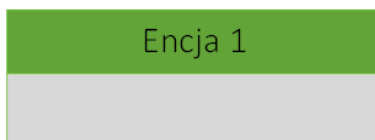


# Komponenty modelu danych

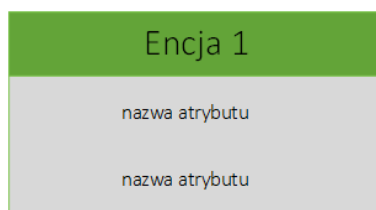
## Encja

- Reprezentuje zbiór obiektów charakteryzujący się identycznymi cechami.
- Posiada unikalną nazwę (rzeczownik w liczbie pojedynczej) i zestaw atrybutów.
- Konkretny obiekt świata rzeczywistego z właściwymi mu cechami jest reprezentowany, jako wystąpienie encji (instancja encji).



## Atrybut encji

- Stanowi własność (cechę) encji.
- Jest funkcją przypisującą obiektowi wartość cechy ze zbioru wartości cechy.
- Wyróżnia się dwa typy atrybutów:
  - identyfikatory – atrybut lub ich zbiór jednoznacznie identyfikujący wystąpienie encji,
  - deskryptory – wszystkie inne atrybuty poza identyfikatorami; reprezentują podstawowe własności encji.
- Atrybut wielowartościowy to atrybut, który może przyjąć wiele wartości w ramach tego samego wystąpienia encji (np. osoba może mieć wiele adresów).
- Atrybut złożony to atrybut posiadający swoją wewnętrzną strukturę (np. adres może składać się z wielu pól takich, jak ulica, numer domu, województwo, kraj etc.).



## Dziedzina atrybutu encji

- Kompletny zbiór wszystkich możliwych wartości, jakie może przyjmować atrybut dowolnego wystąpienia encji.
- Rodzaje dziedzin:
  - typ – specyfikuje typ danych (np.: integer),
  - lista – wylicza skończony zbiór dopuszczalnych wartości (np.: otwarte, wysłane, zamknięte, anulowane),
  - zakres – określa górną i dolną granicę dopuszczalnych wartości (np.: 0-100).

## Związki encji

### Typy asocjacji (kardynalność):

- jeden do jednego



- jeden do wielu



- wiele do wielu



### Klasy przynależności:

- związek opcjonalny



- związek obowiązkowy (obligatoryjny)



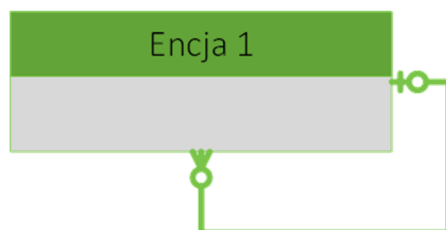
### Nazwy relacji

Nazwa relacji określa stosunek jednej encji do drugiej. Może być różna w zależności od kierunku relacji.

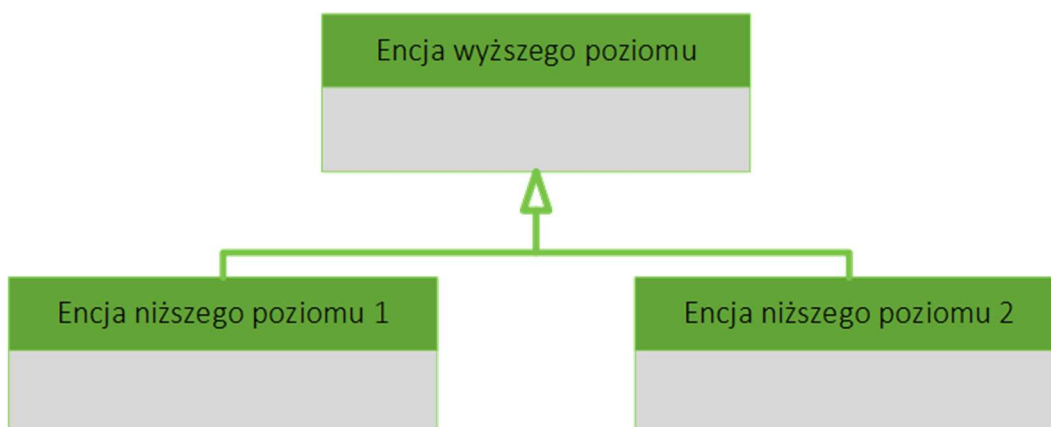


**Związek rekurencyjny**

- Musi być opcjonalny z obu stron. W przeciwnym wypadku powstałaby hierarchia nieskończona.
- Może być dowolnego typu.
- W związku rekurencyjnym encja pełni dwie role, które można zidentyfikować dzięki nazwie relacji

**Związek generalizacji**

Pewne encje o wspólnym zbiorze atrybutów można uogólnić tworząc encję wyższego poziomu (generalizacja). W związku generalizacji wszystkie atrybuty kadencji (encji wyższego poziomu) są dziedziczone przez potencje (encje niższego poziomu).

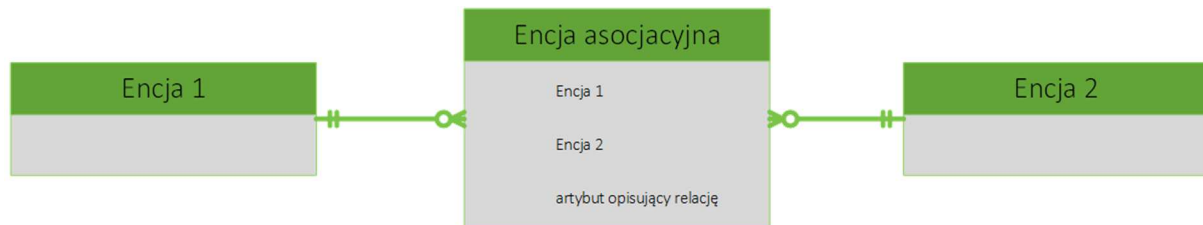
**Podsumowanie**

ERD Information Engineering Notation	
	Zero or one
	One only
	Zero or more
	One or more

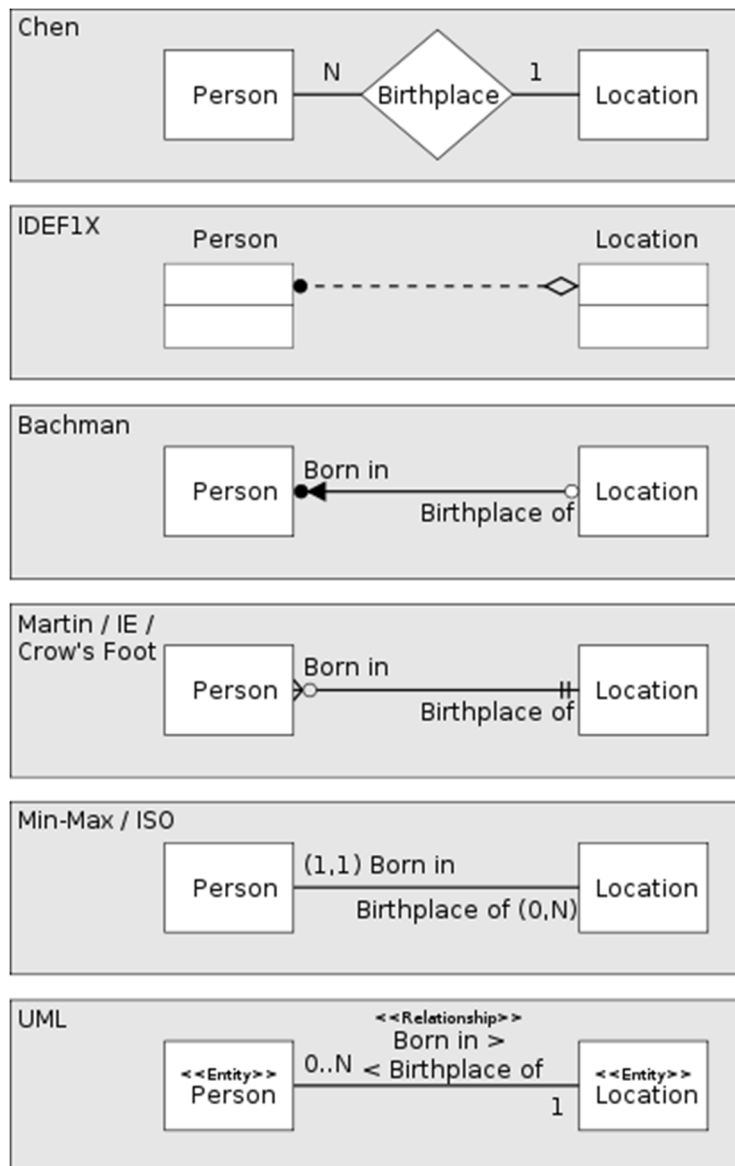
ERD Information Engineering Notation

### Rodzaje encji na poziomie konceptualnym

- Encje silne – ich wystąpienia są niezależne od wystąpień innych encji; posiadają własny identyfikator
- Encje słabe – ich wystąpienia mogą istnieć tylko w kontekście wystąpień encji powiązanych; nie posiadają własnego identyfikatora
- Encje asocjacyjne – służą do przechowywania danych o związkach



### Przykłady różnych notacji ERD



źródło: [http://en.wikipedia.org/wiki/Entity%E2%80%93relationship\\_model](http://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model)

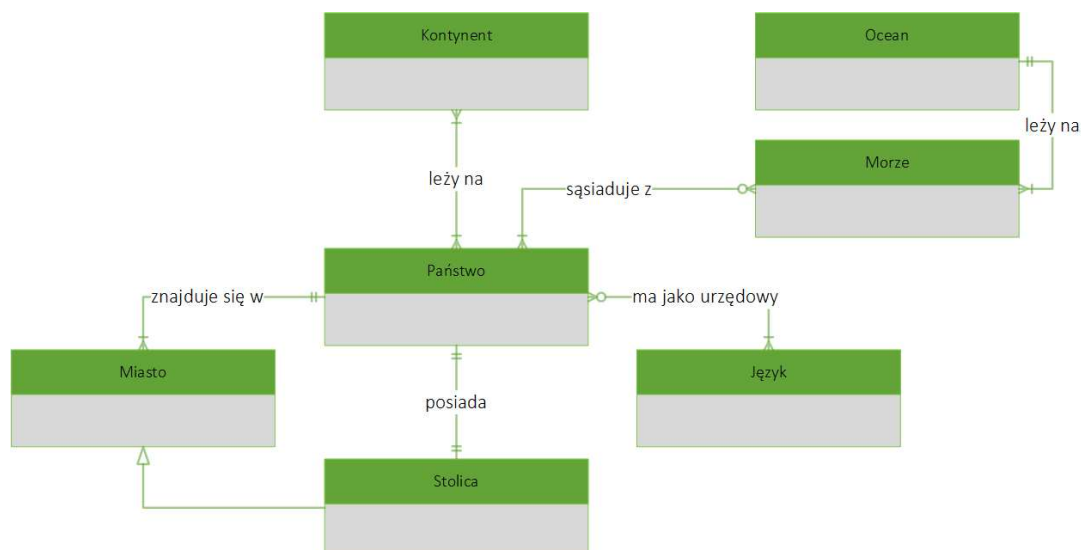
### Reguły biznesowe

Reguły biznesowe to ograniczenia specyficzne dla danej organizacji, zdefiniowane dla całego jej obszaru funkcjonowania<sup>1</sup>. Reguły biznesowe odnoszące się do modelu danych opisują językiem naturalnym wymagania klienta biznesowego dotyczące encji i relacji występujących pomiędzy nimi, które powinny zostać uwzględnione w modelu.

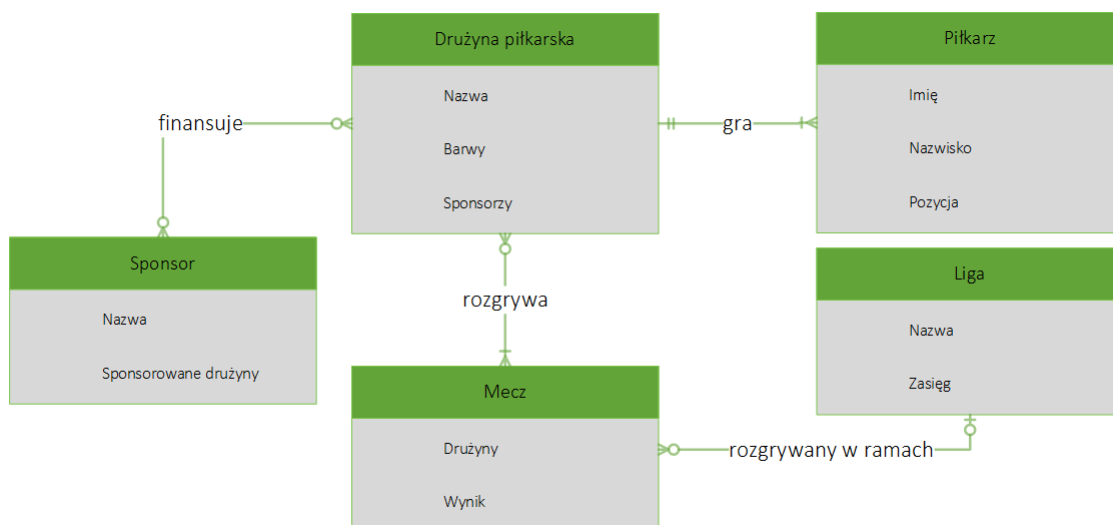
Zazwyczaj pojedyncza reguła biznesowa mówi o tym ile wystąpień encji podanej jako druga odpowiada pojedynczemu wystąpieniu encji wymienionej jako pierwsza. Możemy z takiej reguły odczytać kardynalność i klasę przynależności drugiej z encji w danym związku. Aby móc określić te parametry dla encji pierwszej, konieczne jest zdefiniowanie reguły opisującej związek z perspektywy encji drugiej.

### Przykładowe diagramy relacji encji:

#### Przykład 1



#### Przykład 2



<sup>1</sup> [https://it-consulting.pl/autoinstalator/wordpress/2010/11/22/reguly-biznesowe ---czym-sa/](https://it-consulting.pl/autoinstalator/wordpress/2010/11/22/reguly-biznesowe---czym-sa/)

# Modelowanie relacji encji. Model koncepcyjny

## Model koncepcyjny – etapy przygotowania

1. Określenie występujących encji.
2. Ustalenie typów występujących związków.
3. Określenie atrybutów odpowiadających poszczególnym encjom.
4. Określenie dziedziny poszczególnych atrybutów.
5. Utworzenie diagramu ER.

## Przykład

Reguły biznesowe biblioteki płyt z muzyką:

- Na jednym albumie może znaleźć się między 10 a 20 piosenek.
- Piosenka musi należeć wyłącznie do jednego albumu.
- Jeden album może należeć do wielu kategorii.
- Do jednej kategorii może być przypisanych wiele albumów.
- Album musi być wydany przez jedno studio.
- Studio może wydać wiele albumów.

### 1. Określenie występujących encji.

#### Zbiór encji

Nazwa encji	Opis
Album	Zbiór piosenek wydanych jako spójna całość.
Piosenka	Utwór muzyczny wykonywany przez zespół
Kategoria	Zbiór albumów o podobnych cechach
Studio	Wytwórnia fonograficzna dystrybuująca albumy muzyczne

### 2. Ustalenie typów występujących związków.

#### Typy związków

Nazwa encji	Krotność	Związek	Nazwa encji	Krotność
Piosenka	1..1	należy	Album	1..1
Album	1..1	zawiera	Piosenka	1..*
Album	0..1	należy	Kategoria	0..*
Kategoria	0..1	zawiera	Album	0..*
Album	0..1	wydany przez	Studio	1..1
Studio	1..1	wydaje	Album	0..*

### 3. Określenie atrybutów odpowiadających poszczególnym encjom.

#### Atrybuty

Album					
Atrybut	Opis	Typ Danych	Dziedzina	Wielowartościowy	Złożony
Tytuł	Tytuł albumu	Łańcuch znaków	Max długość 100	nie	nie
Data Wydania	Data wydania albumu	Data	Data	nie	nie
Studio	Wytwórnia wydająca album	Encja	Encja	nie	tak
Piosenki	Lista piosenek znajdujących się na albumie	Encja	Encja	tak	tak

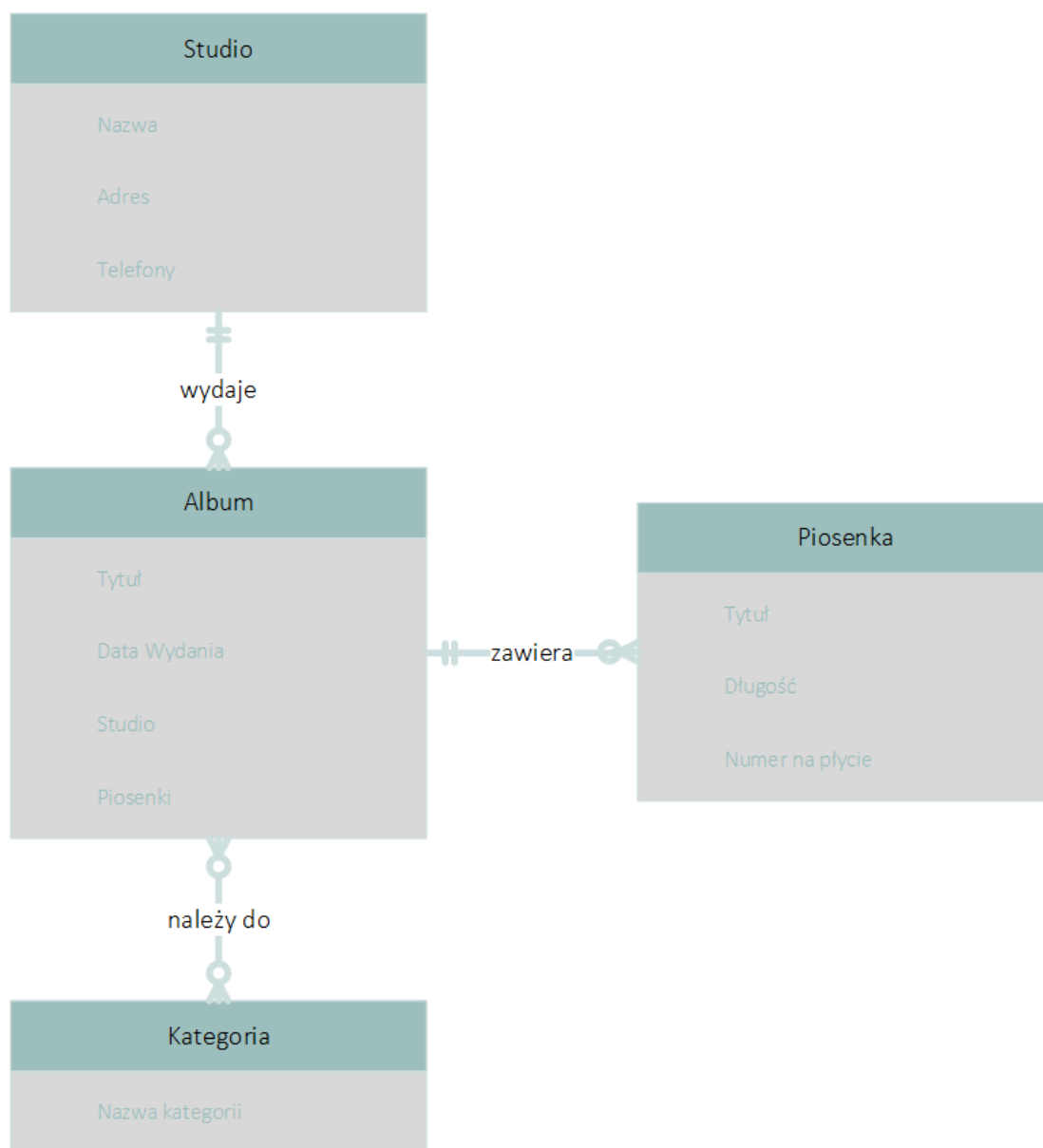
Studio					
Atrybut	Opis	Typ Danych	Dziedzina	Wielowartościowy	Złożony
Nazwa	Nazwa studio	Łańcuch znaków	Max długość 100	nie	nie
Adres	Adres siedziby studio	Łańcuch znaków	Max długość 200	nie	tak
Telefon	Numery telefoniczne studio	Łańcuch znaków	Telefon	tak	nie

...

### 4. Określenie dziedziny poszczególnych atrybutów.

#### Dziedziny

Nazwa dziedziny	Długość	Format	Dopuszczalny zbiór wartości
Data	X	DD-MM-RRR	Dowolna data nie późniejsza niż data bieżąca
Telefon	16	(XX) XXX-XXX-XXX	Cyfry od 0 do 9 zapisane w ustalony formacie

**5. Utworzenie diagramu ER.**



## *Model logiczny i fizyczny*

Modelowanie logiczne polega na utworzeniu modelu bazy danych na podstawie modelu koncepcyjnego. Efektem transformacji jest projekt logiczny bazy danych, który może zostać zaimplementowany w systemie zarządzania bazą danych obsługującym wskazany typ bazy danych (model fizyczny). Model fizyczny (np. kod SQL) może zostać automatycznie wygenerowany na podstawie modelu logicznego lub ręcznie zakodowany.

### *Zasady transformacji (przy założeniu relacyjnego modelu bazy danych)*

- Jedna encja → jedna tabela.
- Jeden atrybut → jedna kolumna (pole).
- Atrybuty nieatomiczne powinny zostać rozbite na kilka atomicznych pól.
- Dla każdej kolumny powinny zostać określone:
  - unikalna (w ramach tabeli) nazwa,
  - typ danych (dla niektórych również długość lub precyzja),
  - fakt, czy wartości kolumny są wymagane, czy nie,
  - fakt, czy wartości kolumny są unikatowe, czy nie,
  - więzy.
- Dla każdej tabeli powinny zostać zidentyfikowane klucze kandydujące i klucze podstawowe (np. atrybuty not null, unique, typ autonumerowany).
- Dla każdej relacji powinny zostać wskazane klucze obce oraz reguły usuwania i modyfikacji rekordów w powiązanych tabelach.
- Relacje „wiele do wielu” należy rozbić na relacje „jeden do wielu” i utworzyć tabele łączące.
- Dla pól wyliczeniowych wskazane jest utworzenie tabel walidacyjnych.
- Należy wyeliminować redundancję.
- Należy usunąć spacje w nazwach (poprzez zastąpienie ich znakami „\_” lub poprzez użycie stylu (Uper)CamelCase

### *Etapy transformacji (przy założeniu relacyjnego modelu bazy danych)*

1. Transformacja modelu koncepcyjnego do modelu logicznego:
  - a. Przekształcenie encji w tabele, dodanie tabel łączących i walidacyjnych
  - b. Przekształcenie atrybutów w pola, rozbitcie pól nieatomicznych na pojedyncze pola (ewentualnie utworzenie osobnej tabeli zawierającej pojedyncze pola z atrybutu złożonego)
  - c. Ustalenie kluczy kandydujących i kluczy głównych.
  - d. Aktualizacja relacji i dodanie kluczy obcych, ustalenie reguł usuwania i modyfikacji dla relacji
  - e. Utworzenie diagramu modelu logicznego<sup>1</sup>
2. Zakodowanie bazy danych w języku SQL – model fizyczny

Realizując etapy transformacji należy przestrzegać zasad transformacji.

<sup>1</sup> Bardzo często w systemach zarządzania bazami danych możliwe jest utworzenie modelu logicznego i na jego podstawie wygenerowanie kodu SQL bazy danych.

## Klucze

### Rodzaje kluczy

- **Klucz prosty** – klucz składający się z jednego pola.
- **Klucz złożony** – klucz składający się z więcej niż jednego pola.
- **Klucz naturalny** – klucz utworzony na potrzeby jednoznacznej identyfikacji rekordu na podstawie atrybutów istniejących w rzeczywistości (np. numer PESEL).
- **Klucz sztuczny** – klucz utworzony na potrzeby jednoznacznej identyfikacji rekordu, nieposiadający znaczenia biznesowego.
- **Superklucz (nadklucz)** – dowolny zestaw pól pozwalający na jednoznaczną identyfikację rekordu w bazie danych.
- **Klucz kandydujący (lub po prostu: klucz)** – minimalny superklucz, tzn. taki, którego żaden podzbiór pól nie jest wystarczający do jednoznacznej identyfikacji obiektu (używany do jednoznacznej identyfikacji rekordów). Klucz kandydujący powinien być:
  - **unikatowy** – zbiór wartości pól klucza kandydującego nie może się powtarzać w kontekście rekordów jednej tabeli, a wartości każdego z tych pól nie mogą być puste,
  - **stały** – zbiór wartości klucza kandydującego dla danego wystąpienia encji nie może się zmieniać,
  - **minimalny** – zbiór pól klucza kandydującego powinien zawierać wyłącznie pola niezbędne do jednoznacznej identyfikacji wystąpienia encji.
- **Klucz główny (podstawowy)** – wybrany przez użytkownika klucz kandydujący, służący do jednoznacznej identyfikacji rekordów w bazie danych.
- **Klucz alternatywny** – klucz kandydujący niebędący kluczem podstawowym.
- **Klucz obcy** – klucz przeniesiony z jednego typu encji do drugiego w celu identyfikacji rekordów po dokonaniu złączenia encji.

## Więzy

Więzy to logiczne warunki nakładające ograniczenia na dane.

- Więzy na poziomie atrybutów (np.:  $0 < \text{Wiek} < 140$ ).
- Więzy na poziomie krotek (np.:  $\text{Data\_urodzenia} < \text{Data\_zatrudnienia}$ ).
- Więzy na poziomie tabel.
- Więzy kluczy głównych i kluczy obcych (więzy spójności referencyjnej).
- Więzy niezwiązane z konkretnym atrybutem, krotką bądź tabelą (np.:  $\text{Suma wszystkich zarobków pracowników działu X} = \text{Fundusz płac działu X}$ ).

**Przykład**

Transformacja modelu koncepcyjnego dla biblioteki płyt z muzyką przedstawionego w temacie 3.

**a. Przekształcenie encji w tabele, dodanie tabel łączących i walidacyjnych****Tabele**

Nazwa encji	Nazwa tabeli	Rodzaj tabeli
Album	albumy	Tabela z danymi
Piosenka	piosenki	Tabela z danymi
Kategoria	kategorie	Tabela z danymi
Studio	studia	Tabela z danymi
	adresy	Tabela podzbiór
	studiaTelefony	
	kategorieAlbumow	Tabela łącząca

**b. Przekształcenie atrybutów w pola****Pola**

albumy			
Pole	Typ Danych	Unikatowy	Wymagalny
tytul	Varchar(100)	nie	tak
dataWydania	Data	nie	tak
studio	(uzupełnić po wyborze klucza podstawowego w tabeli studia)	nie	tak

studia			
Pole	Typ Danych	Unikatowy	Wymagalny
nazwa	Varchar(100)	nie	tak
adres	(uzupełnić po wyborze klucza podstawowego w tabeli adresy)	nie	tak

studiaTelefony			
Pole	Typ Danych	Unikatowy	Wymagalny
studio	(uzupełnić po wyborze klucza podstawowego w tabeli studia)	nie	tak
telefon	varchar(20)	tak	tak

...

**c. Ustalenie kluczy kandydujących i kluczy głównych.**
**Klucze**

Nazwa tabeli	Klucze kandydujące	Klucz główny	Uzasadnienie wyboru
adresy	ulica + numerBudynku + numerLokalu + miejscowosc, ulica + numerBudynku + numerLokalu + kodPocztowy	idAdresu	Użycie wielopolowego klucza podstawowego złożonego z pól tekstowych może znacznie utrudnić pracę z bazą danych.
albumy	tytul + dataWydania	idAlbumu	Wielopolowy klucz podstawowy nie będzie praktyczny jako klucz obcy w licznych relacjach jakie tworzy tabela albumy.
studia	idAdresu + telefon	idStudia	Adres i/lub telefon studia mogą ulec zmianie.
studiaTelefony	telefon	telefon	Jedyne pole, które spełnia warunki klucza podstawowego
kategorie	nazwa	nazwa	Jedyne pole, spełnia warunki klucza podstawowego
kategorieAlbumow	idAlbumu + nazwa	idAlbumu + nazwa	Tabela łącząca kategorie i albumy
piosenki	nrPiosenki + idAlbumu	nrPiosenki + idAlbumu	Użycie dwupolowego klucza pozwoli na przechowanie informacji o albumie oraz o kolejności piosenek na albumie

Pola typu Id pojawiają się już na etapie modelu logicznego. Są to klucze sztuczne i ich użycie nie zawsze jest konieczne i właściwe.

Już klucz kandydujący musi spełniać warunek unikatowości więc niewłaściwym jest użycie pola nie unikatowego jako klucza kandydującego i odrzucenie go ze względu na tą cechę.

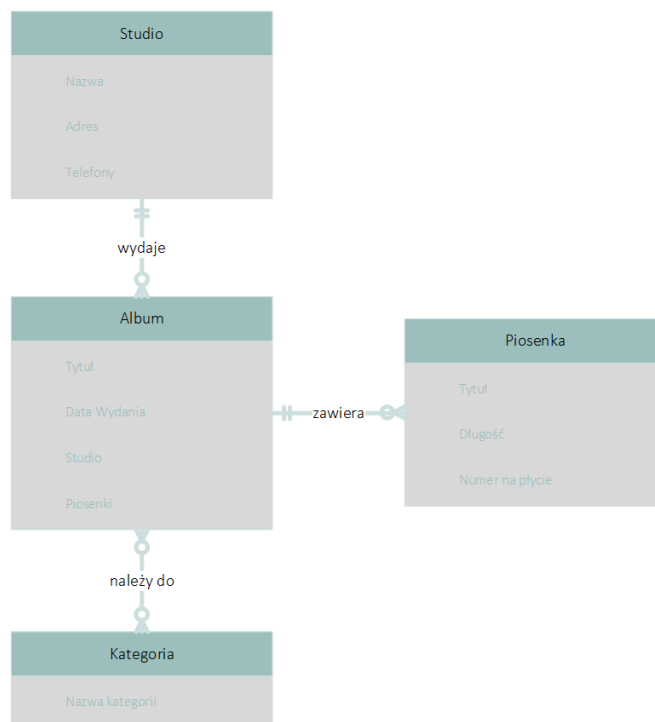
**d. Aktualizacja relacji, dodanie kluczy obcych oraz ustalenie reguł usuwania i modyfikacji dla relacji**

Nazwa tabeli podstawowej	Pole	Krotność	Nazwa tabeli związanej	Pole	Krotność	Reguła usuwania	Reguła modyfikacji
albumy	idAlbumu	1..1	piosenki	idAlbumu	0..*	restrykcyjna	kaskadowa
albumy	idAlbumu	1..1	kategorieAlbumow	idAlbumu	0..*	restrykcyjna	kaskadowa
kategorie	nazwa	1..1	kategorieAlbumow	nazwaKategorii	0..*	restrykcyjna	kaskadowa
studia	idStudia	1..1	albumy	idStudia	0..*	restrykcyjna	kaskadowa
studia	idStudia	1..1	studiaTelefony	idStudia	1..*	kaskadowa	kaskadowa
adresy	idAdresu	1..1	studia	idAdresu	0..*	restrykcyjna	restrykcyjna

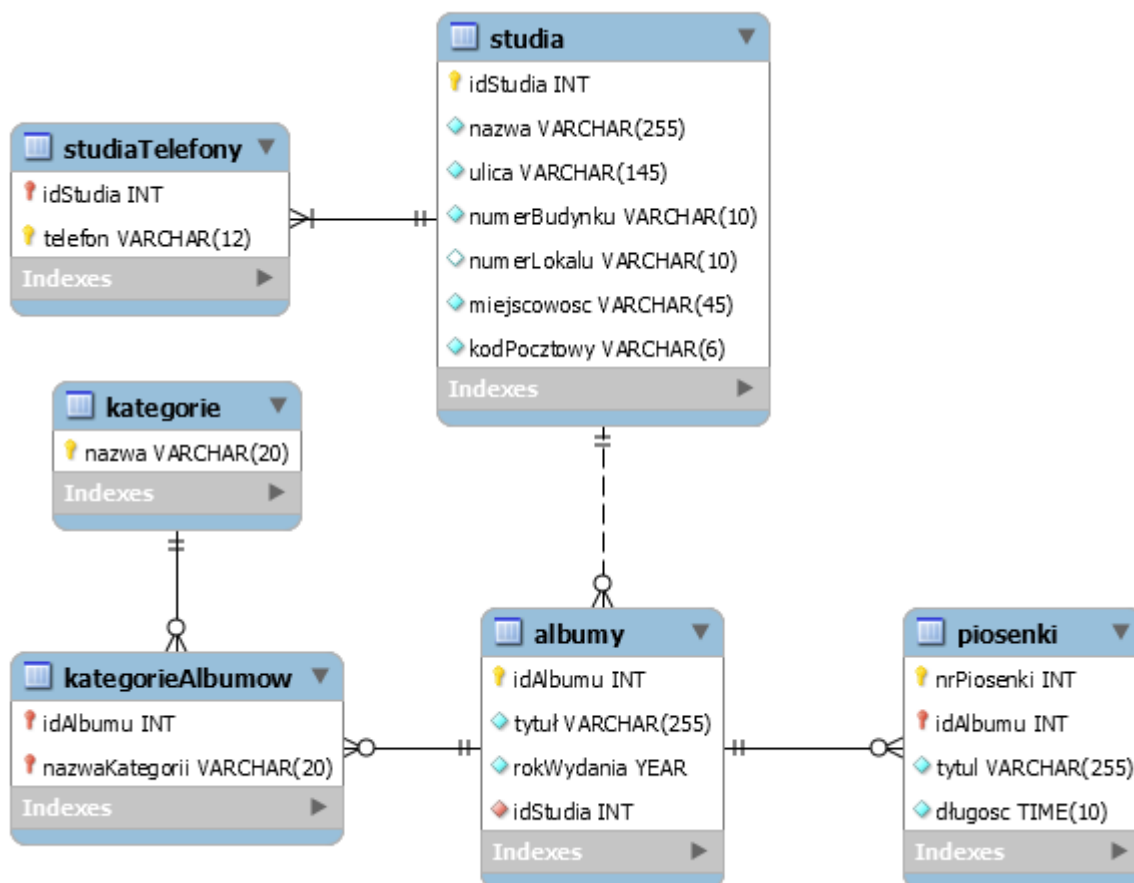
Nazwy pól tworzących relację w tabeli podstawowej i w tabeli związanej nie muszą być takie same.

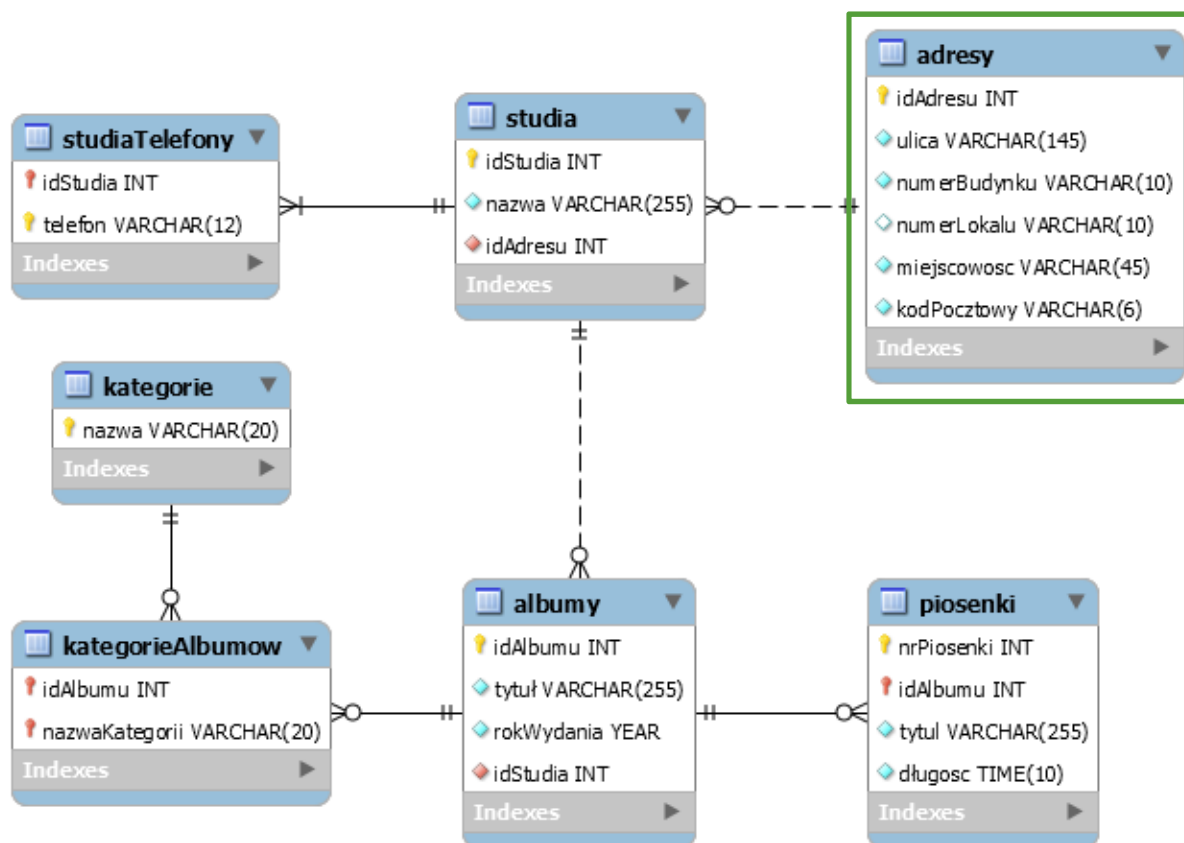
### e. Transformacja diagramu

#### Model koncepcyjny



#### Odpowiadający mu model logiczny





### Zmiany dokonane przy transformacji

- Encje zostały przekształcone na tabele → nazwy zostały zapisane w liczbie mnogiej, z małej litery
- Atrybuty encji zostały przekształcone w pola → nazwy zapisano stylem CamelCase, dodano typy danych, długości pól i precyzję, określono wymagalność i unikatowość, atrybut złożony Adres został rozbity na pola i finalnie wydzielony do osobnej tabeli adresy, a atrybut wielowartościowy Telefony został przeniesiony do osobnej tabeli studiaTelefony.
- Utworzone zostały klucze podstawowe i klucze obce. Tabele zostały połączone relacjami z wykorzystaniem kluczy. Dla każdej relacji określono reguły usuwania i modyfikacji.
- Relacja wiele do wielu pomiędzy kategoriami i albumami została rozbita na dwie relacje jeden do wielu. Została utworzona tabela łącząca kategorieAlbumow

### Kod SQL

Moodle/Modelowanie danych/ Model logiczny i fizyczny – pliki/  
baza wygenerowana na podstawie modelu fizycznego.sql

**Podstawowe różnice między modelem koncepcyjnym a logicznym**

<b>Model koncepcyjny</b>	<b>Model logiczny relacyjnej bazy danych</b>
Nazwy encji w liczbie pojedynczej z wielkiej litery	Nazwy tabel w liczbie mnogiej, najczęściej z małej litery
Występuje relacja wiele do wielu	Relacje wiele do wielu muszą zostać rozbite na dwie relacje jeden do wielu połączone tabelą łączącą
Encje asocjacyjne przechowujące informacje o związkach posiadające własne atrybuty	Tabele łączące
Nazwy relacji	Powiązania po poziomie kluczy (podstawowy-obcy)
Atrybuty złożone i wielowartościowe	Rozbicie atrybutów złożonych na pola i wydzielenie atrybutów wielowartościowych do osobnej tabeli
Atrybuty z dziedzina w postaci listy wyliczeniowej	Tabele walidacyjne

**ORM – Definicja i przykład**

Mapowanie obiektowo-relacyjne (ang. Object-Relational Mapping ORM) – sposób odwzorowania obiektowej architektury systemu informatycznego na bazę danych (lub inny element systemu) o relacyjnym charakterze. Implementacja takiego odwzorowania stosowana jest m.in. w przypadku, gdy tworzony system oparty jest na podejściu obiektowym, a system bazy danych operuje na relacjach<sup>2</sup>.

<sup>2</sup> [https://pl.wikipedia.org/wiki/Mapowanie\\_obiektowo-relacyjne](https://pl.wikipedia.org/wiki/Mapowanie_obiektowo-relacyjne)



**Diagram klas UML****Definicje klas z anotacjami (Doctrine2 ORM)**

Moodle/Modelowanie danych/ Model logiczny i fizyczny – pliki/

Definicje klas z anotacjami potrzebnymi do wygenerowania bazy danych

**Reguły ORM w plikach .yaml lub .xml (Doctrine2 ORM)****Konfiguracje .yaml**

Moodle/Modelowanie danych/ Model logiczny i fizyczny – pliki/

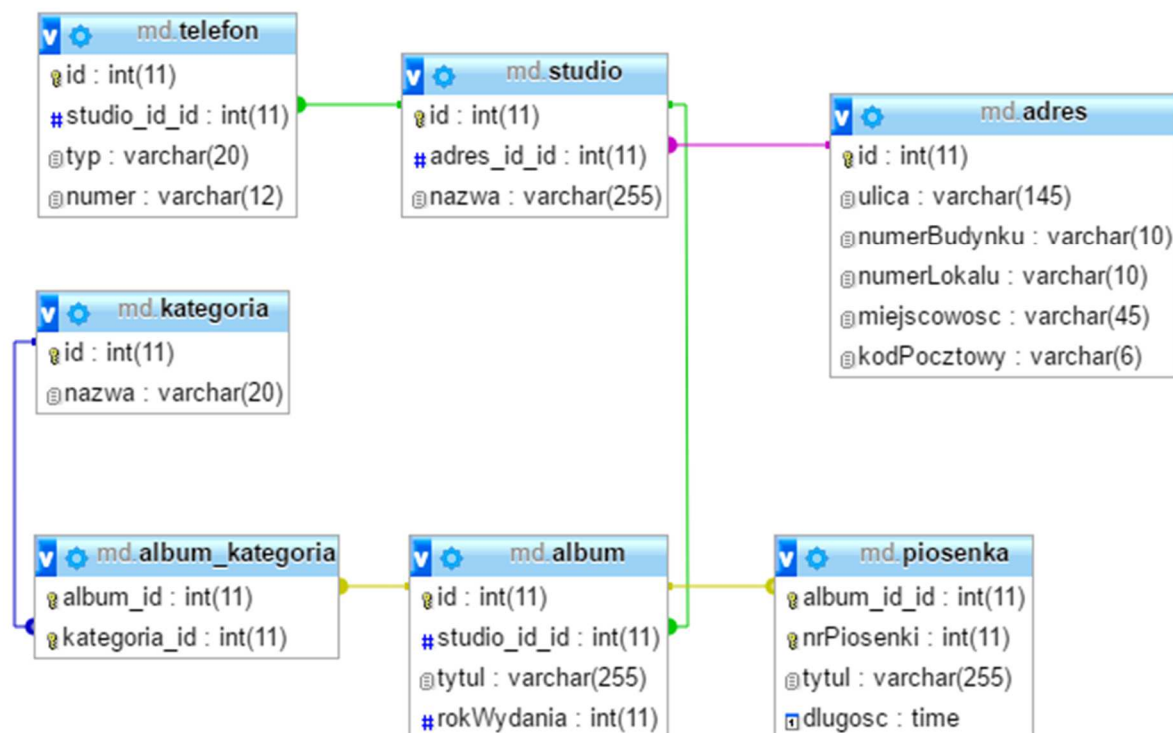
Pliki .yaml z konfiguracją bazy danych

**Konfiguracje .xml**

Moodle/Modelowanie danych/ Model logiczny i fizyczny – pliki/

Pliki .xml z konfiguracją bazy danych

### Relacyjny model bazy danych



### Kod SQL

Moodle/Modelowanie danych/ Model logiczny i fizyczny – pliki/  
baza wygenerowana przez ORM.sql

# Normalizacja – 1NF, 2NF

## Zależność funkcyjna

- Zależność funkcyjna: jeżeli dwie krotki relacji są zgodne co do atrybutów  $A_1, A_2, \dots, A_n$ , to muszą także zgadzać się we wszystkich pozostałych atrybutach  $B_1, B_2, \dots, B_m$ .
- „Jeżeli znam X, to znam Y” (np.: PESEL → Nazwisko, Kod paskowy → Nazwa towaru, cena).
- „X determinuje Y” lub „Y zależy od X” (np.: PESEL determinuje Nazwisko; Nazwa towaru i cena zależą od Kodu paskowego).
- Innymi słowy zależność funkcyjna występuje wtedy, kiedy wartość w jednym polu tabeli zależy od wartości w innym polu tabeli.

## Anomalie

- Redundancja – ta sama informacja zostaje wprowadzona wielokrotnie.
- Anomalie przy aktualizacji – informacja zostaje uaktualniona tylko w niektórych miejscach, np.: zmiana adresu studenta na nowy, jeśli informacja o studencie i jego adresie trzymana jest w kilku miejscach.
- Anomalie przy usuwaniu – wraz z usunięciem ostatniego wiersza szczegółowego znika informacja ogólna, np.: gdy informacje o adresie studenta trzymane są w jednej tabeli wraz z przedmiotami, na które jest zarejestrowany, po zakończeniu sesji adres ten by zniknął.
- Anomalie przy wstawianiu – brak możliwości wstawienia rekordu z uwagi na więzy integralności, np. gdy informacje o adresie studenta trzymane są w jednej tabeli wraz z przedmiotami, na które jest zarejestrowany, podczas próby dopisania nowego studenta, który nie jest jeszcze zapisany na żaden przedmiot, system bazy danych uniemożliwi wprowadzenie rekordu, gdyż przedmiot jest atrybutem wymagany.

## Cele normalizacji

- Wyeliminowanie redundancji.
- Wyeliminowanie anomalii.
- Dekompozycja - rozkład relacji na mniejsze, które nie mają redundancji i anomalii.

## Własności normalizacji

- Zachowanie atrybutów
- Zachowanie informacji
- Zachowanie zależności funkcyjnych w relacjach.

**Definicja normalizacji**

Normalizacja to proces odpowiedniej organizacji danych, którego celem jest wyeliminowanie nadmiarowych danych oraz niespójnych zależności między danymi (anomalii).

**Pierwsza postać normalna (1NF)**

- Wszystkie użyte dziedziny zawierają wyłącznie wartości atomowe (wszystkie dane są atomiczne).
- Reguły:
  - Nazwa każdej kolumny powinna być unikatowa,
  - Porządek kolumn oraz krotek jest nieistotny (nie ma znaczenia).
  - Każda kolumna powinna posiadać pojedynczy typ danych (wartości krotek w tej kolumnie nie mogą być różnych typów).
  - Żadne dwie krotki nie mogą posiadać wszystkich atrybutów identycznych (tabela musi posiadać klucz główny).
  - Każda kolumna powinna zawierać pojedynczą wartość (np.: niedozwolona jest kolumna Aktor przechowująca wartości „George, Clooney”, „Robert De Niro”, „Al Pacino”, etc.).
  - Kolumny nie powinny zawierać powtarzających się grup (atrybuty nie mogą być wielowartościowe, np.: niedozwolony jest zestaw kolumn „Phone1”, „Phone2”, „Phone3”, etc.).

**Druga postać normalna (2NF)**

- Reguły:
  - Tabela jest w pierwszej postaci normalnej (1NF) **ORAZ**
  - Każda kolumna nienależąca do klucza głównego zależy od wszystkich kolumn (razem), składających się na klucz główny
  - **LUB** dowolna kolumna nienależąca do klucza głównego nie zależy od części klucza głównego
  - **LUB** każdy atrybut spoza klucza głównego jest od niego w pełni funkcyjnie zależny (czyli nie jest zależny od jego części).
- Pytanie pomocnicze: „czy ta kolumna może istnieć bez któregoś ze składników klucza?”.
- Naruszenie drugiej postaci normalnej wymaga istnienia klucza złożonego i atrybutu niekluczowego.
- Atrybut taki musi być zależny funkcyjnie od fragmentu tego klucza.

# Normalizacja – 3NF, BCNF

## Przechodnia zależność funkcyjna

- Występuje, gdy dowolna niekluczowa kolumna zależy od innej niekluczowej kolumny.
- W relacji  $R(A, B, C)$  zachodzą zależności funkcyjne  $A \rightarrow B$ ,  $B \rightarrow C$ . Wnioskujemy stąd, że zachodzi także zależność  $A \rightarrow C$ .

## Trzecia postać normalna (3NF)

- Reguły:
  - Tabela jest w drugiej postaci normalnej (2NF) **ORAZ**
  - Dowolna kolumna nienależąca do klucza głównego nie zależy od innej kolumny nienależącej do klucza głównego
  - **LUB** każdy niekluczowy atrybut jest bezpośrednio zależny od klucza głównego
  - **LUB** tabela nie posiada żadnych przechodnich zależności funkcyjnych<sup>1</sup>.
  - **LUB** żaden atrybut spoza klucza głównego nie jest od niego przechodnio zależny (czyli jest zależny bezpośrednio).
- Pytanie pomocnicze: „Czy ta kolumna jest zależna od innej kolumny niekluczowej?”

## Postać BCNF

- Boyce-Codd Normal Form.
- Reguły:
  - Tabela jest w trzeciej postaci normalnej (3NF) **ORAZ**
  - Każdy determinant jest kluczem kandydującym
  - **LUB** wszystkie zależności kluczowe są zależnościami od kluczy.

**Determinant** to atrybut, który przynajmniej częściowo determinuje inny atrybut.

---

<sup>1</sup> Więcej o zależnościach częściowych i funkcyjnych:  
<http://edu.pjwstk.edu.pl/wyklady/rbd/scb/wyklad5/norm.htm>

# Normalizacja – 4NF, 5NF

## Zależność wielowartościowa

- Zbiór atrybutów Y jest zależny wielowartościowo od zbioru X gdy z każdą konfiguracją wartości atrybutów z X jest związany zbiór konfiguracji wartości z Y niezależnie od wartości pozostałych atrybutów.
- Występuje, gdy zbiór wartości kolumny X, zależnej od każdej wartości kolumny Y jest niezależny od wszystkich innych kolumn.

## Zależność trywialna

- Jeśli zbiór złożony z atrybutów B jest podzbiorem zbioru złożonego z atrybutów A.

## Zależność nietrywialna

- Jeśli co najmniej jeden atrybut zbioru złożonego z atrybutów B nie jest podzbiorem zbioru złożonego z atrybutów A.

## Czwarta postać normalna (4NF)

- Reguły:
  - Tabela jest w postaci BCNF **ORAZ**
  - usunięto nietrywialne wielowartościowe zależności funkcyjne
  - **LUB** dla każdej nietrywialnej zależności wielowartościowej  $X \twoheadrightarrow Y$ , X jest nadkluczem.

## Zależność połączeniowa (złączeniowa)

- Jeśli R jest relacją, a A, B, ..., Z są dowolnymi podzbiórmi zbioru atrybutów R, to R spełnia zależność połączeniową JD (join dependency):  
 $*(A, B, \dots, Z)$   
wtedy i tylko wtedy, gdy R jest równe połączeniu swoich projekcji na A, B, ..., Z.
- Niech  $R = \{R_1, R_2, \dots, R_p\}$  oznacza zbiór schematów relacji zdefiniowany na zbiorze atrybutów  $U = \{A_1, A_2, \dots, A_n\}$  takich, że  $R_1 \cup R_2 \cup \dots \cup R_p = U$ .  
Mówimy, że relacja  $r(U)$  spełnia zależność połączeniową, oznaczaną przez JD  $[R_1, R_2, \dots, R_p]$ , jeżeli można ją zdekomponować bez utraty informacji na podrelacje  $r_1(R_1), r_2(R_2), \dots, r_p(R_p)$ .<sup>1</sup>

<sup>1</sup> <http://tomasz.kubik.staff.iiar.pwr.wroc.pl/dydaktyka/RelacyjneBazyDanych/>

**Piąta postać normalna (5NF)**

- Reguły:
  - Tabela jest w postaci 4NF **ORAZ**
  - nie zawiera połączeniowej zależności funkcyjnej
  - **LUB** każda zależność połączeniowa jest implikowana kluczami kandydującymi.

**Postacie normalne**

Postać normalna	Operacje	Uwagi
<b>1NF</b>	usunięcie danych nieatomicznych	Rozbicie pól złożonych na szczegółowe pola, a pól wielowartościowych na osobne rekordy (chwilowa rozbudowa tabeli i redundancja).
<b>2NF</b>	usunięcie niepełnej zależności funkcyjnej	Sprawdzone są tylko tabele z wielopolowym kluczem podstawowym. Badanie, czy pole niekluczowe zależy od całego klucza podstawowego, czy tylko od jego składowej. W tym drugim przypadku wydzielana jest tabela, której kluczem podstawowym jest część klucza tabeli oryginalnej. Przenoszone są do niej wszystkie pola zależne od tego determinantu.
<b>3NF</b>	usunięcie przechodniej zależności funkcyjnej	Badanie zależności pomiędzy polami nie należącymi do klucza podstawowego. Jeśli jakieś pole zależy od innego pola niekluczowego, a dopiero to pole zależy od klucza podstawowego, to wydzielana jest tabela, której kluczem podstawowym jest determinant. Przenoszone są do niej wszystkie pola zależne od tego determinantu, a nie bezpośrednio od klucza podstawowego.
<b>BCNF</b>		Każdy determinant (atrybut, który przynajmniej częściowo determinuje inny atrybut) jest kluczem kandydującym

Postać normalna	Operacje	Uwagi
4NF	usunięcie nietrywialnej wielowartościowej zależności funkcyjnej	Jeśli dwa (lub więcej) pola zależą od innego pola, ale między nimi nie ma zależności, to należy podzielić tą tabelę na mniejsze przenosząc do nich pole powodujące zależność wielowartościową wraz z pojedynczymi polami zależnymi od niego.
5NF	usunięcie połączeniowej zależności funkcyjnej	Jeśli dwa (lub więcej) pola zależą od innego pola, a między nimi samymi również występują zależności, to należy podzielić tą tabelę na mniejsze przenosząc do nich pola parami na zasadzie „każde z każdym”. Jedna z tabel pełni z reguły rolę słownika – ogranicza wartości dopuszczalne w pozostałych tabelach.

Każda kolejna postać normalna musi spełniać warunki wszystkich poprzednich.