki

**Omówienie funkcji, widoków, procedur i wyzwalaczy:**

* najwyzsza\_cena – wyświetla informacje o wyrobach cukierniczych o najwyższej cenie
* dodaj\_adres – procedura dodająca wartości do tabeli Adresy
* cena\_ss – uaktualnia wartość ceny w tabeli Szczegóły Sprzedaży
* cena\_sz - uaktualnia wartość ceny w tabeli Szczegóły Zamówień
* zamowienia – wyświetla zamówienia złożone po podanej dacie
* zarobki – zabrania nadawania pensji wartości zero
* ilosc – zabrania zamawiania większej ilości produktów niż jest dostępna
* dostawca – zabrania zamawiania produktów od nieznanego dostawcy
* stanowisko – zabrania sprzedaży pracownikom o stanowisku innym niż kasjer
* znizka – wprowadza zniżkę dla klientów, którzy wpłacili zaliczkę
* produkty\_weganskie – wyświetla informacje o produktach wegańskich
* wyroby\_bezglutenowe – wyświetla informacje o wyrobach bezglutenowych
* staly\_klient – wyświetla informacje o stałych klientach
* staly\_dostawca – wyswietla informacje o stałych dostawcach
* zaliczki – wyświetla informacje o zaliczkach
* zamowione\_produkty – wyświetla informacje o zamówionych produktach
* kalorie – wyświetla informacje o wyrobach posiadających mniejszą kaloryczność niż podana ilość
* adresy\_dostawcow – wyświetla informacje o adresach dostawców
* adresy\_pracowniow – wyświetla informacje o adresach pracowników
* adresy\_klientow – wyświetla informacje o adresach klientów

**Kod SQL:**

1. tabele

CREATE TABLE Adresy(

id\_adres INT IDENTITY(1,1) PRIMARY KEY,

ulica VARCHAR(60) NULL,

numer CHAR(10) NULL,

kod\_pocztowy VARCHAR(10) NULL,

miasto VARCHAR(30) NULL

)

CREATE TABLE Pracownicy(

id\_pracownik INT IDENTITY(1,1) PRIMARY KEY,

id\_adres\_pracownika INT NOT NULL,

imie VARCHAR(15) NOT NULL CHECK(LEN(imie)>2),

nazwisko VARCHAR(25) NOT NULL CHECK(LEN(nazwisko)>2),

pesel CHAR(15) NOT NULL UNIQUE,

pensja MONEY NOT NULL CHECK(pensja BETWEEN 2000 AND 3500),

stanowisko VARCHAR(20) NULL

)

ALTER TABLE Pracownicy

ADD FOREIGN KEY (id\_adres\_pracownika)

REFERENCES Adresy(id\_adres)

ON DELETE CASCADE

ON UPDATE CASCADE

CREATE TABLE Dostawcy(

id\_dostawca INT IDENTITY(1,1) PRIMARY KEY,

id\_adres\_dostawcy INT NOT NULL,

imie VARCHAR(15) NOT NULL CHECK(LEN(imie)>2),

nazwisko VARCHAR(25) NOT NULL CHECK(LEN(nazwisko)>2),

nazwa\_firmy VARCHAR(30) NULL,

staly\_dostawca VARCHAR(3) NULL

)

ALTER TABLE Dostawcy

ADD FOREIGN KEY (id\_adres\_dostawcy)

REFERENCES Adresy(id\_adres)

ON DELETE CASCADE

ON UPDATE CASCADE

CREATE TABLE Klienci(

id\_klient INT IDENTITY(1,1) PRIMARY KEY,

id\_adres\_klienta INT NOT NULL,

imie VARCHAR(15) NOT NULL CHECK(LEN(imie)>2),

nazwisko VARCHAR(25) NOT NULL CHECK(LEN(nazwisko)>2),

telefon VARCHAR(15) NULL,

staly\_klient VARCHAR(3) NULL

)

ALTER TABLE Klienci

ADD FOREIGN KEY (id\_adres\_klienta)

REFERENCES Adresy(id\_adres)

ON DELETE CASCADE

ON UPDATE CASCADE

CREATE TABLE Produkty(

id\_produkt INT IDENTITY(1,1) PRIMARY KEY,

id\_dostawcy INT NOT NULL,

nazwa\_produktu VARCHAR(20) NOT NULL,

cena MONEY NOT NULL,

dostepna\_ilosc INT NULL,

weganski VARCHAR(3) NULL

)

ALTER TABLE Produkty

ADD FOREIGN KEY (id\_dostawcy)

REFERENCES Dostawcy(id\_dostawca)

ON DELETE CASCADE

ON UPDATE CASCADE

CREATE TABLE Zamowienia(

id\_zamowienia INT IDENTITY(1,1) PRIMARY KEY,

data\_zamowienia DATE NOT NULL,

data\_odbioru DATE NOT NULL

)

ALTER TABLE Zamowienia

ADD CHECK (data\_odbioru>=data\_zamowienia)

CREATE TABLE [Szczegoly Zamowien](

id\_szczegolyz INT IDENTITY(1,1) PRIMARY KEY,

id\_szamowienia INT NOT NULL,

id\_sprodukt INT NOT NULL,

cena MONEY NULL,

ilosc INT NOT NULL,

znizka REAL NULL

)

ALTER TABLE [Szczegoly Zamowien]

ADD FOREIGN KEY (id\_szamowienia)

REFERENCES Zamowienia (id\_zamowienia)

ON DELETE CASCADE

ON UPDATE CASCADE

ALTER TABLE [Szczegoly Zamowien]

ADD FOREIGN KEY (id\_sprodukt)

REFERENCES Produkty (id\_produkt)

ON DELETE CASCADE

ON UPDATE CASCADE

CREATE TABLE [Wyroby Cukiernicze](

id\_wyroby INT IDENTITY(1,1) PRIMARY KEY,

nazwa VARCHAR(20) NOT NULL,

cena MONEY NOT NULL,

kalorie INT NULL,

bezglutenowe VARCHAR(3)

)

CREATE TABLE Sprzedaz(

id\_sprzedaz INT IDENTITY(1,1) PRIMARY KEY,

id\_spracownik INT NOT NULL,

id\_sklient INT NOT NULL,

data\_sprzedazy DATE NOT NULL,

zaliczka VARCHAR(3) NULL

)

ALTER TABLE Sprzedaz

ADD FOREIGN KEY (id\_spracownik)

REFERENCES Pracownicy(id\_pracownik)

ON DELETE CASCADE

ON UPDATE CASCADE

ALTER TABLE Sprzedaz

ADD FOREIGN KEY (id\_sklient)

REFERENCES Klienci(id\_klient)

ON DELETE CASCADE

ON UPDATE CASCADE

CREATE TABLE [Szczegoly Sprzedazy](

id\_szczegolys INT IDENTITY(1,1) PRIMARY KEY,

id\_swyroby INT NOT NULL,

id\_ssprzedaz INT NOT NULL,

cena MONEY NULL,

ilosc INT NOT NULL,

znizka REAL NULL

)

ALTER TABLE [Szczegoly Sprzedazy]

ADD FOREIGN KEY (id\_swyroby)

REFERENCES [Wyroby Cukiernicze](id\_wyroby)

ON DELETE CASCADE

ON UPDATE CASCADE

ALTER TABLE [Szczegoly Sprzedazy]

ADD FOREIGN KEY (id\_ssprzedaz)

REFERENCES Sprzedaz(id\_sprzedaz)

ON DELETE CASCADE

ON UPDATE CASCADE

1. funkcje

--max cena wyrobow

CREATE PROC najwyzsza\_cena

AS

DECLARE @makscena MONEY

SET @makscena = (SELECT MAX(cena) FROM [Wyroby Cukiernicze])

SELECT id\_wyroby,nazwa,cena FROM [Wyroby Cukiernicze]

WHERE cena=@makscena

GO

--EXEC najwyzsza\_cena

--dodaj adres

CREATE PROC dodaj\_adres

@ulica VARCHAR(30),@numer VARCHAR(10),@kod\_pocztowy VARCHAR(10),@miasto VARCHAR(30)

AS

INSERT INTO Adresy(ulica,numer,kod\_pocztowy,miasto)

VALUES(@ulica,@numer,@kod\_pocztowy,@miasto)

GO

--EXEC dodaj\_adres 'dsss',12,'23-22','sads'

--cena1

CREATE PROC cena\_ss

AS

UPDATE [Szczegoly Sprzedazy]

SET cena=(cena\*ilosc)-(cena\*znizka)

RETURN

--cena2

CREATE PROC cena\_sz

AS

UPDATE [Szczegoly Zamowien]

SET cena=(cena\*ilosc)-(cena\*znizka)

RETURN

--zamowienia zlozone po podanej dacie

CREATE PROC zamowienia

@data DATE

AS

BEGIN

SELECT data\_zamowienia,data\_odbioru FROM Zamowienia

WHERE data\_zamowienia>@data

END

GO

--zarobki nie moga byc zerowe

CREATE TRIGGER zarobki

ON Pracownicy

FOR INSERT

AS

DECLARE @pensja MONEY

SELECT @pensja = pensja FROM inserted

If @pensja = 0

BEGIN

ROLLBACK

RAISERROR('zarobki nie moga byc zerowe',1,1)

END

--dostepna ilosc produktow

CREATE TRIGGER ilosc

ON [Szczegoly Zamowien]

FOR INSERT,UPDATE

AS

DECLARE @ilosc INT,@dostepna\_ilosc INT

SELECT @ilosc=ilosc FROM inserted

SELECT @dostepna\_ilosc=dostepna\_ilosc FROM Produkty

IF @ilosc>@dostepna\_ilosc

BEGIN

ROLLBACK

RAISERROR('nie ma takiej ilosci dostepnych produktow',2,2)

END

--staly dostawca

CREATE TRIGGER dostawca

ON Produkty

FOR INSERT,UPDATE

AS

DECLARE @dostawca VARCHAR(3)

SELECT @dostawca=staly\_dostawca FROM Dostawcy

IF @dostawca!='tak'

BEGIN

ROLLBACK

RAISERROR('brak mozliwosci zamowien od nieznanego dostawcy',3,3)

END

--stanowisko

CREATE TRIGGER stanowisko

ON Sprzedaz

FOR INSERT,UPDATE

AS

DECLARE @stanowisko VARCHAR(20)

SELECT @stanowisko=stanowisko FROM Pracownicy

IF @stanowisko!='kasjer'

BEGIN

ROLLBACK

RAISERROR('tylko kasjer jest upowazniony do sprzedazy',4,4)

END

--znizka dla klientow, ktorzy wplacili zaliczke

CREATE TRIGGER znizka

ON [Szczegoly Sprzedazy]

FOR INSERT,UPDATE

AS

DECLARE @zaliczka VARCHAR(3),@znizka REAL

SELECT @znizka=znizka FROM [Szczegoly Sprzedazy]

SELECT @zaliczka=zaliczka FROM Sprzedaz

IF @zaliczka='tak'

BEGIN

SET @znizka=0.2

END

GO

CREATE VIEW produkty\_weganskie

AS

SELECT d.imie,d.nazwisko,d.nazwa\_firmy,p.nazwa\_produktu,p.cena

FROM Dostawcy d JOIN Produkty p

ON d.id\_dostawca=p.id\_dostawcy

GROUP BY d.nazwa\_firmy,d.imie,d.nazwisko,p.nazwa\_produktu,p.cena,p.weganski

HAVING p.weganski='tak'

CREATE VIEW wyroby\_bezglutenowe

AS

SELECT nazwa,cena

FROM [Wyroby Cukiernicze]

GROUP BY nazwa,cena,bezglutenowe

HAVING bezglutenowe='tak'

CREATE VIEW staly\_klient

AS

SELECT imie,nazwisko,telefon

FROM Klienci

GROUP BY imie,nazwisko,telefon,staly\_klient

HAVING staly\_klient='tak'

CREATE VIEW staly\_dostawca

AS

SELECT imie,nazwisko,nazwa\_firmy

FROM Dostawcy

GROUP BY nazwa\_firmy,imie,nazwisko,staly\_dostawca

HAVING staly\_dostawca='tak'

CREATE VIEW zaliczki

AS

SELECT k.imie,k.nazwisko

FROM Klienci k JOIN Sprzedaz s

ON k.id\_klient=s.id\_sklient

GROUP BY k.imie,k.nazwisko,s.zaliczka

HAVING zaliczka='tak'

CREATE VIEW zamowione\_produkty

AS

SELECT p.nazwa\_produktu,z.data\_zamowienia,z.data\_odbioru

FROM Produkty p JOIN [Szczegoly Zamowien] s

ON p.id\_produkt=s.id\_sprodukt RIGHT JOIN Zamowienia z

ON s.id\_szamowienia=z.id\_zamowienia

CREATE FUNCTION kalorie(@kalorie INT)

RETURNS TABLE AS

RETURN (SELECT \* FROM [Wyroby Cukiernicze]

WHERE kalorie<=@kalorie)

GO

--SELECT nazwa,kalorie

--FROM kalorie(2200)

CREATE VIEW adresy\_dostawcow

AS

SELECT d.imie,d.nazwisko,d.nazwa\_firmy,a.ulica,a.numer,a.kod\_pocztowy,a.miasto

FROM Dostawcy d RIGHT JOIN Adresy a

ON d.id\_adres\_dostawcy=a.id\_adres

CREATE VIEW adresy\_pracownikow

AS

SELECT p.imie,p.nazwisko,a.ulica,a.numer,a.kod\_pocztowy,a.miasto

FROM Pracownicy p RIGHT JOIN Adresy a

ON p.id\_adres\_pracownika=a.id\_adres

CREATE VIEW adresy\_klientow

AS

SELECT k.imie,k.nazwisko,k.telefon,a.ulica,a.numer,a.kod\_pocztowy,a.miasto

FROM Klienci k RIGHT JOIN Adresy a

ON k.id\_adres\_klienta=a.id\_adres

1. dane

INSERT INTO Adresy

VALUES

('Skrzydlata',22,'82-300','Elblag'),

('Krawiecka',6,'82-300','Elblag'),

('Grunwaldzka',83,'82-300','Elblag'),

('Morska',122,'83-400','Stegna'),

('Portowa',21,'81-200','Ustka'),

('Lewa',6,'81-211','Szczecin'),

('Warszawska',43,'61-222','Gdynia'),

('Torowa',33,'11-229','Tarnow'),

('Roczna',44,'23-222','Warszawa'),

('Mroczna',11,'82-300','Elblag'),

('Rycerska',31,'82-300','Elblag'),

('Rynkowa',77,'82-300','Elblag'),

('Otwarta',421,'83-400','Stegna'),

('Glowna',22,'81-200','Ustka'),

('Dawna',211,'81-211','Szczecin')

INSERT INTO Dostawcy

VALUES

(1,'Maciej','Kunach','Warzywko','tak'),

(2,'Regina','Phalange','Figa','tak'),

(3,'Mortimer','Cwir','Mewka','tak'),

(4,'Wlodzimierz','Len','Stokrotka','tak'),

(5,'Marek','Trzask','Castorama','nie')

INSERT INTO Pracownicy

VALUES

(6,'Tomek','Rot','98122827177',3400,'piekarz'),

(7,'Marta','Kot','81222274444',2200,'kasjer'),

(8,'Asia','Mak','27272828282',2330,'kasjer'),

(9,'Pola','Tlok','91111822221',2220,'kasjer'),

(10,'Marcin','Rok','98222222122',3200,'cukiernik')

INSERT INTO Klienci

VALUES

(11,'Eryk','Mars','512-333-333','tak'),

(12,'Piotr','Kowal','333-333-333','tak'),

(13,'Bartek','Szmal','553-222-444','tak'),

(14,'Marek','Topol','922-332-333','tak'),

(15,'Basia','Kruk','373-444-444','nie')

INSERT INTO Produkty

VALUES

(1,'kiwi',20,12,'tak'),

(1,'banany',10,50,'tak'),

(2,'maslo',44,40,'nie'),

(2,'mleko',22,44,'nie'),

(3,'kawa',111,22,'tak'),

(4,'maka',12,22,'tak'),

(4,'truskawki',10,22,'tak')

INSERT INTO Zamowienia

VALUES

('12-12-2017','12-12-2017'),

('13-12-2017','14-12-2017'),

('14-12-2017','15-12-2017'),

('15-12-2017','16-12-2017'),

('16-12-2017','17-12-2017'),

('17-12-2017','18-12-2017'),

('18-12-2017','22-12-2017'),

('19-12-2017','22-12-2017')

INSERT INTO [Szczegoly Zamowien]

VALUES

(1,1,null,10,0.1),

(2,1,null,9,0.15),

(3,2,null,11,0.1),

(4,3,null,22,0.11),

(5,4,null,11,0.1),

(6,5,null,7,0.3),

(7,6,null,1,0),

(8,7,null,20,0.2)

INSERT INTO Sprzedaz

VALUES

(2,1,'22-03-2018','tak'),

(2,2,'23-03-2018','tak'),

(3,3,'22-02-2018','nie'),

(3,4,'21-04-2018','tak'),

(4,5,'12-02-2018','nie')

INSERT INTO [Wyroby Cukiernicze]

VALUES

('piernik',22,1200,'nie'),

('drozdzowka',12,700,'nie'),

('tarta owocowa',14,300,'tak'),

('sernik',11,223,'nie'),

('tort',44,1200,'nie')

INSERT INTO [Szczegoly Sprzedazy]

VALUES

(1,1,null,2,0.2),

(2,2,null,10,0.3),

(3,3,null,4,0.1),

(4,4,null,6,0),

(5,5,null,4,0.11)