

# Filozofia baz danych

**Lokalne systemy** baz danych (lepszą nazwą: nieserwerowe):

- *Microsoft Excel / Libre Calc;*
- *Microsoft Access;*
- *JSON;*
- *VCF.*

**Sieciowe systemy** baz danych (klient / serwer):

- *Microsoft SQL Server;*
- *MySQL;*
- *Oracle;*
- *PostgreSQL;*

# Relacyjne bazy danych

Relacja to połączenie między dwiema tabelami na podstawie klucza.

# Relacyjne bazy danych

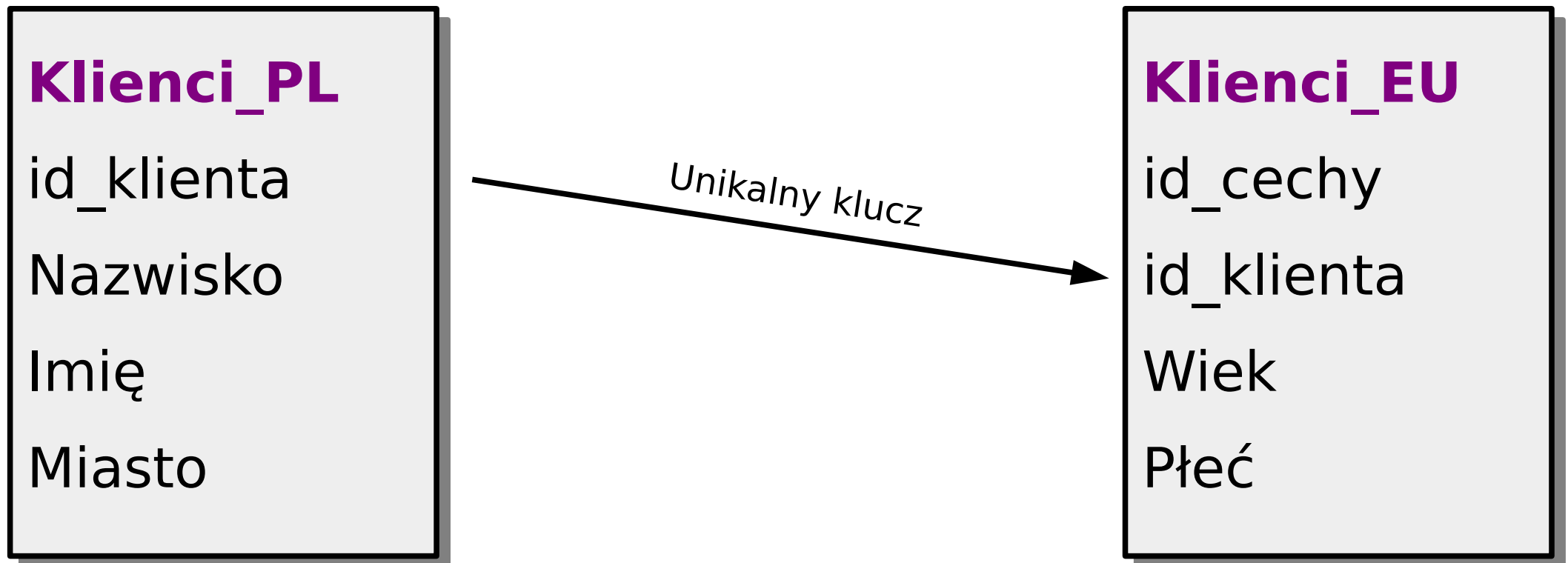
Trzy typy relacji:

- **jeden-do-jednego** (każdy element w każdej tabeli występuje tylko raz, jest unikatowy);
- **jeden-do-wielu** (element w tabeli może być powiązany z wieloma elementami w drugiej tabeli);
- **wiele-do-wielu** (czytelnik może posiadać wiele książek, ale każda książka może należeć do wielu czytelników; potrzebna trzecia tabela pośrednicząca z dwoma kluczami obcymi).

# Relacyjne bazy danych

*Między danymi w tabelach zachodzą różne relacje.*

*Tabele mają wspólny klucz (id).*



# Przykład powiązania tabel **jeden-do-wielu**

```
CREATE TABLE Klienci (  
  ID_Klienta INT PRIMARY KEY,  
  Imie VARCHAR(50),  
  Nazwisko VARCHAR(50),  
  Email VARCHAR(100)  
);
```

-- Jeden klient może posiadać wiele książek z antykwariatu,  
-- ale każda książka posiada jednego właściciela.

```
CREATE TABLE Książki (  
  ID_Książki INT PRIMARY KEY,  
  Tytuł VARCHAR(100),  
  Autor VARCHAR(100),  
  ID_Klienta INT, -- Klucz obcy  
  FOREIGN KEY (ID_Klienta) REFERENCES Klienci(ID_Klienta)  
);
```

# Przykład powiązania tabel **wiele-do-wielu**

W czasie tworzenia tabel:

```
CREATE TABLE Sprzedaż (  
  id_Sprzedaży INT PRIMARY KEY,  
  id_Projektu INT,  
  id_Klienta INT,  
  -- Obcy klucz w tej tabeli to id_Projektu powiązany z id_Projektu z tabeli Produkty  
  FOREIGN KEY (id_Projektu) REFERENCES Produkty(id_Projektu),  
  -- Drugi obcy klucz to id_Klienta powiązany z id_Klienta w tabeli Klienci  
  FOREIGN KEY (id_Klienta) REFERENCES Klienci(id_Klienta)  
);
```

W czasie pobierania danych:

```
SELECT  
  Sprzedaż.id_Sprzedaży,  
  Sprzedaż.id_Projektu,  
FROM  
  Sprzedaż  
JOIN  
  -- Dołączamy zewnętrzną tabelę Produkty, tworząc powiązanie kolumny  
  -- id_Projektu z tabeli Sprzedaż z id_Projektu z tabeli Produkty  
  Produkty ON Sprzedaż.id_Projektu = Produkty.id_Projektu;
```

# Oprogramowanie wspomagające

(CASE - *Computer-Aided Software Engineering*)

- **DB Designer4** - tryb wizualny (freeware)
- **SQL** - tryb tekstowy

# Wybrane typy danych

**CHAR** - 255 znaków (puste miejsca zajmują spacje)

**VARCHAR(n)** - do 255 znaków (długość dynamiczna)

**INTEGER(n)** - liczba całkowita

**FLOAT** - liczba rzeczywista, zmiennoprzecinkowa

**MONEY** - waluta (dwa miejsca po przecinku)

**DATE** - data

**ENUM** ('A','B') - wybór predefiniowanych znaków

**BINARY** - np. fotky

**BIT** - 0, 1 lub NULL



# Microsoft SQL Server

1. Otwieramy *Microsoft SQL Server Managment Studio*.
2. Tworzymy nową bazę: *Databases / New Database: nazwaBazy*.
3. Tworzymy strukturę bazy: *nazwaBazy / Tables / New Table... / Column Name* (tworzymy unikatowy klucz główny [*Set Primary Key*] oraz typy danych w kolumnach).
4. Tworzymy drugą tabelę z jej unikatowym kluczem głównym, ale także z „kluczem obcym” (np. *tabela2\_klucz\_obcy*).
5. Wiążemy klucz obcy z kluczem głównym pierwszej tabeli (tworzymy relację między tabelami): PPM -> **tabela2\_klucz\_obcy** (ikona strzałki) / *Relationship / Tables And Columns Specific / Nazwa Relacji*,  
**Primary Key Table:** *Tabela1* -> **klucz\_główny**,  
**Foreign Key Table:** *Tabela2* -> **tabela2\_klucz\_obcy**.
6. Sprawdzamy poprawność powiązania: *nazwaBazy / Database Diagrams / New Database Diagram*.

# Microsoft SQL Server

- Wczytywanie obcej bazy danych: PPM -> *Databases / Restore Database... / Device*