

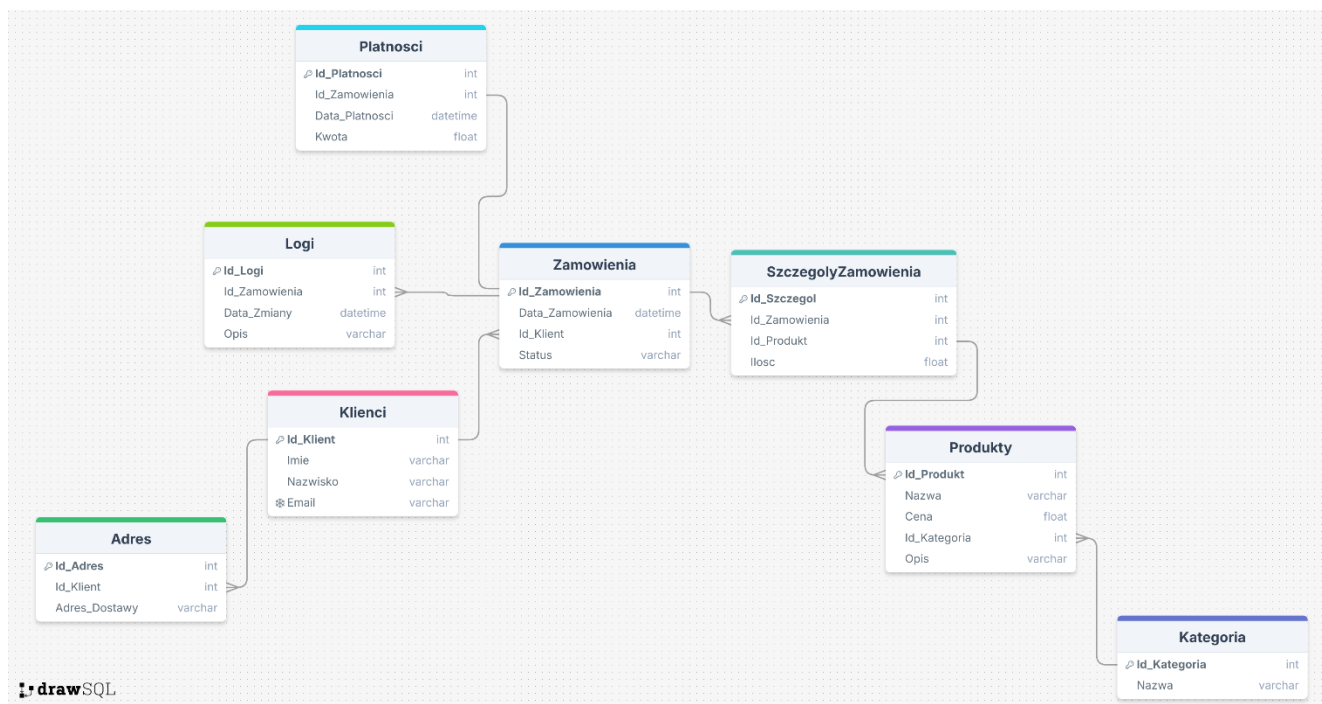
Butik internetowy

Natalia Ciarka

Spis treści

Spis treści	2
Model diagramu koncepcyjnego	2
Cele i założenia diagramu	3
Byty (Entity).....	3
Relacje lub interakcje między jednostkami	4
Model ERD.....	4
Schemat fizyczny	5
Widoki, procedury, triggerzy.....	5
Procedury:	5
Trigger	6
Widok:	6
Testowy zestaw danych	7
Skrypt SQL.....	9

Model diagramu koncepcyjnego



Cele i założenia

Cele:

- Jasne przedstawienie proponowanego systemu.
- Identyfikacja ważnych zależności.
- Zapewnienie wizualnego przeglądu systemu w celu wykrycia potencjalnych problemów.
- Przygotowanie diagramu jako narzędzia do dalszego projektowania.

Założenia:

Zakres funkcjonalności:

- Rejestracja klientów.
- Przeglądanie produktów.
- Składanie zamówień.
- Zarządzanie płatnościami i historią transakcji.

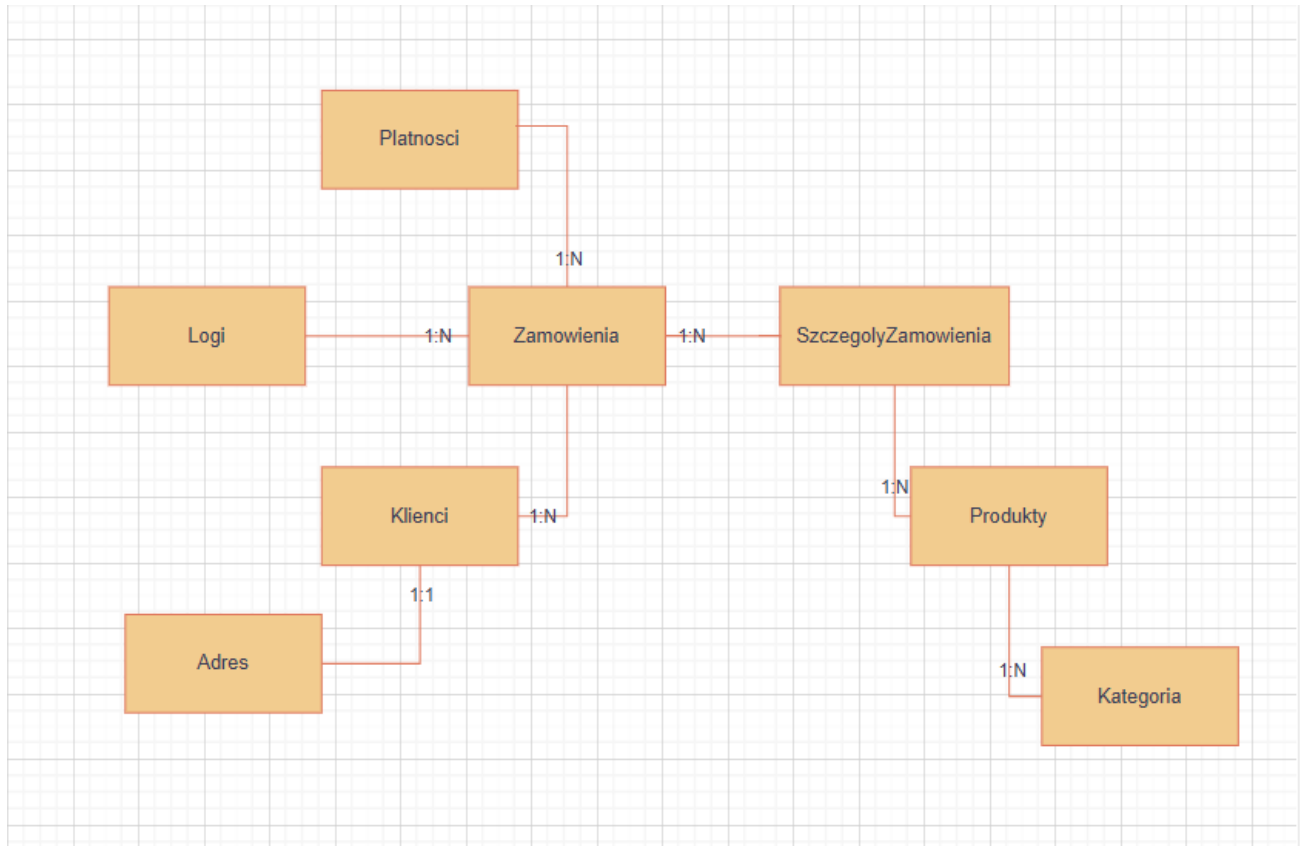
Ograniczenia:

- Każdy klient musi mieć unikalny email.
- Każdy produkt ma przypisaną kategorię i cenę.
- Zamówienie musi być powiązane z klientem i zawierać co najmniej jeden produkt.
- Wszystkie operacje są zapisywane w logach (historia zmian).

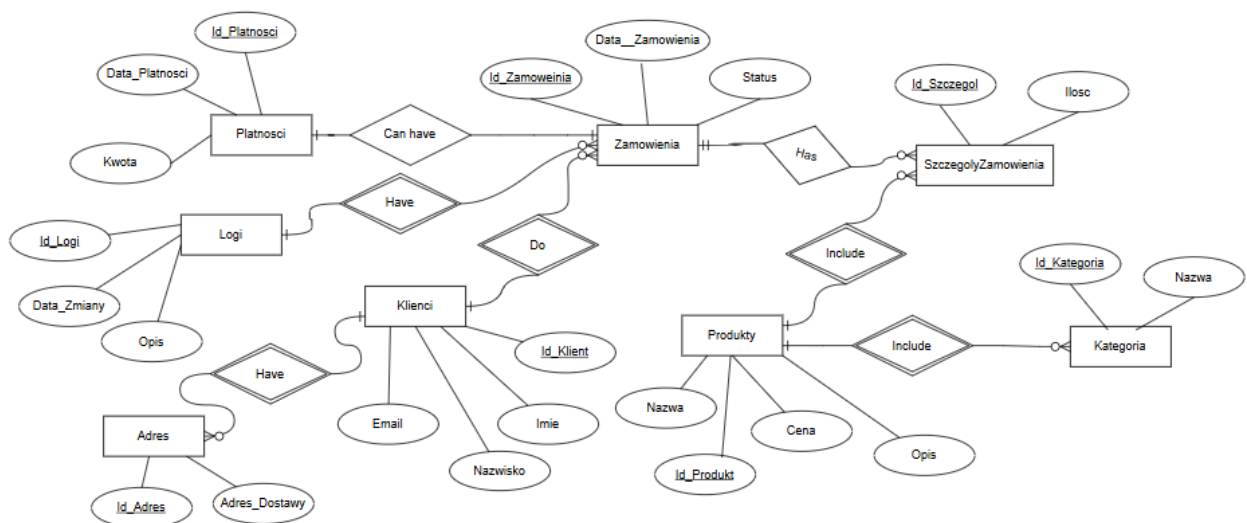
Byty (Entity)

- **Klienci:** Przechowuje dane o klientach sklepu, ich imiona, nazwiska oraz unikalne adresy email.
- **Produkty:** Informacje o asortymencie sklepu, takie jak nazwa, cena, kategoria i opis produktu.
- **Kategorie:** Kategoria przypisana do produktów.
- **Zamowienia:** Szczegóły zamówień klientów wraz z ich datą i statusem.
- **SzczegółyZamówienia:** Informacje o produktach w zamówieniu, informujące co i w jakiej ilości zostało zamówione.
- **Płatnosci:** Szczegóły dotyczące dokonanych płatności odnośnie zamówień produktów z ich datą i konkretną kwotą.
- **Logi:** Rejestr zmian w systemie.
- **Adres:** Przechowuje informacje o adresach klientów zakładając, że jeden klient może mieć dokładnie jeden adres.

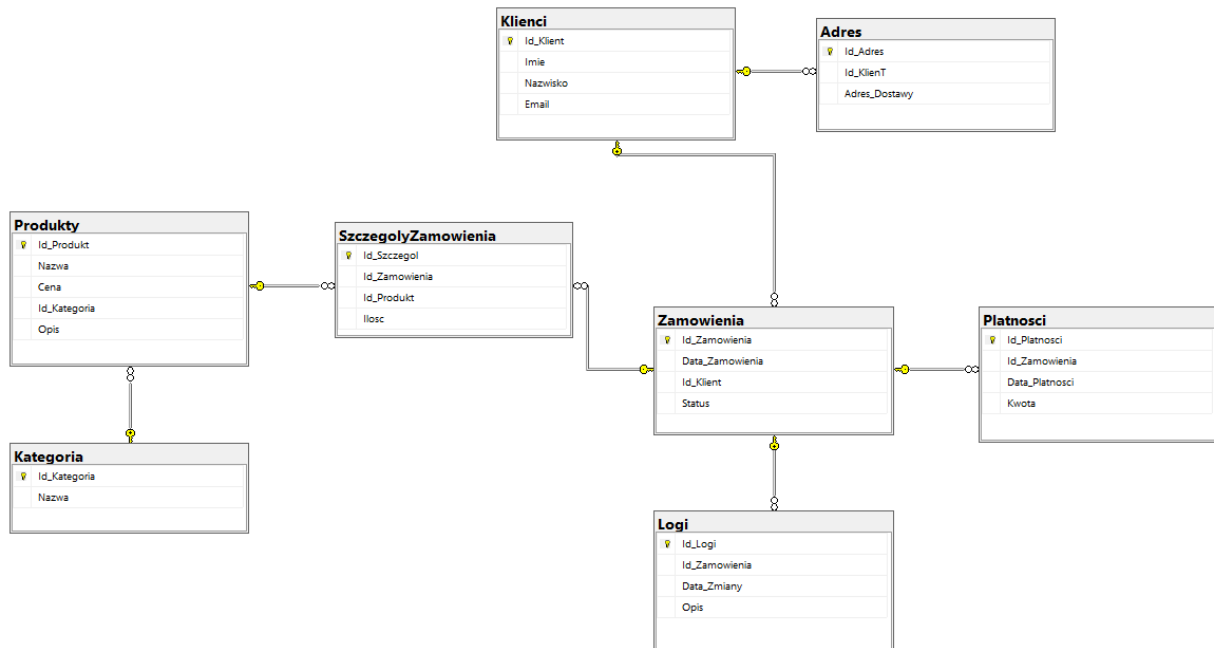
Relacje lub interakcje między jednostkami



Model ERD



Schemat fizyczny



Widoki, procedury, triggerzy

Procedury:

- Dodanie nowego klienta

```
CREATE PROCEDURE DodajKlienta
@Imie VARCHAR(255),
@Nazwisko VARCHAR(255),
@email VARCHAR(255)
AS
BEGIN
IF NOT EXISTS (SELECT 1 FROM Klienci WHERE Email = @Email)
BEGIN
INSERT INTO Klienci (Imie, Nazwisko, Email)
VALUES (@Imie, @Nazwisko, @Email)
END
END

EXEC DodajKlienta @Imie = 'Kasia', @Nazwisko = 'Kowalska', @Email =
'kasia.kowalska@example.com'
```
- Aktualizacja produktu

```
CREATE PROCEDURE AktualizujProdukt
@Id_Produkt INT,
@Nazwa VARCHAR(255),
@Cena FLOAT,
```

```

@Id_Kategoria INT, nume
@Opis VARCHAR(255)
AS
BEGIN
IF EXISTS (SELECT 1 FROM Produkty WHERE Id_Produkt = @Id_Produkt)
BEGIN
UPDATE Produkty
SET Nazwa = @Nazwa, Cena = @Cena, Id_Kategoria = @Id_Kategoria, Opis = @Opis
WHERE Id_Produkt = @Id_Produkt
END
END

EXEC AktualizujProdukt @Id_Produkt = 5, @Nazwa = 'Mikser', @Cena = 277.99, @Id_Kategoria=
5, @Opis = 'Mikser kuchenny'

```

- Usuwanie zamówienia

```

CREATE PROCEDURE UsunZamowienie
@Id_Zamowienia INT
AS
BEGIN
IF EXISTS (SELECT 1 FROM Zamowienia WHERE Id_Zamowienia = @Id_Zamowienia)
BEGIN
DELETE FROM Logi
WHERE Id_Zamowienia = @Id_Zamowienia
DELETE FROM Platnosci
WHERE Id_Zamowienia = @Id_Zamowienia;
DELETE FROM SzczegolyZamowienia
WHERE Id_Zamowienia = @Id_Zamowienia
DELETE FROM Zamowienia
WHERE Id_Zamowienia = @Id_Zamowienia
END
END

EXEC UsunZamowienie @Id_Zamowienia = 2

```

Trigger

- Trigger dodający nowe zamówienie do tabeli Logi

```

CREATE TRIGGER DodajNoweZamowienieDoLogi
ON Zamowienia
AFTER INSERT
AS
BEGIN
INSERT INTO Logi (Id_Zamowienia, Data_Zmiany, Opis)
SELECT Id_Zamowienia, GETDATE(), 'Nowe zamówienie utworzone'
FROM Inserted
END

INSERT INTO Zamowienia (Data_Zamowienia, Id_Klient, Status)
VALUES ('2025-01-22 12:00', 1, 'Nowe')

```

Widok:

- Widok z danymi o zamówieniach

```

CREATE VIEW DaneZamowien AS
SELECT
Klienci.Id_Klient,
Klienci.Imie,

```

```

Klienci.Nazwisko,
Zamowienia.Id_Zamowienia,
Zamowienia.Data_Zamowienia,
Zamowienia.Status,
Platnosci.Kwota AS KwotaPlatnosci
FROM Klienci, Zamowienia, Platnosci
WHERE Klienci.Id_Klient = Zamowienia.Id_Klient
AND Zamowienia.Id_Zamowienia = Platnosci.Id_Zamowienia; zamówieniach

```

Id_Klient	Imie	Nazwisko	Id_Zamowienia	Data_Zamowienia	Status	KwotaPlatnosci
1	Anna	Kowalska	1	2025-01-10 10:30:00.000	Nowe	149,97
3	Kasia	Wisniewska	3	2025-01-12 12:15:00.000	Zrealizowane	359,97
4	Piotr	Zielinski	4	2025-01-13 08:45:00.000	Nowe	199,99
5	Maria	Jankowska	5	2025-01-14 09:50:00.000	Nowe	119,96
6	Tomasz	Kowalczyk	6	2025-01-15 16:30:00.000	W realizacji	499,9
7	Joanna	Lewandowska	7	2025-01-16 10:00:00.000	Zrealizowane	249,99
8	Michał	Włodarczyk	8	2025-01-17 11:00:00.000	Nowe	299,98
9	Magda	Węglarz	9	2025-01-18 15:20:00.000	Zrealizowane	149,97
10	Adam	Nowicki	10	2025-01-19 14:10:00.000	W realizacji	749,97
11	Ewa	Domanska	11	2025-01-20 10:25:00.000	Nowe	89,96
12	Kamil	Szymczak	12	2025-01-21 13:40:00.000	Zrealizowane	599,97

Testowy zestaw danych

```

INSERT INTO Klienci (Imie, Nazwisko, Email) VALUES
('Anna', 'Kowalska', 'anna.kowalska@example.com'),
('Jan', 'Nowak', 'jan.nowak@example.com'),
('Kasia', 'Wiśniewska', 'kasia.wisniewska@example.com'),
('Piotr', 'Zieliński', 'piotr.zielinski@example.com'),
('Maria', 'Jankowska', 'maria.jankowska@example.com'),
('Tomasz', 'Kowalczyk', 'tomasz.kowalczyk@example.com'),
('Joanna', 'Lewandowska', 'joanna.lewandowska@example.com'),
('Michał', 'Włodarczyk', 'michal.wlodarczyk@example.com'),
('Magda', 'Węglarz', 'magda.weglarz@example.com'),
('Adam', 'Nowicki', 'adam.nowicki@example.com'),
('Ewa', 'Domańska', 'ewa.domanska@example.com'),
('Kamil', 'Szymczak', 'kamil.szymczak@example.com')

```

```

INSERT INTO Kategoria (Nazwa) VALUES
('Odzież'),
('Elektronika'),
('Książki'),
('Sport'),
('AGD'),
('Zabawki'),
('Meble'),
('Kosmetyki'),
('Jedzenie'),
('Obuwie'),
('Bizuteria'),
('Ogród')

```

```

INSERT INTO Produkty (Nazwa, Cena, Id_Kategoria, Opis) VALUES
('Koszulka', 49.99, 1, 'Koszulka z nadrukiem'),
('Laptop', 2499.99, 2, 'Laptop gamingowy'),

```

```
( 'Książka', 19.99, 3, 'Książka fantasy'),
( 'Piłka nożna', 79.99, 4, 'Piłka do gry na trawie'),
( 'Mikser', 299.99, 5, 'Mikser kuchenny'),
( 'Lalka', 59.99, 6, 'Lalka Barbie'),
( 'Krzesło', 199.99, 7, 'Krzesło biurowe'),
( 'Szampon', 29.99, 8, 'Szampon do włosów suchych'),
( 'Czekolada', 4.99, 9, 'Czekolada mleczna'),
( 'Buty sportowe', 249.99, 10, 'Buty do biegania'),
( 'Naszyjnik', 149.99, 11, 'Naszyjnik srebrny'),
( 'Grabie', 49.99, 12, 'Grabie do ogrodu')
```

```
INSERT INTO Zamowienia (Data_Zamowienia, Id_Klient, Status) VALUES
( '2025-01-10 10:30', 1, 'Nowe'),
( '2025-01-11 14:00', 2, 'W realizacji'),
( '2025-01-12 12:15', 3, 'Zrealizowane'),
( '2025-01-13 08:45', 4, 'Nowe'),
( '2025-01-14 09:50', 5, 'Nowe'),
( '2025-01-15 16:30', 6, 'W realizacji'),
( '2025-01-16 10:00', 7, 'Zrealizowane'),
( '2025-01-17 11:00', 8, 'Nowe'),
( '2025-01-18 15:20', 9, 'Zrealizowane'),
( '2025-01-19 14:10', 10, 'W realizacji'),
( '2025-01-20 10:25', 11, 'Nowe'),
( '2025-01-21 13:40', 12, 'Zrealizowane')
```

```
INSERT INTO SzczegolyZamowienia (Id_Zamowienia, Id_Produkt, Ilosc) VALUES
(1, 1, 2),
(1, 2, 1),
(2, 3, 1),
(2, 4, 3),
(3, 5, 1),
(3, 6, 2),
(4, 7, 1),
(5, 8, 4),
(6, 9, 10),
(7, 10, 1),
(8, 11, 2),
(9, 12, 3)
```

```
INSERT INTO Platnosci (Id_Zamowienia, Data_Platnosci, Kwota) VALUES
(1, '2025-01-10 11:00', 149.97),
(2, '2025-01-11 14:30', 259.97),
(3, '2025-01-12 13:00', 359.97),
(4, '2025-01-13 09:00', 199.99),
(5, '2025-01-14 10:00', 119.96),
(6, '2025-01-15 17:00', 499.90),
(7, '2025-01-16 11:00', 249.99),
(8, '2025-01-17 12:00', 299.98),
(9, '2025-01-18 16:00', 149.97),
(10, '2025-01-19 15:00', 749.97),
(11, '2025-01-20 11:00', 89.96),
(12, '2025-01-21 14:00', 599.97)
```

```
INSERT INTO Logi (Id_Zamowienia, Data_Zmiany, Opis) VALUES
(1, '2025-01-10 11:00', 'Zamówienie utworzone'),
(1, '2025-01-10 11:10:', 'Płatność zrealizowana'),
(2, '2025-01-11 14:30', 'Zamówienie w realizacji'),
(3, '2025-01-12 13:00', 'Zamówienie zrealizowane'),
(4, '2025-01-13 09:00', 'Zamówienie utworzone'),
(5, '2025-01-14 10:00', 'Płatność zrealizowana'),
(6, '2025-01-15 17:00', 'Zamówienie w realizacji'),
```



```
(7, '2025-01-16 11:00', 'Zamówienie zrealizowane'),
(8, '2025-01-17 12:00', 'Zamówienie utworzone'),
(9, '2025-01-18 16:00', 'Zamówienie zrealizowane'),
(10, '2025-01-19 15:00', 'Zamówienie w realizacji'),
(11, '2025-01-20 11:00', 'Zamówienie utworzone')
```

```
INSERT INTO Adres (Id_Klient, Adres_Dostawy) VALUES
(1, 'ul. Polna 1, Warszawa'),
(2, 'ul. Zielona 2, Kraków'),
(3, 'ul. Szeroka 3, Poznań'),
(4, 'ul. Długa 4, Wrocław'),
(5, 'ul. Krótka 5, Gdańsk'),
(6, 'ul. Miła 6, Szczecin'),
(7, 'ul. Jasna 7, Bydgoszcz'),
(8, 'ul. Nowa 8, Lublin'),
(9, 'ul. Stara 9, Katowice'),
(10, 'ul. Wąska 10, Łódź'),
(11, 'ul. Rynek 11, Toruń'),
(12, 'ul. Kamienna 12, Rzeszów')
```

Skrypt SQL

```
CREATE DATABASE [BUTIK_INTERNETOWY]
```

```
USE [BUTIK_INTERNETOWY]
```

```
-- Utworzenie tabel
```

```
CREATE TABLE Klienci (
Id_Klient INT NOT NULL IDENTITY PRIMARY KEY,
Imie VARCHAR(255) NOT NULL,
Nazwisko VARCHAR(255) NOT NULL,
Email VARCHAR (255) UNIQUE NOT NULL
);
```

```
CREATE TABLE Produkty(
Id_Produkt INT NOT NULL IDENTITY PRIMARY KEY,
Nazwa VARCHAR(255) NOT NULL,
Cena FLOAT(53) NOT NULL,
Id_Kategoria INT NOT NULL,
Opis VARCHAR(255) NOT NULL
);
```

```
CREATE TABLE Kategoria(
Id_Kategoria INT NOT NULL IDENTITY PRIMARY KEY,
Nazwa VARCHAR(255) NOT NULL
);
```

```
CREATE TABLE Zamowienia(
Id_Zamowienia INT NOT NULL IDENTITY PRIMARY KEY,
Data_Zamowienia DATETIME NOT NULL,
Id_Klient INT NOT NULL,
Status VARCHAR(255) NOT NULL
);
```

```
CREATE TABLE SzczegolyZamowienia(
Id_Szczegol INT NOT NULL IDENTITY PRIMARY KEY,
Id_Zamowienia INT NOT NULL,
Id_Produkt INT NOT NULL,
Ilosc FLOAT(53) NOT NULL
```

```

);

CREATE TABLE Platnosci(
Id_Platnosci INT NOT NULL IDENTITY PRIMARY KEY,
Id_Zamowienia INT NOT NULL,
Data_Platnosci DATETIME NOT NULL,
Kwota FLOAT(53) NOT NULL
);

CREATE TABLE Logi(
Id_Logi INT NOT NULL IDENTITY PRIMARY KEY,
Id_Zamowienia INT NOT NULL,
Data_Zmiany DATETIME NOT NULL,
Opis VARCHAR(255) NOT NULL
);

CREATE TABLE Adres(
Id_Adres INT NOT NULL IDENTITY PRIMARY KEY,
Id_Klient INT NOT NULL,
Adres_Dostawy VARCHAR(255) NOT NULL
);

-- Dodanie kluczy obcych do tabel
ALTER TABLE Logi
ADD CONSTRAINT logi_id_zamowienia_foreign FOREIGN KEY(Id_Zamowienia) REFERENCES
Zamowienia(Id_Zamowienia)

ALTER TABLE Adres
ADD CONSTRAINT adres_id_klient_foreign FOREIGN KEY(Id_Klient) REFERENCES Klienci(Id_Klient)

ALTER TABLE SzczegolyZamowienia
ADD CONSTRAINT szczegolyzamowienia_id_zamowienia_foreign FOREIGN KEY(Id_Zamowienia) REFERENCES
Zamowienia(Id_Zamowienia)

ALTER TABLE Platnosci
ADD CONSTRAINT platnosci_id_zamowienia_foreign FOREIGN KEY(Id_Zamowienia) REFERENCES
Zamowienia(Id_Zamowienia)

ALTER TABLE Produkty
ADD CONSTRAINT produkty_id_kategoria_foreign FOREIGN KEY(Id_Kategoria) REFERENCES
Kategoria(Id_Kategoria)
ALTER TABLE Zamowienia
ADD CONSTRAINT zamowienia_id_klient_foreign FOREIGN KEY(Id_Klient) REFERENCES
Klienci(Id_Klient)

ALTER TABLE SzczegolyZamowienia
ADD CONSTRAINT szczegolyzamowienia_id_produk_t_foreign FOREIGN KEY(Id_Produkt) REFERENCES
Produkty(Id_Produkt);

-- Procedura dodająca nowego klienta
CREATE PROCEDURE DodajKlienta
@Imie VARCHAR(255),
@Nazwisko VARCHAR(255),
@Email VARCHAR(255)
AS
BEGIN
IF NOT EXISTS (SELECT 1 FROM Klienci WHERE Email = @Email)
BEGIN
INSERT INTO Klienci (Imie, Nazwisko, Email)
VALUES (@Imie, @Nazwisko, @Email)
END
END

```

END

```
EXEC DodajKlienta @Imie = 'Kasia', @Nazwisko = 'Kowalska', @Email =  
'kasia.kowalska@example.com'
```

-- Procedura aktualizująca produkt

```
CREATE PROCEDURE AktualizujProdukt
```

```
@Id_Produkt INT,  
@Nazwa VARCHAR(255),  
@Cena FLOAT,  
@Id_Kategoria INT,  
@Opis VARCHAR(255)
```

```
AS
```

```
BEGIN
```

```
IF EXISTS (SELECT 1 FROM Produkty WHERE Id_Produkt = @Id_Produkt)
```

```
BEGIN
```

```
UPDATE Produkty
```

```
SET Nazwa = @Nazwa, Cena = @Cena, Id_Kategoria = @Id_Kategoria, Opis = @Opis
```

```
WHERE Id_Produkt = @Id_Produkt
```

```
END
```

```
END
```

```
EXEC AktualizujProdukt @Id_Produkt = 5, @Nazwa = 'Mikser', @Cena = 277.99, @Id_Kategoria = 5,  
@Opis = 'Mikser kuchenny'
```

-- Procedura usuwająca zamówienie

```
CREATE PROCEDURE UsunZamowienie
```

```
@Id_Zamowienia INT
```

```
AS
```

```
BEGIN
```

```
IF EXISTS (SELECT 1 FROM Zamowienia WHERE Id_Zamowienia = @Id_Zamowienia)
```

```
BEGIN
```

```
DELETE FROM Logi
```

```
WHERE Id_Zamowienia = @Id_Zamowienia
```

```
DELETE FROM Platnosci
```

```
WHERE Id_Zamowienia = @Id_Zamowienia;
```

```
DELETE FROM SzczegolyZamowienia
```

```
WHERE Id_Zamowienia = @Id_Zamowienia
```

```
DELETE FROM Zamowienia
```

```
WHERE Id_Zamowienia = @Id_Zamowienia
```

```
END
```

```
END
```

```
EXEC UsunZamowienie @Id_Zamowienia = 2
```

-- Trigger dodając nowe zamówienie do tabeli Logi

```
CREATE TRIGGER DodajNoweZamowienieDoLogi
```

```
ON Zamowienia
```

```
AFTER INSERT
```

```
AS
```

```
BEGIN
```

```
INSERT INTO Logi (Id_Zamowienia, Data_Zmiany, Opis)
```

```
SELECT Id_Zamowienia, GETDATE(), 'Nowe zamówienie utworzone'
```

```
FROM Inserted
```

```
END
```

```
INSERT INTO Zamowienia (Data_Zamowienia, Id_Klient, Status)
```

```
VALUES ('2025-01-22 12:00', 1, 'Nowe')
```

-- Tworzenie widoku z danymi o zamówieniach

```

CREATE VIEW DaneZamowien AS
SELECT
Klienci.Id_Klient,
Klienci.Imie,
Klienci.Nazwisko,
Zamowienia.Id_Zamowienia,
Zamowienia.Data_Zamowienia,
Zamowienia.Status,
Platnosci.Kwota AS KwotaPlatnosci
FROM Klienci, Zamowienia, Platnosci
WHERE Klienci.Id_Klient = Zamowienia.Id_Klient
AND Zamowienia.Id_Zamowienia = Platnosci.Id_Zamowienia;

SELECT*FROM DaneZamowien

-- Dodanie danych do tabel
INSERT INTO Klienci (Imie, Nazwisko, Email) VALUES
('Anna', 'Kowalska', 'anna.kowalska@example.com'),
('Jan', 'Nowak', 'jan.nowak@example.com'),
('Kasia', 'Wiśniewska', 'kasia.wisniewska@example.com'),
('Piotr', 'Zieliński', 'piotr.zielinski@example.com'),
('Maria', 'Jankowska', 'maria.jankowska@example.com'),
('Tomasz', 'Kowalczyk', 'tomasz.kowalczyk@example.com'),
('Joanna', 'Lewandowska', 'joanna.lewandowska@example.com'),
('Michał', 'Włodarczyk', 'michal.wlodarczyk@example.com'),
('Magda', 'Węglarz', 'magda.weglarz@example.com'),
('Adam', 'Nowicki', 'adam.nowicki@example.com'),
('Ewa', 'Domańska', 'ewa.domanska@example.com'),
('Kamil', 'Szymczak', 'kamil.szymczak@example.com')

INSERT INTO Kategoria (Nazwa) VALUES
('Odzież'),
('Elektronika'),
('Książki'),
('Sport'),
('AGD'),
('Zabawki'),
('Meble'),
('Kosmetyki'),
('Jedzenie'),
('Obuwie'),
('Biżuteria'),
('Ogród')

INSERT INTO Produkty (Nazwa, Cena, Id_Kategoria, Opis) VALUES
('Koszulka', 49.99, 1, 'Koszulka z nadrukiem'),
('Laptop', 2499.99, 2, 'Laptop gamingowy'),
('Książka', 19.99, 3, 'Książka fantasy'),
('Piłka nożna', 79.99, 4, 'Piłka do gry na trawie'),
('Mikser', 299.99, 5, 'Mikser kuchenny'),
('Lalka', 59.99, 6, 'Lalka Barbie'),
('Krzesło', 199.99, 7, 'Krzesło biurowe'),
('Szampon', 29.99, 8, 'Szampon do włosów suchych'),
('Czekolada', 4.99, 9, 'Czekolada mleczna'),
('Buty sportowe', 249.99, 10, 'Buty do biegania'),
('Naszyjnik', 149.99, 11, 'Naszyjnik srebrny'),
('Grabie', 49.99, 12, 'Grabie do ogrodu')

INSERT INTO Zamowienia (Data_Zamowienia, Id_Klient, Status) VALUES
('2025-01-10 10:30', 1, 'Nowe'),

```

```
( '2025-01-11 14:00', 2, 'W realizacji'),
( '2025-01-12 12:15', 3, 'Zrealizowane'),
( '2025-01-13 08:45', 4, 'Nowe'),
( '2025-01-14 09:50', 5, 'Nowe'),
( '2025-01-15 16:30', 6, 'W realizacji'),
( '2025-01-16 10:00', 7, 'Zrealizowane'),
( '2025-01-17 11:00', 8, 'Nowe'),
( '2025-01-18 15:20', 9, 'Zrealizowane'),
( '2025-01-19 14:10', 10, 'W realizacji'),
( '2025-01-20 10:25', 11, 'Nowe'),
( '2025-01-21 13:40', 12, 'Zrealizowane')
```

```
INSERT INTO SzczegolyZamowienia (Id_Zamowienia, Id_Produkt, Ilosc) VALUES
```

```
(1, 1, 2),
(1, 2, 1),
(2, 3, 1),
(2, 4, 3),
(3, 5, 1),
(3, 6, 2),
(4, 7, 1),
(5, 8, 4),
(6, 9, 10),
(7, 10, 1),
(8, 11, 2),
(9, 12, 3)
```

```
INSERT INTO Platnosci (Id_Zamowienia, Data_Platnosci, Kwota) VALUES
```

```
(1, '2025-01-10 11:00', 149.97),
(2, '2025-01-11 14:30', 259.97),
(3, '2025-01-12 13:00', 359.97),
(4, '2025-01-13 09:00', 199.99),
(5, '2025-01-14 10:00', 119.96),
(6, '2025-01-15 17:00', 499.90),
(7, '2025-01-16 11:00', 249.99),
(8, '2025-01-17 12:00', 299.98),
(9, '2025-01-18 16:00', 149.97),
(10, '2025-01-19 15:00', 749.97),
(11, '2025-01-20 11:00', 89.96),
(12, '2025-01-21 14:00', 599.97)
```

```
INSERT INTO Logi (Id_Zamowienia, Data_Zmiany, Opis) VALUES
```

```
(1, '2025-01-10 11:00', 'Zamówienie utworzone'),
(1, '2025-01-10 11:10', 'Płatność zrealizowana'),
(2, '2025-01-11 14:30', 'Zamówienie w realizacji'),
(3, '2025-01-12 13:00', 'Zamówienie zrealizowane'),
(4, '2025-01-13 09:00', 'Zamówienie utworzone'),
(5, '2025-01-14 10:00', 'Płatność zrealizowana'),
(6, '2025-01-15 17:00', 'Zamówienie w realizacji'),
(7, '2025-01-16 11:00', 'Zamówienie zrealizowane'),
(8, '2025-01-17 12:00', 'Zamówienie utworzone'),
(9, '2025-01-18 16:00', 'Zamówienie zrealizowane'),
(10, '2025-01-19 15:00', 'Zamówienie w realizacji'),
(11, '2025-01-20 11:00', 'Zamówienie utworzone')
```

```
INSERT INTO Adres (Id_Klient, Adres_Dostawy) VALUES
```

```
(1, 'ul. Polna 1, Warszawa'),
(2, 'ul. Zielona 2, Kraków'),
(3, 'ul. Szeroka 3, Poznań'),
(4, 'ul. Długa 4, Wrocław'),
(5, 'ul. Krótka 5, Gdańsk'),
(6, 'ul. Miła 6, Szczecin'),
```

```
(7, 'ul. Jasna 7, Bydgoszcz'),  
(8, 'ul. Nowa 8, Lublin'),  
(9, 'ul. Stara 9, Katowice'),  
(10, 'ul. Wąska 10, Łódź'),  
(11, 'ul. Rynek 11, Toruń'),  
(12, 'ul. Kamienna 12, Rzeszów')
```

Streszczenie

Projekt bazy danych dla butiku internetowego przedstawia strukturę systemu pozwalającą na efektywne zarządzanie klientami, produktami oraz zamówieniami.

Zakres funkcjonalności

1. Rejestracja klientów:

- Tabela **Klienci** zawiera Id_Klient, Imie, Nazwisko, i Email, co pozwala na rejestrację klientów.
- Ograniczenie **Email (UNIQUE)** zapewnia unikalność adresów e-mail dla każdego klienta.

2. Przeglądanie produktów:

- Tabela **Produkty** zawiera Id_Produkt, Nazwa, Cena, Opis oraz Id_Kategoria, co umożliwia przeglądanie produktów z przypisaną kategorią.

3. Składanie zamówień:

- Tabela **Zamowienia** powiązana jest z tabelą **Klienci** (relacja 1:N), co umożliwia składanie zamówień przez klientów.
- Tabela **SzczegolyZamowienia** przechowuje szczegóły zamówienia (produkty, ilości).

4. Zarządzanie płatnościami i historią transakcji:

- Tabela **Platnosci** jest powiązana z tabelą **Zamowienia** (relacja 1:N), co umożliwia zarządzanie płatnościami.
- Tabela **Logi** zapisuje operacje związane z zamówieniami i innymi działaniami, co umożliwia prowadzenie historii transakcji.

Ograniczenia

1. Unikalny email dla każdego klienta:

- Tabela **Klienci** ma kolumnę Email oznaczoną jako **UNIQUE**.

2. Przypisana kategoria i cena dla każdego produktu:

- Tabela **Produkty** zawiera Id_Kategoria jako klucz obcy do tabeli **Kategorie** oraz Cena jako obowiązkowy atrybut.

3. Zamówienie powiązane z klientem i zawierające co najmniej jeden produkt:

- Relacja 1:N między tabelami **Klienci** i **Zamowienia** oraz N:N między **Zamowienia** i **Produkty** (poprzez **SzczegolyZamowienia**).

4. Zapisywanie operacji w logach:

- Tabela **Logi** przechowuje Id_Log, Data_Zmiany i Opis, co umożliwia zapis historii zmian.