



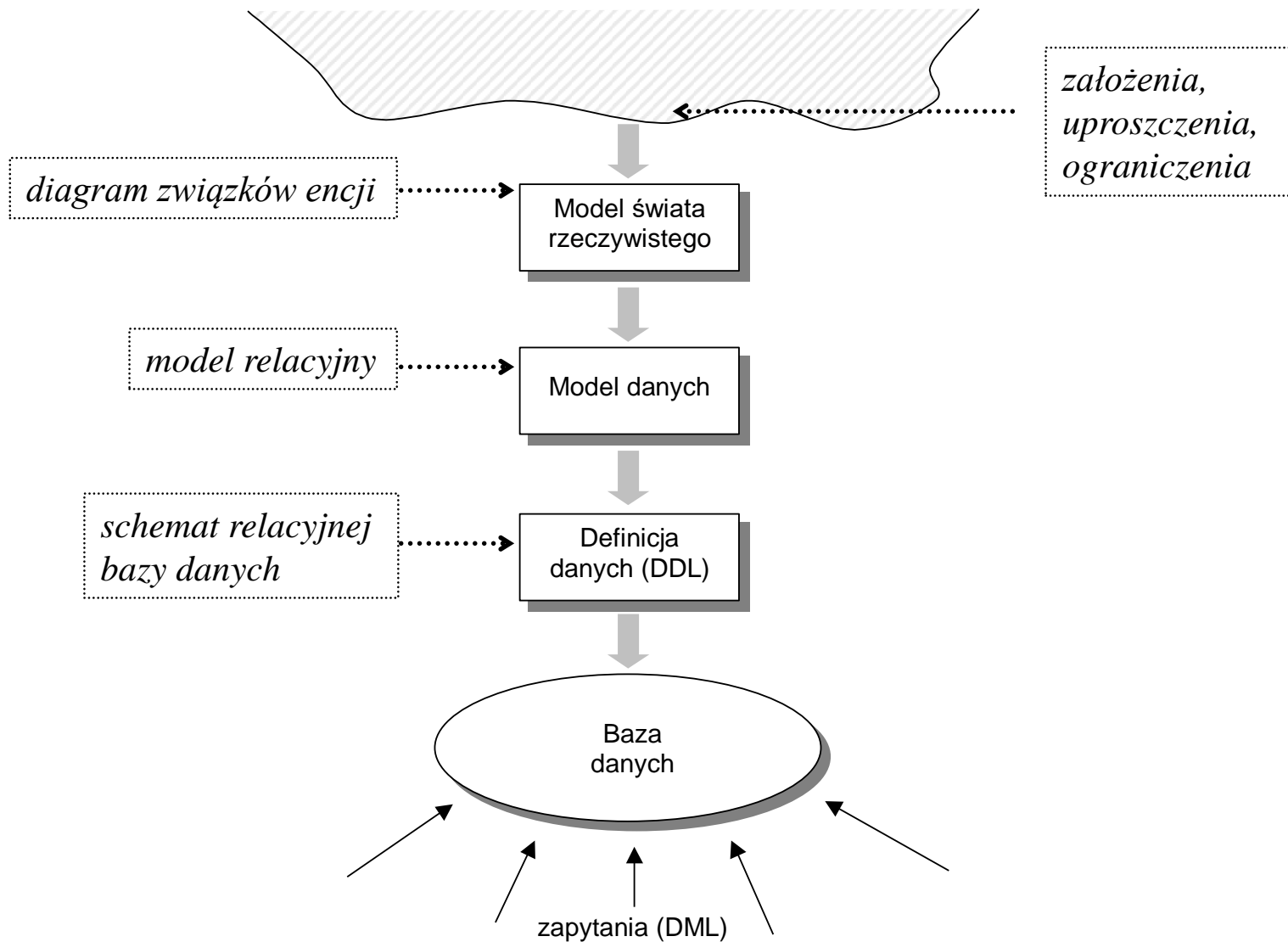
Bazy Danych

Dr inż. Przemysław Falkowski-Gilski

Wydział Elektroniki, Telekomunikacji i Informatyki



- Wykład opracowany na podstawie materiałów pomocniczych z Baz Danych autorstwa: K. Goczyły, A. Landowskiej, M. Piechówka oraz A. Ramsa
- Lektura materiału nie zastępuje obecności na wykładzie
- Wykorzystanie materiałów w innym celu oraz ich rozpowszechnianie jest zabronione



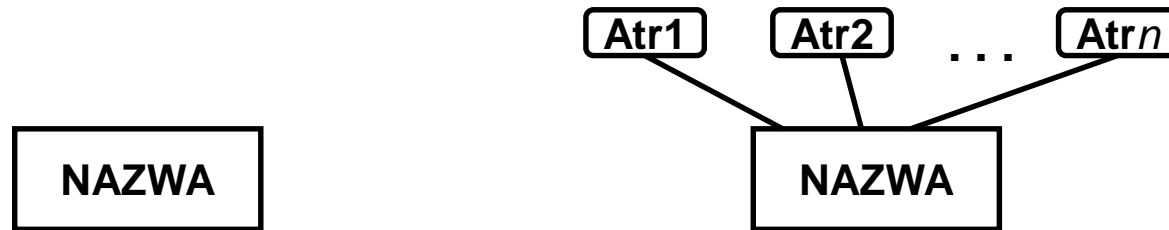


- Pojęcie bazowe (niedefiniowalne); podstawową cechą encji jest to, że jest rozróżnialna od innych encji
- Przykłady:
 - rzeczy – osoba, samochód, tablica, książka
 - zdarzenia – wypadek samochodowy, lot samolotu, egzamin
 - pojęcia – uczucie, wyobrażenie

Każda encja jest inna. W celu uproszczenia analizy świata rzeczywistego, grupujemy (klasyfikujemy) podobne do siebie encje w zbiory encji (stosujemy zasadę abstrakcji).



- Grupa podobnych do siebie (wg określonych kryteriów) encji



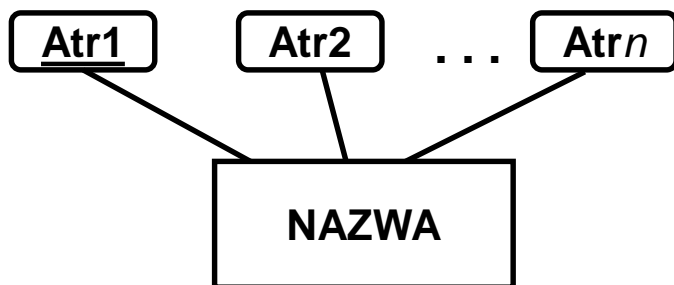
- Przykłady:
 - zbiór wszystkich osób
 - zbiór wszystkich kobiet zamieszkałych w Gdańsku
 - zbiór wszystkich wypadków samochodowych na skrzyżowaniu X
- Zbiory encji posiadają jednoznaczną nazwę oraz zbiór atrybutów



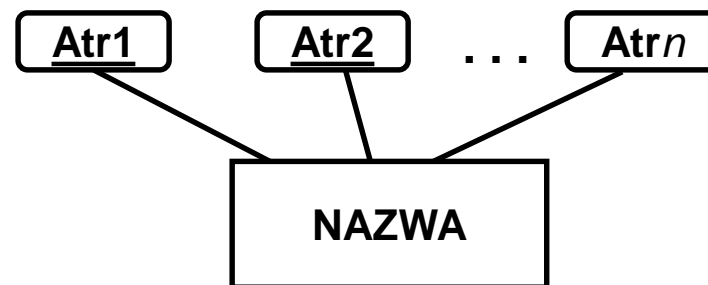
- Encje w zbiorze encji są rozróżniane przez wartości atrybutów
- Własności atrybutów zbioru encji:
 - liczba atrybutów jest ustalona (≥ 1)
 - typy atrybutów są proste (wartości są niepodzielne, czyli atomowe)
 - nazwy atrybutów są różne (jednoznaczne)



- Atrybuty:
 - reprezentują podstawowe własności/cechy encji
 - cechy te będą przechowywane w bazie danych
 - mogą być obowiązkowe lub opcjonalne
 - powinny być typu prostego
- Definicja atrybutu encji:
 - nazwa
 - dziedzina
 - typ danych i maksymalny rozmiar lub/i precyzja
 - zbiór dozwolonych wartości
 - zakres dozwolonych wartości
 - dozwolone/niedozwolone wartości puste
 - opcjonalnie unikatowość wartości

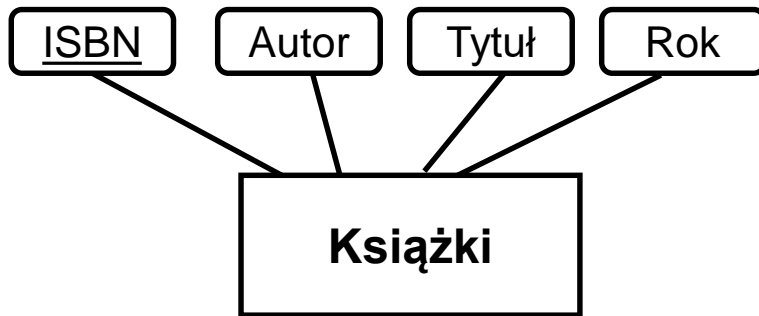


Klucz prosty: Atr1

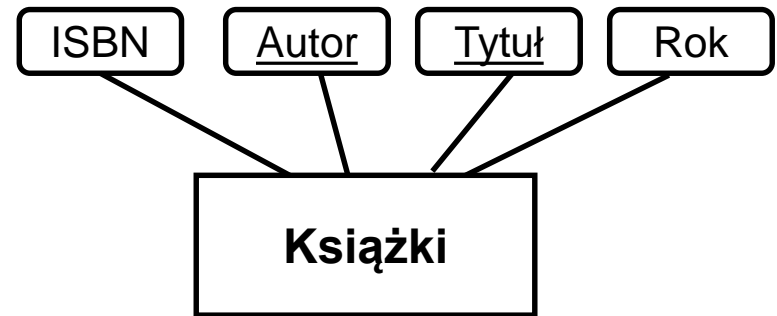


Klucz złożony: (Atr1, Atr2)

- Własności klucza zbioru encji:
 - jednoznaczność – nie istnieją takie 2 encje w zbiorze encji, które mają takie same wartości atrybutów kluczowych; innymi słowy wartość klucza jednoznacznie identyfikuje encję
 - minimalność (dot. kluczy złożonych) – nie można z klucza usunąć żadnego atrybutu bez naruszenia wymogu jednoznaczności; innymi słowy każda składowa klucza złożonego jest potrzebna.



Klucz: ISBN

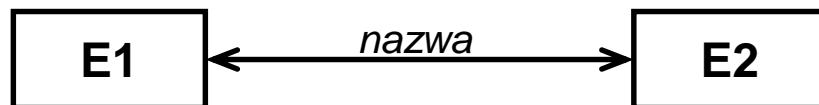


Klucz: (Autor, Tytuł)

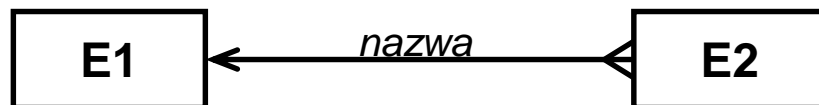
- Jeden z kluczy wybieramy jako klucz główny (podstawowy), drugi klucz (lub pozostałe klucze, jeśli jest ich więcej) są kluczami alternatywnymi
- Tam, gdzie nie rodzi to niejednoznaczności: pojęcia „klucz” i „klucz główny” będziemy utożsamiać



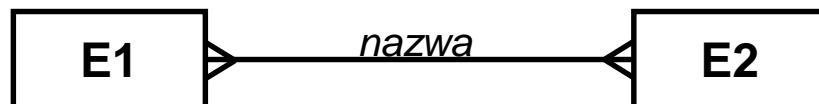
- Związek REL pomiędzy zbiorami encji E_1, E_2, \dots, E_n :
 - $REL(E_1, E_2, \dots, E_n)$
jest to zbiór krotek n -elementowych (e_1, e_2, \dots, e_n)
 - zbiory encji E_i nie muszą być różne!
- Przykłady:
 - związek TEACH (TEACHERS, STUDENTS) zawiera wszystkie pary (t, s) takie że nauczyciel t uczy studenta s
 - związek MOTHER (PERSONS, PERSONS) zawiera wszystkie pary (p_1, p_2) takie że osoba p_1 jest matką osoby p_2
 - związek TEACHING (TEACHERS, STUDENTS, SUBJECTS) zawiera wszystkie trójki (t, s, b) takie, że nauczyciel t uczy studenta s przedmiotu b
- TEACH i MOTHER to związki binarne ($n = 2$)
- TEACHING jest związkiem potrójnym ($n = 3$)



1:1 (jeden do jeden)

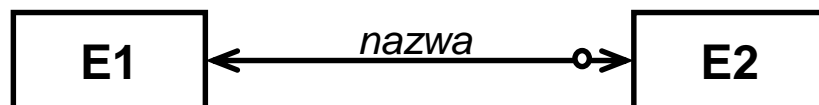


1:n (jeden do wiele; $n \geq 1$)

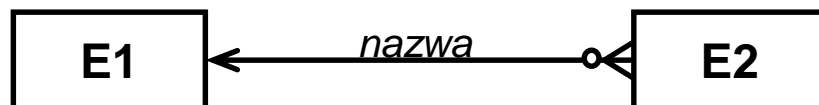


n:m (wiele do wiele; $n, m \geq 1$)

Można też wskazać opcjonalność związku, np.:



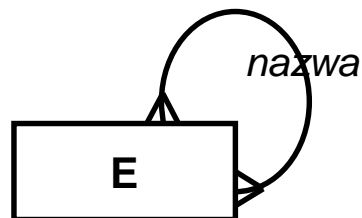
1:0..1



1:0..n



Inna notacja:

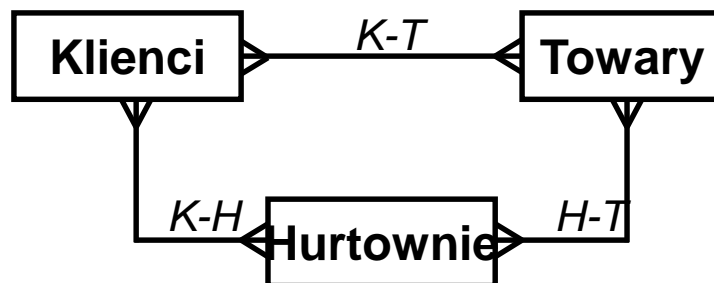


Uwaga:

Zbiory encji pozostające w danym związku nie muszą być różne!

Zapis słowny związków pomiędzy encjami:

Nazwa (E1, E2)



Klienci = { (Nowak), (Kowalski) }

Towary = { (książki), (komputery) }

Hurtownie = { (Mała), (Nowa) }

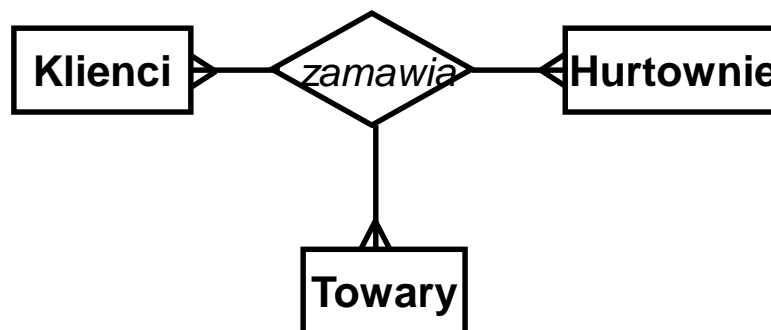
K-T = { (Nowak, książki), (Kowalski, książki),
(Nowak, komputery), (Kowalski, komputery) }

K-H = { (Nowak, Mała), (Kowalski, Mała),
(Nowak, Nowa), (Kowalski, Nowa) }

H-T = { (Mała, książki), (Nowa, książki),
(Mała, komputery), (Nowa, komputery) }

• Problem:

- klienci zamawiają w hurtowniach towary
- dowolny towar może być zamówiony w dowolnej hurtowni przez dowolnego klienta
- ale w której hurtowni Nowak zamawiał książki?
- w której hurtowni Kowalski zamawiał komputery?



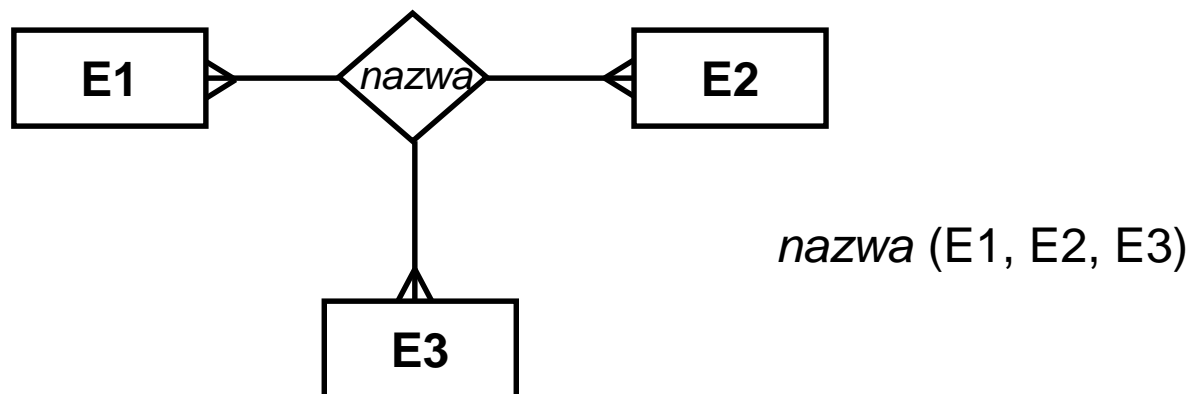
Zamawia (Klienci, Hurtownie, Towary) jest **związkiem potrójnym** (stopnia 3)

Przykładowe *wystąpienie* tego związku:

zamawia = { (Nowak, Mała, książki), (Nowak, Nowa, książki),
(Nowak, Nowa, komputery), (Kowalski, Mała, komputery),
(Kowalski, Mała, książki) }

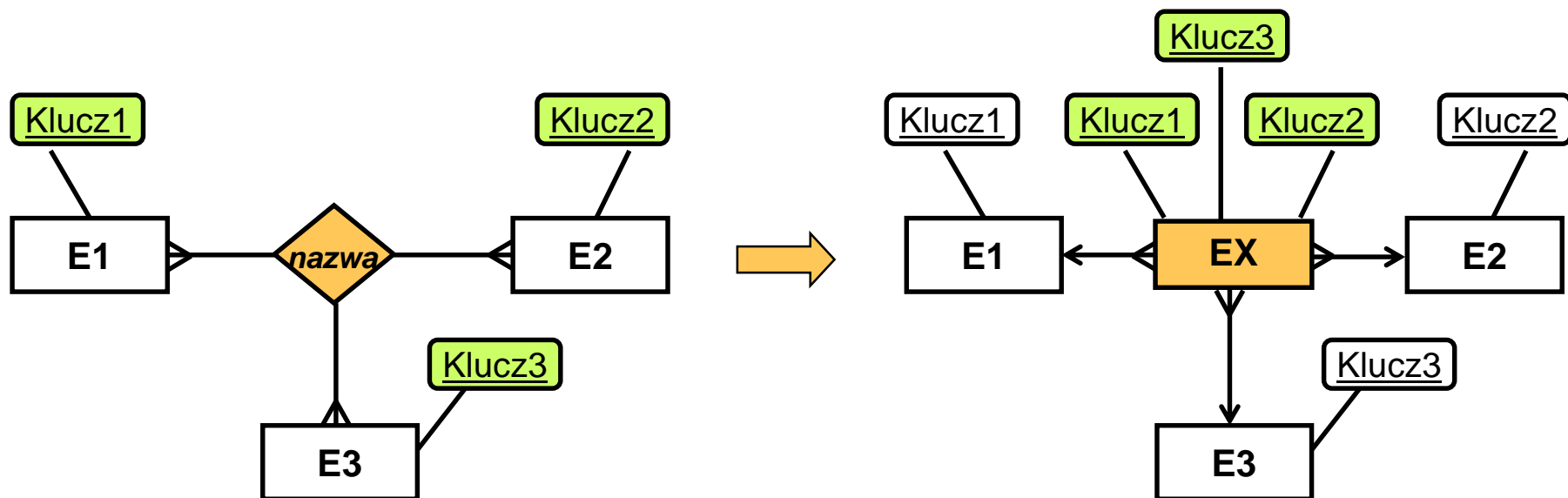
Pytanie:

Gdzie zamieścić takie ważne atrybuty jak: data zamówienia, zamawiana ilość?



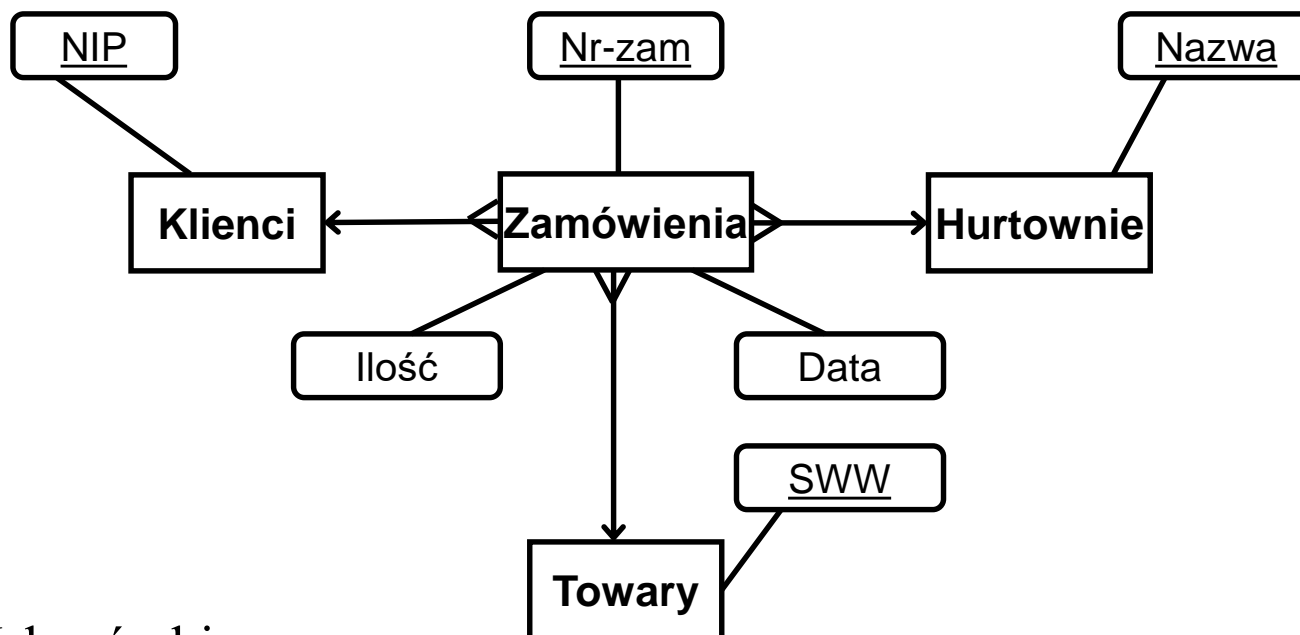
Liczności mogą być różne dla różnych zbiorów encji.

Uwaga: **Stopień** związku nazywamy też **krotnością** związku



Zbiór encji **EX** reprezentuje związek *nazwa*.

Kluczem zbioru encji **EX** jest złożenie kluczy encji pozostających w związku *nazwa*.

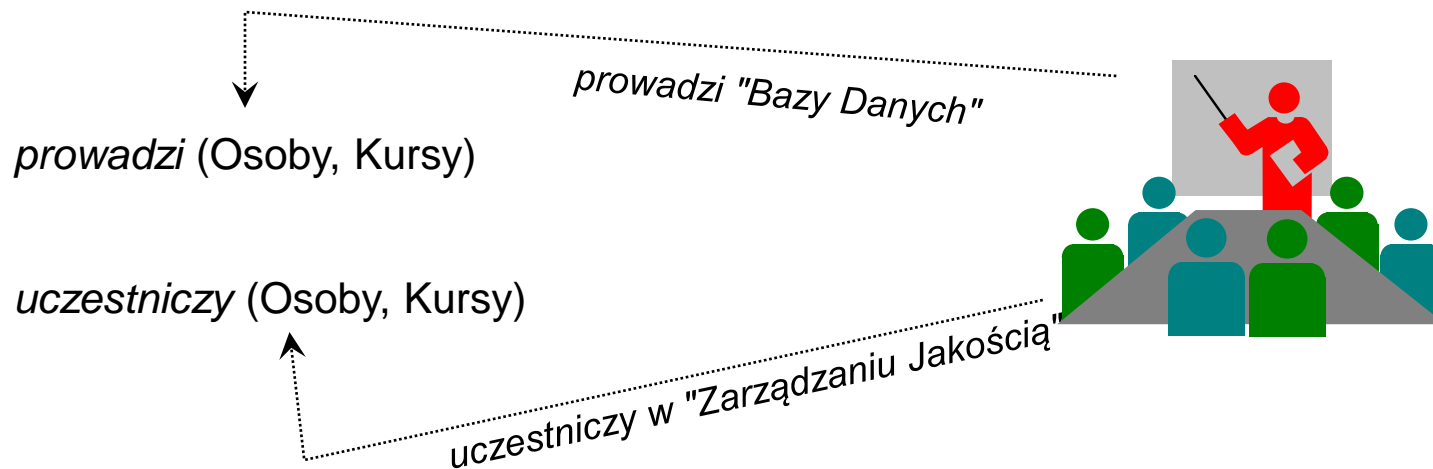
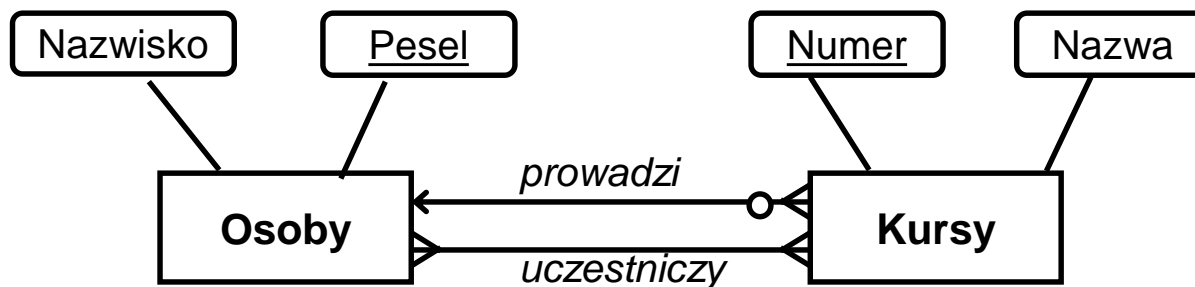


- Wskazówki:

- jeśli klucz encji reprezentującej związek staje się długi (wieloczłonowy), warto zastąpić go kluczem prostym typu identyfikator (np. Nr-zam)
- związki pomiędzy zbiorami encji są reprezentowane symbolami graficznymi (strzałkami)
- sposób implementacji tych związków nie musi być przedstawiony na modelu

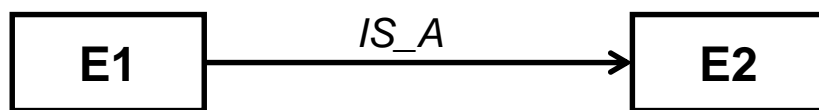


Diagram związków encji

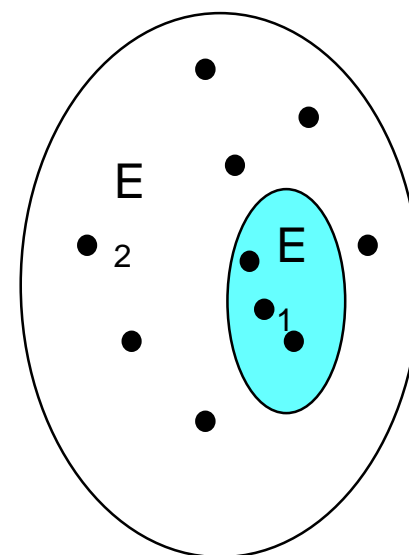


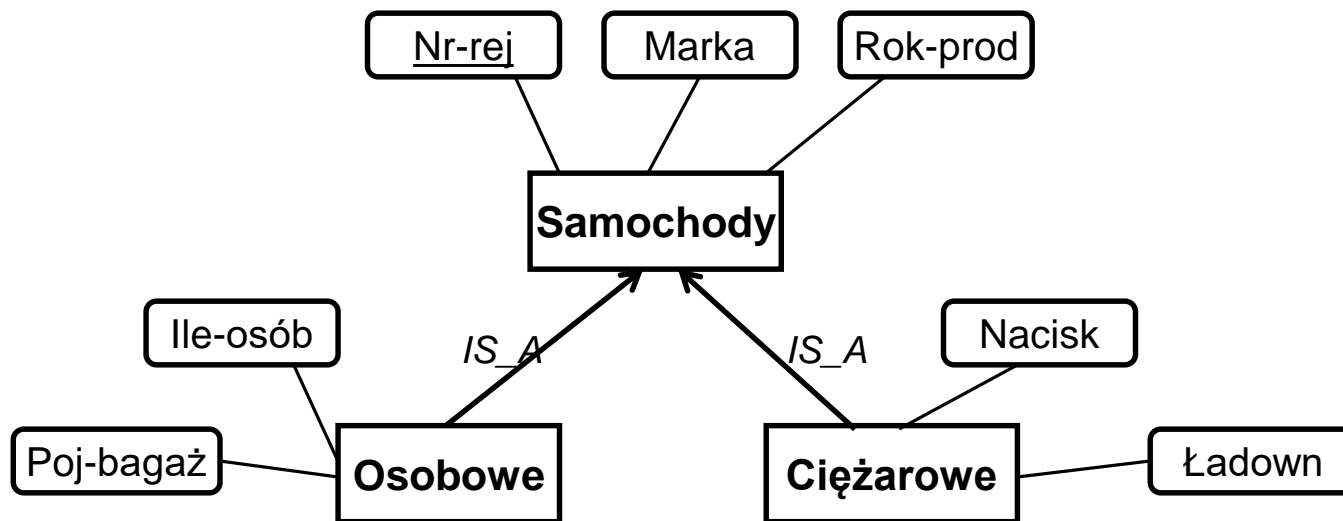


- $IS_A(E1, E2)$ oznacza, że każda encja ze zbioru $E1$ jest także encją ze zbioru $E2$.
- Innymi słowy, zbiór encji $E1$ jest podzbiorem zbioru encji $E2$.
- Związek IS_A jest związkiem wbudowanym w model związków encji (ERM), typu 0..1:1.



$IS_A(E_1, E_2)$





- Każda encja ze zbiorów Osobowe i Ciężarowe ma wszystkie atrybuty encji ze zbioru Samochody (także klucz Nr-rej), a ponadto ma atrybuty dodatkowe, specyficzne dla wszystkich encji z jej zbioru
- Mowa tu o dziedziczeniu atrybutów pomiędzy zbiorami encji



1. Zidentyfikuj encje i zbiory encji.
2. Zdefiniuj stosowne atrybuty i klucze encji.
3. Zidentyfikuj wszystkie związki typu IS_A. Wprowadź niezbędne podzbiory encji i zdefiniuj dodatkowe atrybuty.
4. Zidentyfikuj związki pomiędzy zbiorami encji. Dla każdego związku sprecyzuj jego liczność i stopień.
5. Zredukuj wszystkie związki stopni $n > 2$ do związków binarnych. Wprowadź zbiory encji reprezentujące te związki, określ ich klucze i ew. inne atrybuty.
6. Zweryfikuj powstały model: Czy adekwatnie (tj. z wystarczającą dokładnością) reprezentuje on świat rzeczywisty? Jeśli nie, cofnij się do któregoś z poprzednich punktów.



1. Określ klienta.
2. Zidentyfikuj cel bazy danych.
3. Określ użytkowników.
4. Zbierz scenariusze użycia systemu.
5. Zastanów się, jakiego rodzaju dane będą potrzebne, by zrealizować te scenariusze.
6. Narysuj diagram ERD.
7. Sprawdź, czy można zrealizować scenariusze, ew. popraw ERD.
8. J.w.... (x razy).
9. Opisz diagram ERD (słownik danych).
10. Zdefiniuj schemat bazy danych, np. relacyjnej.



- Klient – szkoła
- Cel bazy – przechowywanie i poszukiwanie informacji o uczniach, nauczanych przedmiotach i nauczycielach
- Przykładowe pytania:
 - jakie są przedmioty?
 - jakie są klasy?
 - jaki nauczyciel uczy danego przedmiotu?
 - jakie są oceny uczniów?
 - ilu jest uczniów w klasie?
 - jakie są średnie ocen ucznia?
 - z których przedmiotów są najlepsze wyniki?
 - jacy uczniowie są najlepsi w klasie? W szkole?
 - czy jest uczeń, który nie ma ocen z więcej niż 3 przedmiotów?
 - czy jest nauczyciel, który prowadzi dwa przedmioty w tej samej klasie?



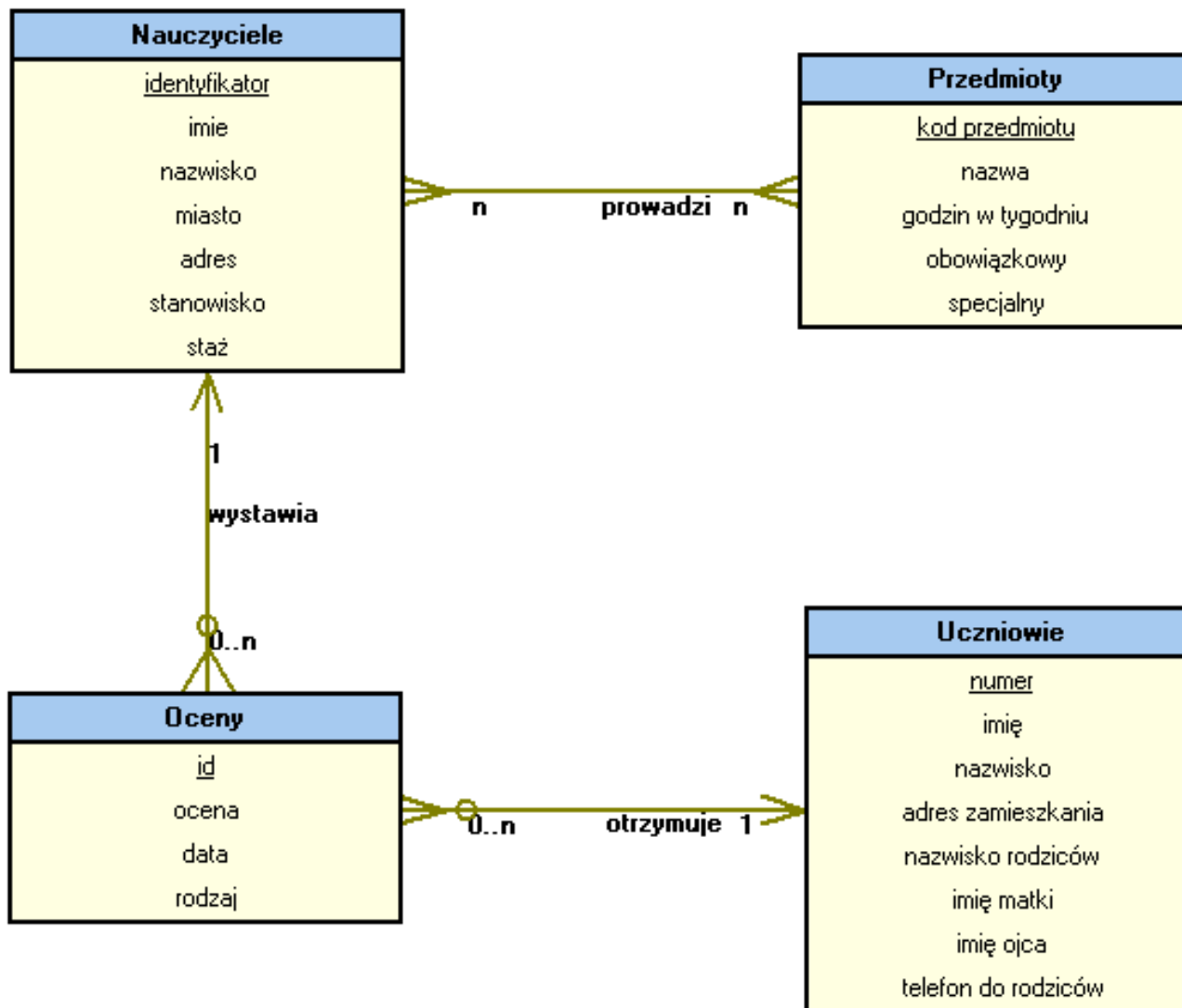
- Scenariusz użycia systemu – opis w języku naturalnym wykorzystania powstającego systemu w docelowym kontekście przez użytkownika w celu rozwiązania jego problemów:
 - opisuje pracę z systemem z punktu widzenia użytkownika
 - dla danego produktu może być wiele scenariuszy użycia
 - stanowi podstawę identyfikacji funkcji systemu i jego cech



- Przykładowe scenariusze (opis skrócony):
 - nauczyciel wyszukuje klasę IA i wpisuje oceny z matematyki wszystkim uczniom
 - dyrektor sprawdza, czy są uczniowie regularnie nieuczęszczający na zajęcia i wysyła im ostrzeżenia
 - nauczyciel sprawdza listę obecności na zajęciach z języka polskiego
 - uczeń sprawdza oceny z historii
 - rada pedagogiczna analizuje sytuację uczniów po wystawieniu ocen na półrocze
 - dyrektor sprawdza, czy we wszystkich klasach jest tyle samo uczniów, czy też jest jakaś klasa przeciążona
 - przed zakończeniem roku nauczyciel podsumowuje nieobecności uczniów na poszczególnych przedmiotach
 - dyrektor sprawdza, ile zajęć w tygodniu mają poszczególni nauczyciele i czy nie ma osoby, która ma za mało godzin



Diagram związków encji dla szkoły





- Relacja (bazodanowa) – dowolny podzbiór produktu kartezyjskiego skończonej liczby dziedzin prostych
- Dziedzina jest prosta, jeśli jej elementy są nierozkładalne (atomowe)
- Poszczególne pozycje krotek relacji nazywamy atrybutami relacji
- Wymóg atomowości elementów dziedzin atrybutów relacji formułujemy też następująco:
 - relacja jest zawsze w pierwszej postaci normalnej (1NF)



- Relację reprezentujemy w postaci tabeli, w której każdy wiersz odpowiada jednej krotce, a każda kolumna odpowiada jednemu atrybutowi
- Nazwa kolumny jest nazwą atrybutu. Nazwy atrybutów nie mogą się powtarzać
- Przykład tablicy reprezentującej pewną relację o 3 atrybutach:

NIP	Nazwisko	Status
123-123-34-56	Kowalski	10
345-321-55-66	Nowak	20
123-333-21-13	Kwiatkowski	30
345-111-12-12	Nowak	90



- W relacji nie ma dwóch identycznych krotek
 - ponieważ relacja jest matematycznym zbiorem
- Krotki nie są uporządkowane
 - ponieważ relacja jest matematycznym zbiorem
- Wartości atrybutów są atomowe (są pojedynczymi wartościami):
 - na każdej pozycji każdej krotki (w jednej komórce tabeli reprezentującej relację) istnieje tylko jedna, elementarna, niepodzielna wartość
 - innymi słowy, relacja bazodanowa jest zawsze w pierwszej postaci normalnej (1NF)
- Relacja matematyczna nie musi być 1NF – dziedziny mogą być złożone



1. **NicNieZnaczy** (Numer, Łańcuch, Data, Litera)

```
create table NicNieZnaczy
  (numer    integer,
   lancuch  varchar(128),
   data     date,
   litera   char(1))
```

2. **Klienci** (NIP, Nazwisko, Imię, Adres, Status),

NIP \in {zbiór legalnych numerów NIP},

Nazwisko, Imię, Adres \in {zbiór łańcuchów znaków}

Status \in {zbiór liczb naturalnych}

```
create table Klienci
  (nip      char(10),
   nazwisko varchar(64),
   imie     varchar(32),
   adres    varchar(128),
   status   smallint)
```

3. **Książki** (ISBN, Autor, Tytuł, Rok),

ISBN \in {zbiór numerów ISBN},

Autor, Tytuł \in {zbiór łańcuchów},

Rok \in {zbiór lat}

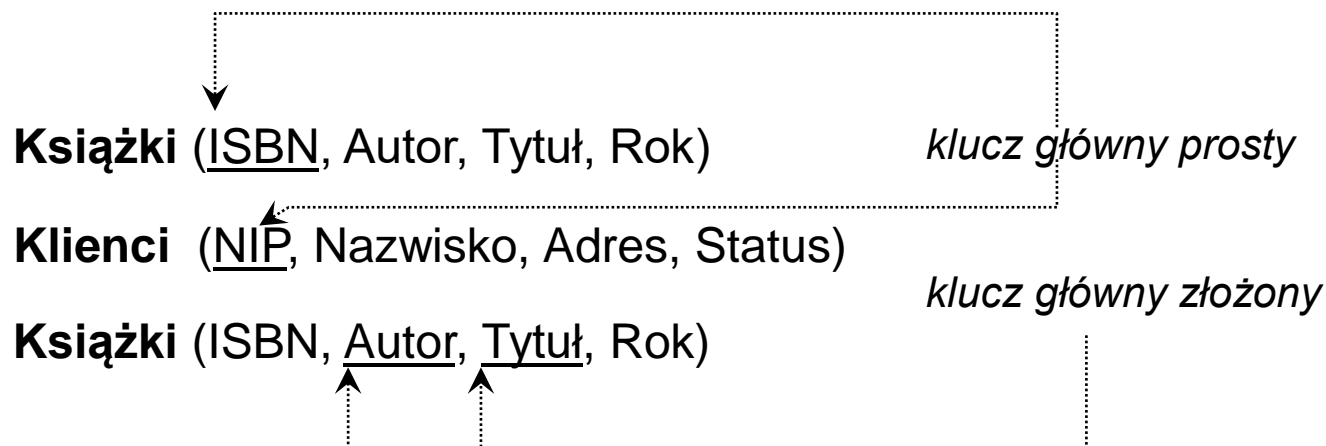
```
create table Ksiazki
  (isbn    char(24),
   autor   varchar(64),
   tytul   varchar(128),
   rok     integer)
```



- Niech R jest relacją o atrybutach A_1, A_2, \dots, A_n
- Rozważmy podzbiór atrybutów $K = (A_i, A_j, \dots, A_k)$
- Warunek jednoznaczności:
 - w żadnym momencie nie istnieją w relacji R dwie krotki, które mają te same wartości atrybutu A_i , te same wartości atrybutu A_j , ..., te same wartości atrybutu A_k
- Warunek minimalności:
 - ze zbioru atrybutów A_i, A_j, \dots, A_k nie można usunąć żadnego atrybutu bez naruszenia warunku jednoznaczności



- Klucz relacji (klucz kandydujący) – atrybut (lub zbiór atrybutów) relacji spełniający warunki jednoznaczności i minimalności
- Klucz prosty – ma tylko jeden atrybut
- Klucz złożony – składa się z wielu atrybutów
- Klucz główny (inaczej: klucz podstawowy) relacji jest arbitralnie wybranym kluczem kandydującym relacji
- Pozostałe klucze (jeśli istnieją) są kluczami alternatywnymi



```
create table Ksiazki
(isbn char(24),
autor varchar(64),
tytul varchar(128),
rok integer,
PRIMARY KEY (isbn))
```

```
create table Klienci
(nip char(10),
nazwisko varchar(64),
imie varchar(32),
adres varchar(128),
status smallint,
PRIMARY KEY (nip))
```

```
create table Ksiazki
(isbn char(24),
autor varchar(64),
tytul varchar(128),
rok integer,
PRIMARY KEY (autor,tytul))
```



Książki (ISBN, Autor, Tytuł, Rok, Id-wyd REF Wydawnictwa)

Wydawnictwa (Ident, Nazwa, Miasto)

klucz obcy
odpowiadający mu klucz główny

```
create table Wydawnictwa
(ident      char(24),
 nazwa     varchar(64),
 miasto    varchar(32),
 PRIMARY KEY (ident))
```

```
create table Ksiazki
(isbn      char(24),
 autor     varchar(64),
 tytul     varchar(128),
 rok       integer,
 id_wyd    char(24),
 PRIMARY KEY (isbn),
 FOREIGN KEY (id_wyd)
 REFERENCES Wydawnictwa)
```



Pracownicy (Nr-ewid, Nazwisko, Stanowisko, Szef REF Pracownicy)


*TEN klucz obcy odpowiada
TEMU kluczowi głównemu*

```
create table Pracownicy
(nr_ewid    char(8),
 nazwisko  varchar(64),
 stanowisko char(24),
 szef      char(8),
 PRIMARY KEY (nr_ewid),
 FOREIGN KEY (szef)
 REFERENCES Pracownicy)
```



Zamówienia (Nr-zam, Data, Klient)

Poz-Zam (Nr-zam REF Zamówienia, Lp, Towar, Ilość)

 *ten klucz obcy wchodzi w skład
klucza głównego*

```
create table Zamowienia
(nr_zam    char(16),
 data     date,
 klient   char(10),
PRIMARY KEY (nr_zam))
```

```
create table Poz_Zam
(nr_zam    char(16),
 lp        smallint,
 towar     varchar(128),
 ilosc     real,
PRIMARY KEY (nr_zam,lp),
FOREIGN KEY (nr_zam)
REFERENCES Zamowienia)
```



Złożony klucz obcy:

*ten klucz obcy składa się
z dwóch atrybutów*

Zamówienia (**Nr-zam**, Data, Klient, ...)

Nr-zam KEY

Poz-Zam (Nr-zam, Lp, Towar, Ilość, ...)

(**Nr-zam**, **Lp**) KEY

Nr-zam REF Zamówienia

Opis-Poz-Zam (**Nr**, **Nr-zam**, **Lp**, Tekst)

(**Nr**, **Nr-zam**, **Lp**) KEY

(**Nr-zam**, **Lp**) REF Poz-zam

- Klucz obcy nie musi mieć takiej samej nazwy jak klucz główny relacji przezeń wskazywanej, jednak w miarę możliwości należy stosować tę konwencję, gdyż polepsza czytelność schematu bazy danych



- Zestaw operatorów, które mogą być stosowane w relacjach

Operatory algebry relacji	
Operatory zbiorowe	Specjalne operatory relacyjne
* suma (unia)	* rzutowanie (projekcja)
* różnica	* wybór (selekcja)
Iloczyn (przecięcie)	złączenie
* produkt kartezjański	* zmiana nazwy

- Opis:
 - operatory oznaczone * są prymitywne, tzn. żaden z nich nie może być wyrażony jako kombinacja pozostałych
 - wynikiem każdego operatora algebry relacji jest relacja (algebra relacji jest domknięta)
 - umożliwia to budowę dowolnie zagnieżdżonych wyrażeń relacyjnych, można np. wykonać projekcję sumy, selekcję złączenia, złączenie wyników kilku selekcji, itd.

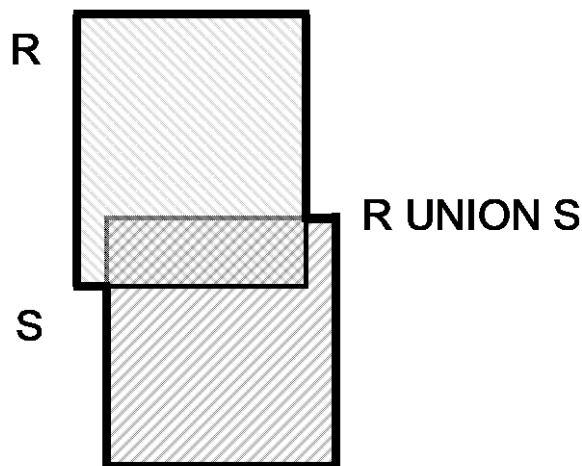


Definicja

Dwie relacje są **zgodne** (kompatybilne), jeśli:

- mają ten sam stopień n ,
- i -te atrybuty każdej z nich ($i = 1, \dots, n$) mają tę samą dziedzinę.

R UNION S (gdzie R i S są **zgodne**):
zbiór wszystkich krotek, które należą do R lub do S.



Przykład:

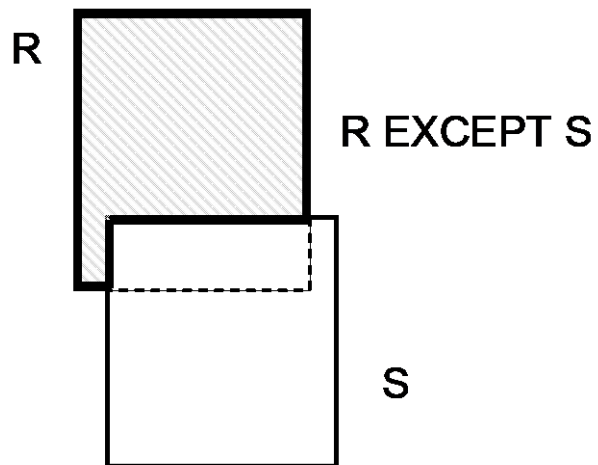
R – zbiór nauczycieli, którzy uczą studenta S5

S – zbiór nauczycieli, którzy uczą studenta S6

R UNION S – zbiór nauczycieli, którzy uczą studenta S5 *lub* studenta S6



R EXCEPT S (gdzie R i S są **zgodne**):
zbiór wszystkich krotek, które należą do R i nie należą do S.



Przykład:

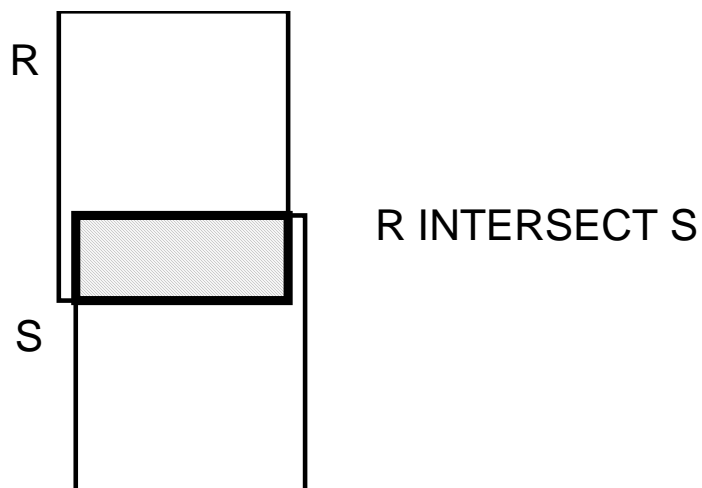
R – zbiór nauczycieli, którzy uczą studenta S5

S – zbiór nauczycieli, którzy uczą studenta S6

R EXCEPT S – zbiór nauczycieli, którzy uczą studenta S5, ale nie uczą studenta S6



$R \text{ INTERSECT } S$ (gdzie R i S są **zgodne**):
zbiór wszystkich krotek, które należą do R i jednocześnie należą do S .



Przykład:

R – zbiór nauczycieli, którzy uczą studenta $S5$

S – zbiór nauczycieli, którzy uczą studenta $S6$

$R \text{ INTERSECT } S$ – zbiór nauczycieli, którzy uczą studenta $S5$ i uczą studenta $S6$

Uwaga:

INTERSECT nie jest operatorem *prymitywnym*, gdyż można zapisać:

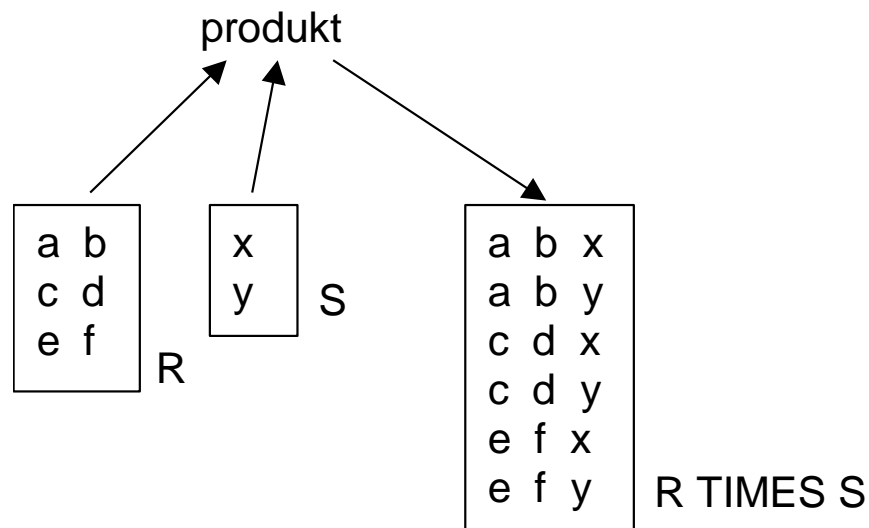
$$\begin{aligned} R \text{ INTERSECT } S &\equiv R \text{ MINUS } (R \text{ MINUS } S) \\ \text{lub } R \text{ INTERSECT } S &\equiv S \text{ MINUS } (S \text{ MINUS } R) \end{aligned}$$



R TIMES S:

zbiór wszystkich krotek t takich, że t jest konkatencją krotki $r \in R$ i krotki $s \in S$.

Stopień relacji będącej wynikiem operacji R TIMES S jest sumą stopni relacji R i relacji S.



Przykład:

R – zbiór wszystkich identyfikatorów nauczycieli

S – zbiór wszystkich identyfikatorów studentów

R TIMES S – zbiór wszystkich par (ident-nauczyciela, ident-studenta):

$\{(T1,S1), (T1,S2), \dots, (T1,S6), (T2,S1), \dots, (T2,S6), \dots, (T5,S1), \dots, (T5,S6)\}$



R JOIN S WHERE *warunek*:

zbiór wszystkich krotek t takich, że t jest konkatencją krotki r z relacji R i krotki s z relacji S
i dla każdej krotki t warunek ma wartość TRUE.

Uwaga:

JOIN nie jest operatorem *prymitywnym*, gdyż można zapisać:

R JOIN S WHERE *warunek* \equiv **SELECT FROM (R TIMES S) WHERE**
warunek

Przykład 1:

```
TEACHERS JOIN STUDENTS
  WHERE TEACHERS.CITY = STUDENTS.CITY
        AND SYEAR > 1983
```



Złączenie można stosować do kilku tablic:

```
R JOIN S JOIN T JOIN ... WHERE warunek ≡  
SELECT FROM (R TIMES S TIMES T TIMES ...) WHERE warunek
```

Złączenia są zwykle składane z projekcją, w przeciwnym razie wynik może mieć bardzo dużą liczbę kolumn.

Przykład 2:

PROJECT

(TEACHERS JOIN STUDENTS

WHERE TEACHERS.CITY = STUDENTS.CITY AND SYEAR > 1983)

OVER TNO, TNAME, TITLE, TEACHERS.CITY, SUPNO, SNO, SNAME, SYEAR



```
SELECT FROM (
PROJECT
(Klienci JOIN Zamówienia
WHERE Klienci.NIP = Zamówienia.Klient)
OVER NIP, Nazwisko, Nr-zam, Data)
WHERE Data >= '01.01.1998';
```

lub :

```
SELECT NIP, Nazwisko, Nr-zam, Data
FROM Klienci, Zamówienia
WHERE Klienci.NIP = Zamówienia.Klient
AND Data >= '01.01.1998';
```

```
SELECT NIP, Nazwisko, Nr-zam, Data
FROM Klienci JOIN Zamówienia
ON Klienci.NIP = Zamówienia.Klient
WHERE Data >= '01.01.1998';
```

NIP	Nazwisko	Nr-zam	Data
123-123-34-56	Kowalski	001	12.01.1998
123-123-34-56	Kowalski	002	14.03.1998
123-333-21-13	Kowalski	004	01.01.1998
345-321-55-66	Nowak	005	13.04.1998
345-321-55-66	Nowak	012	24.11.1998
123-333-21-13	Kwiatkowski	034	12.01.1998
123-123-34-56	Kowalski	045	11.11.1998

Dziękuję za uwagę