

BAZY DANYCH

3

GRZEGORZ SURÓWKA

2018

PLAN WYKŁADU

1. WPROWADZENIE
2. DBMS, MODELE DANYCH, MODEL RELACYJNY
3. **DIAGRAMY E/R**
4. E/R → MODEL RELACYJNY
5. ALGEBRA RELACJI
6. ZŁĄCZENIA
7. NORMALIZACJA BAZY DANYCH
8. TRANSAKCJE
9. OPTYMALIZACJA ZAPYTAŃ
10. WYDAJNOŚĆ
11. PODZAPYTANIA SQL
12. GRUPOWANIE
13. WIDOKI, KURSORY, WYZWALACZE, SEKWENCJE
14. SQL

PLAN

- PROJEKTOWANIE BAZ DANYCH
- DIAGRAMY E/R
- NOTACJE

PROJEKTOWANIE BAZ DANYCH

zasady projektowania baz danych

dokładność

- projekt powinien odpowiadać specyfikacji
- tabele powinny odzwierciedlać świat rzeczywisty

unikanie redundancji

- oszczędzanie zasobów (miejsce, czas)
- ryzyko, że nie wszystkie wystąpienia danej informacji będą uaktualnione

prostota

- tylko tyle elementów, ile potrzeba

dobór właściwych elementów

- nie wszystko modelujemy jako atrybuty

zebranie i analiza informacji

- cel powstania systemu
- wywiady z zainteresowanymi
- dokumentacja wymogów użytkowników
- analiza istniejącego systemu (jeśli istnieje)

identyfikacja jednostek i pojęć

- identyfikacja encji: tworzenie listy atrybutów, które ją charakteryzują

projekt/model danych

- powszechnie przyjęty model:
model związków-encji (E/R)
- struktura danych jest reprezentowana
graficznie przez diagram związków-encji (ERD)

schemat relacyjny

- model abstrakcyjny E/R jest przekształcany
do schematu relacyjnego
- DBMS korzysta najczęściej z modelu relacyjnego

utworzenie bazy danych

- użytkownik posiadający uprawnienia administratora tworzy bazę danych
- baza otrzymuje strukturę wg. opracowanego schematu relacyjnego

model systemu (konceptcja)

→ model encji

→ model relacyjny z encji

→ tabele w bazie danych

rodzaje modeli

konceptualny

- umożliwia reprezentowanie obiektów w uniwersalnym modelu niezależnym od modelu implementacyjnego
- najpopularniejsze: model związków-encji, model UML

implementacyjny (=model danych)

- jest wykorzystywany na etapie implementacji systemu
- zamienia model konceptualny w konkretne struktury danych
- najpopularniejsze: relacyjny, obiektowy, obiektowo-relacyjny i semistukturalny

metody projektowania schematu relacyjnego

Top-Down

- utworzyć model E/R
- zastosować reguły transformacji modelu E/R na schemat relacyjny

Down-Top

- zebrać jak najwięcej danych, które będą tworzyć zawartość bazy danych
- zidentyfikować tematy oraz ich właściwości: zdefiniować tabele relacyjne
- przeprowadzić proces normalizacji do 3 lub 4 postaci normalnej

mieszana

- utworzyć model E/R
- zastosować reguły transformacji modelu E/R na schemat relacyjny
- przeprowadzić proces normalizacji do 3 lub 4 postaci normalnej

MODEL E/R

- nazewnictwo: model związków encji, model jednostka-związek, model obiektowo-związkowy, E/R-Entity-Relation
- metoda graficzna modelowania schematu logicznego bazy danych
- twórca: Peter Chen (The Entity-Relationship Data Model: Toward a unified view of data, ACM Transactions of Database Systems, vol. 1, 1976)
- fundamentalny model wykorzystywany przy projektowaniu relacyjnych baz danych
- komponenty: encje, atrybuty, relacje=związki między encjami, hierarchia generalizacji

zalety

- przedstawia zakres koniecznej informacji
- przedstawia mapę/projekt bazy danych
- jest precyzyjny
- wydajny model integracji aplikacji

ERD – DIAGRAMY E/R

- notacja pozwalająca opisać schemat (strukturę) bazy danych

notacja

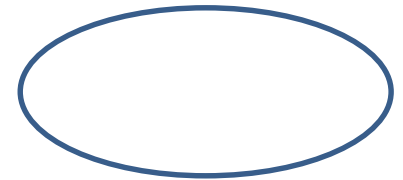
encja

prostokąt z nazwą w środku

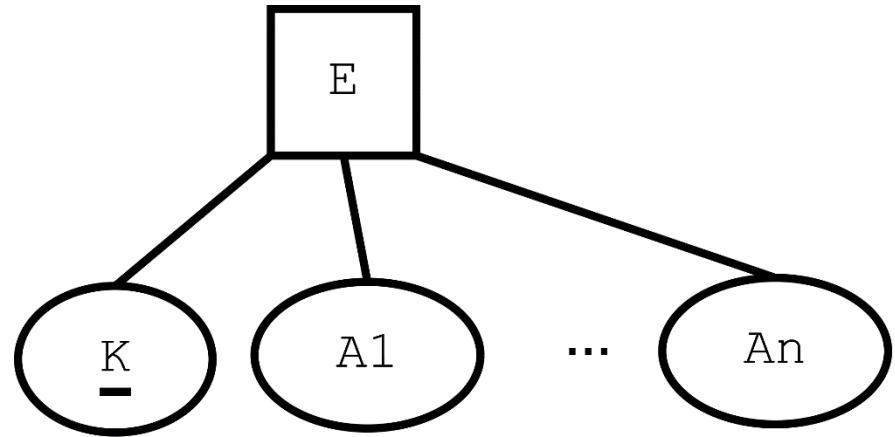


atrybut

- elipsa z nazwą atrybutu połączona z prostokątem linią ciągłą (atrybut kluczowy jest podkreślony)
- atrybut pochodny: elipsa przerywana



encja E reprezentowana jest przez relację=tabelę
 $R_E(K, A_1, A_2, \dots, A_n)$



decyzje projektowe

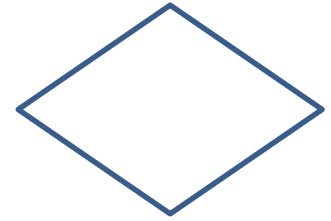
- co jest niezależną encją, a co wartością atrybutu?
- przykłady:

$R_{\text{Towar}}(\text{Id}, \text{Nazwa}, \text{Cena}, \text{Dostawca})$

„Nowy Sącz, ul. Batorego 7/12” (adres jako atrybut)

Adres(Miejscowość, Ulica, NrDomu, NrLokalu)
(adres jako encja)

związek



- jest reprezentowany przez romb
- nazewnictwo: czasowniki lub rzeczowniki odczasownikowe
- encje uczestniczą w związku i mogą być tego samego lub różnych typów
- związki nie mogą uczestniczyć w związkach
- nie posiada niezależnego identyfikatora - jest natomiast jednoznacznie identyfikowany przez zbiór identyfikatorów encji uczestniczących w związku
- posiada atrybuty
- przykłady: np. Pracownik **pracuje_w** Dziale,
Student **uczęszcza_na** Kurs (podkreślono encje)

- cechy związku: stopień, krotność, uczestnictwo

stopień

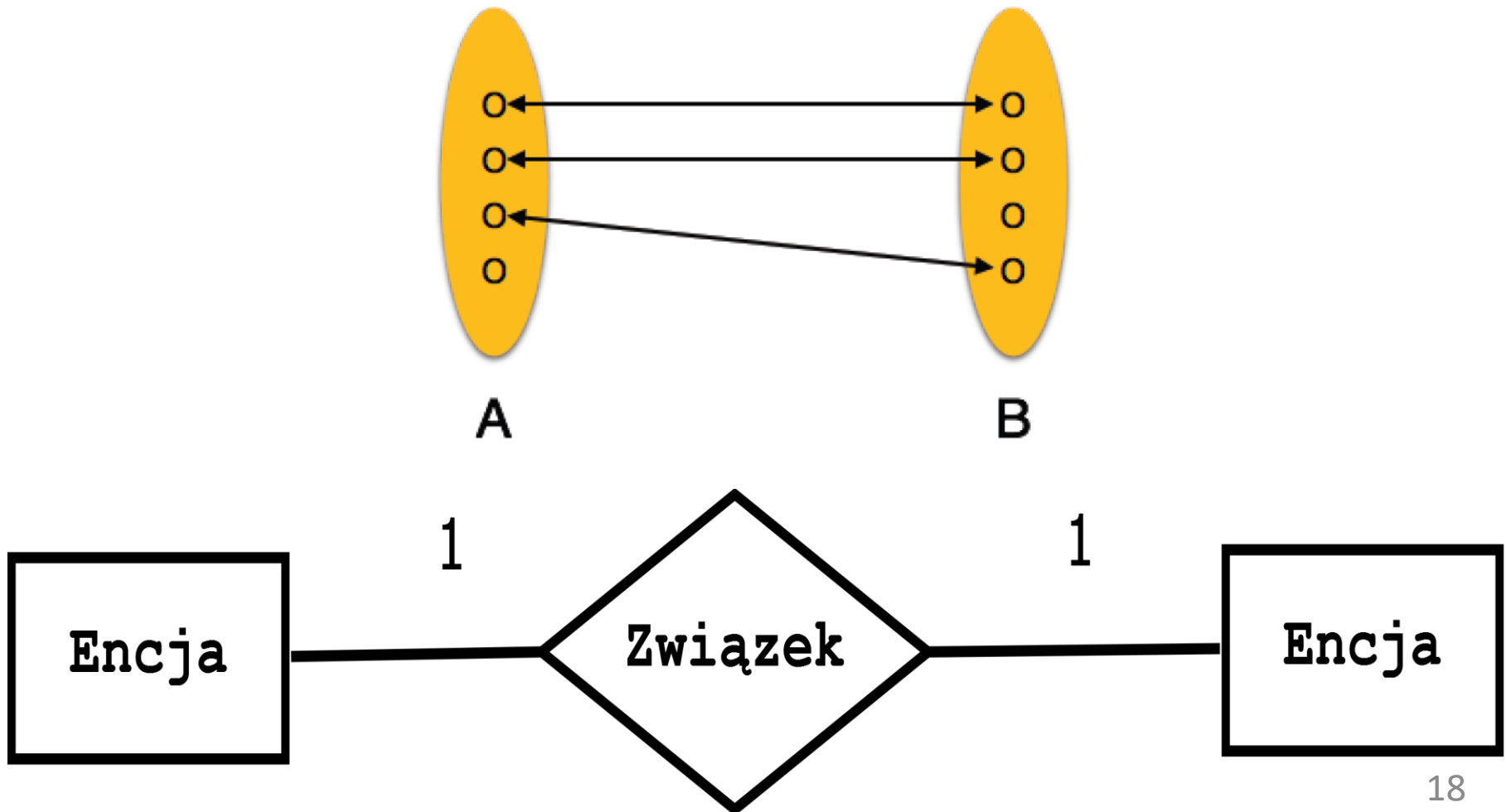
- to liczba encji uczestnicząca w relacji
- związki unarne (binarne rekursywne),
binarne = 2, ternarne = 3, n-arne = n

krotność (liczebność/moc/kardynalność/typ asocjacji)

- liczba encji w jednej tabeli (zbiorze encji), jaka może być w relacji z liczbą encji w innej tabeli

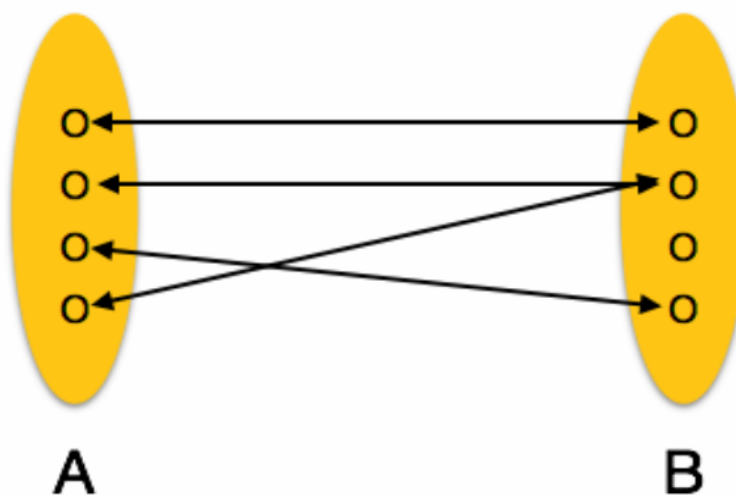
jeden-do-jeden

- jedna encja z tabeli A ma związek z co najwyżej jedną encją z tabeli B i odwrotnie



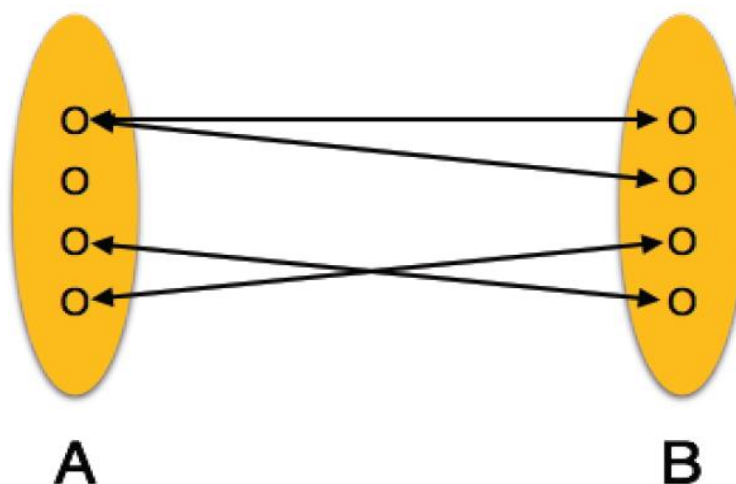
wiele-do-jeden

- więcej niż jedna encja z tabeli A ma związek z co najwyżej jedną encją z tabeli B, encja z tabeli B ma związek z więcej niż jedną encją z tabeli A



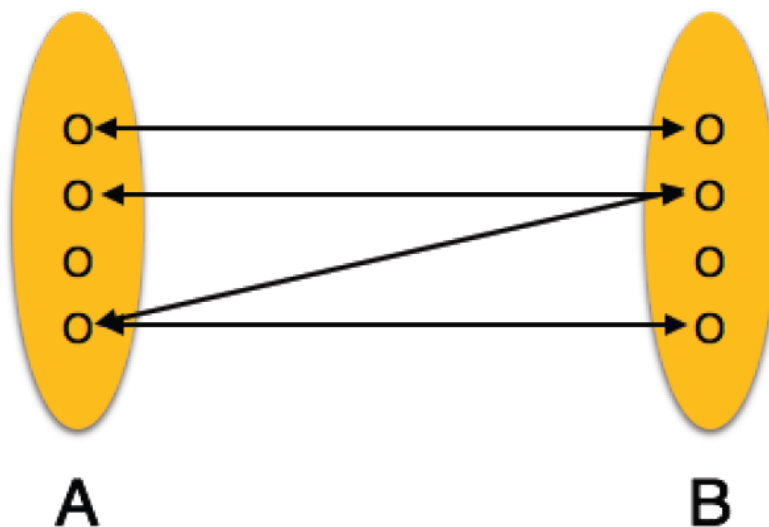
jeden-do-wiele

- jedna encja z tabeli A ma związek z więcej niż jedną encją z tabeli B, ale encja z tabeli B ma związek z co najwyżej jedną encją z tabeli A



wiele-do-wiele

- jedna encja z tabeli A ma związek z więcej niż jedną encją z tabeli B i na odwrót



opcjonalność (uczestnictwo)

- określa, czy wszystkie encje biorą udział w związku
- **uczestnictwo opcjonalne (częściowe):**
nie każda instancja encji jest uczestnikiem relacji
reprezentacja: pojedyncza linia
przykład: klient może nie mieć przypisanego konta
- **uczestnictwo wymagane (obowiązkowe, całkowite):**
każda instancja encji jest uczestnikiem relacji
reprezentacja: podwójna linia
przykład: konto musi być przypisane do klienta

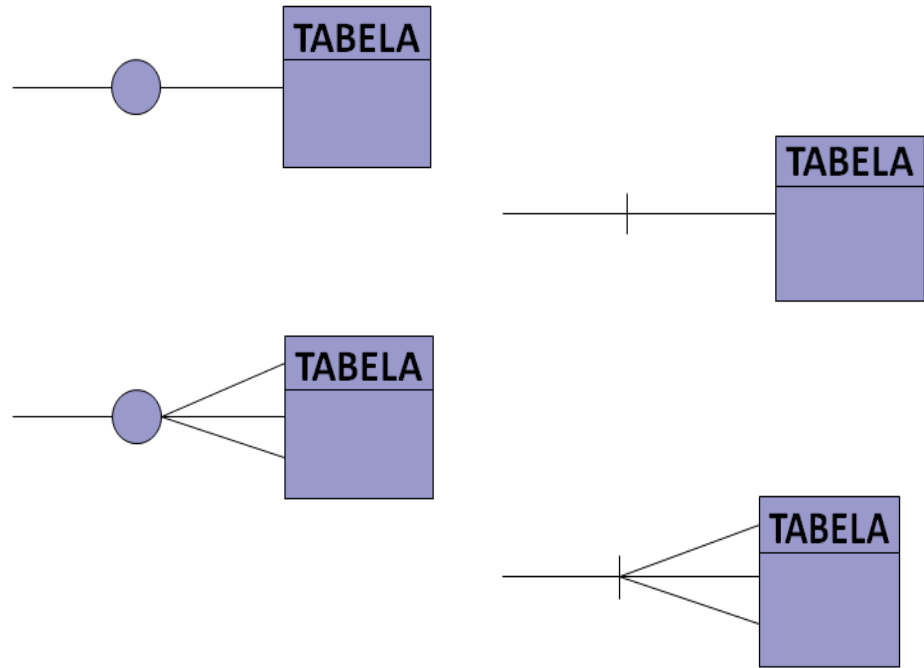
INNE NOTACJE

notacja Barker'a (Oracle)

- notacja wzorowana na notacji kruczej stopki
- klucz: poprzedzony znakiem #/◆
- atrybut obowiązkowy: poprzedzony znakiem */●
- atrybut mogący przyjmować wartości puste: poprzedzony kółkiem ○

Pracownik
PESEL
* adres
* pensja
○ telefon

- zero lub jeden
- dokładnie jeden
- zero lub więcej
- jeden lub więcej

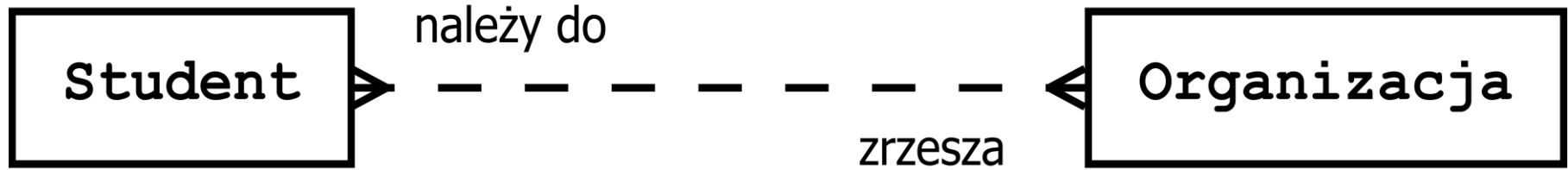


interpretacja symboli

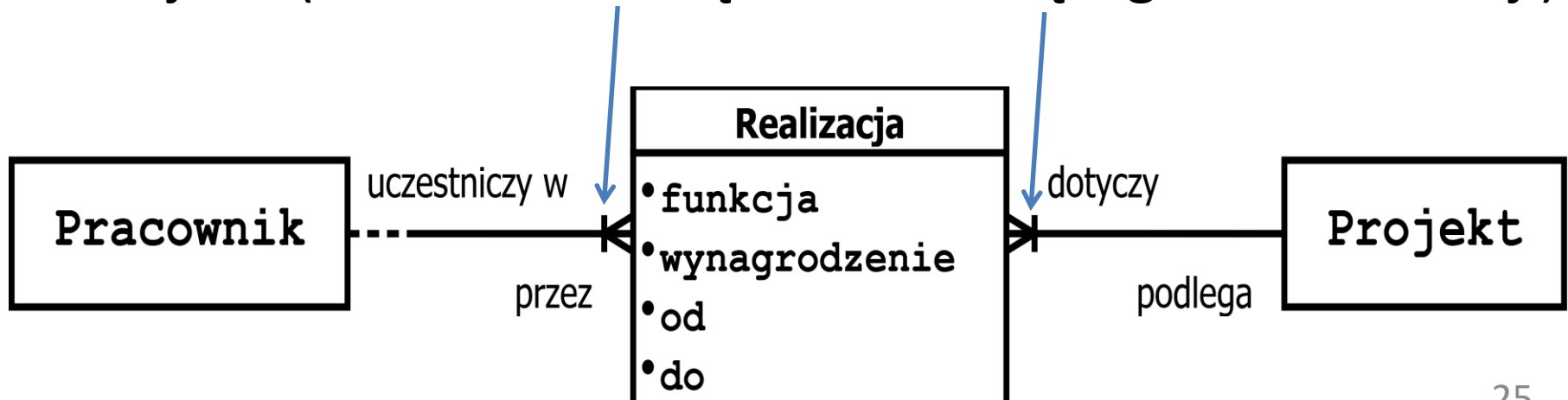
jedna encja z jednej tabeli może być powiązana z taką liczbą encji należących do drugiej tabeli, na jaką wskazuje symbol leżący bezpośrednio przy drugim zbiorze encji

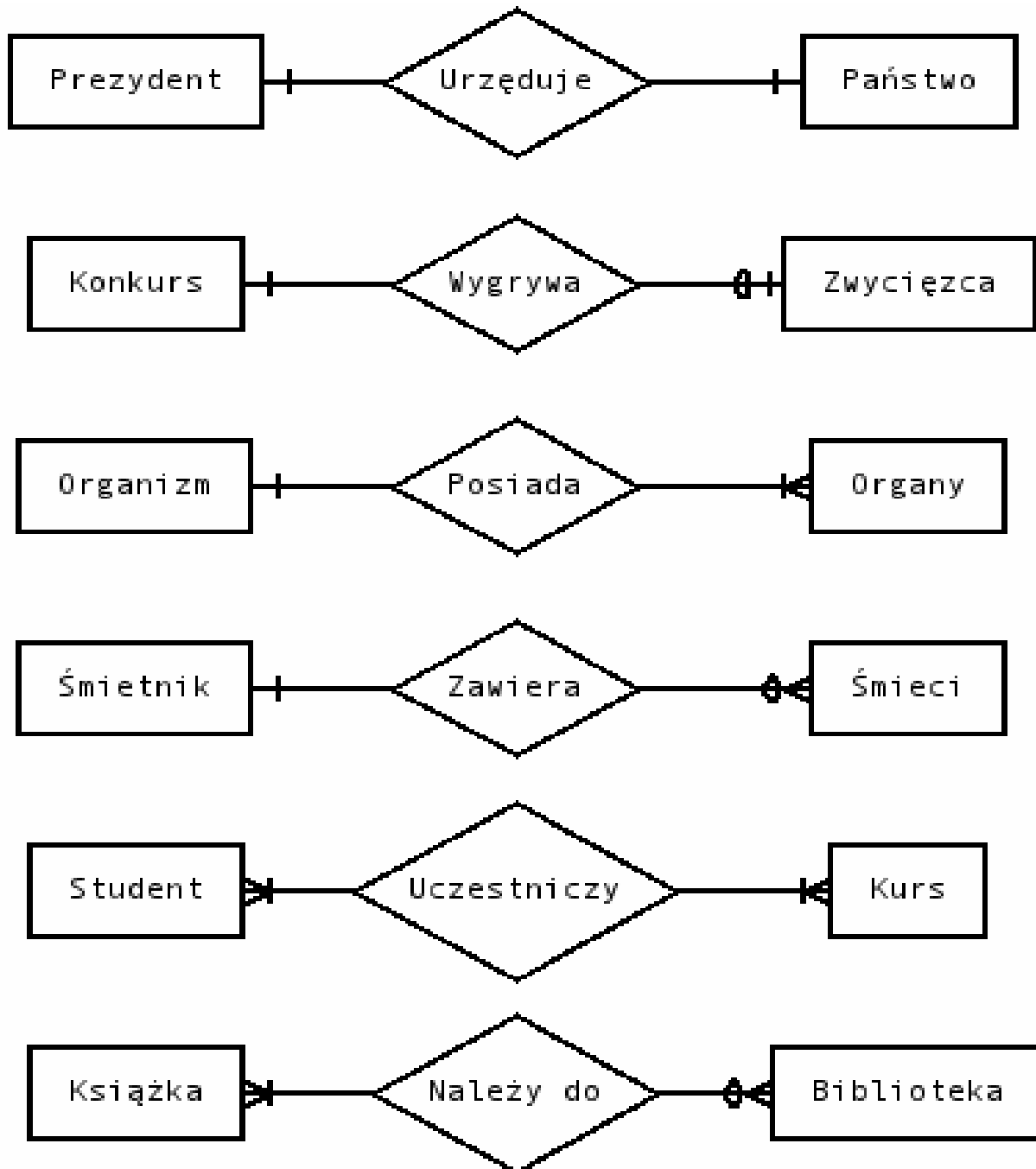
- uczestnictwo obowiązkowe: linia ciągła
- uczestnictwo opcjonalne: linia przerywana

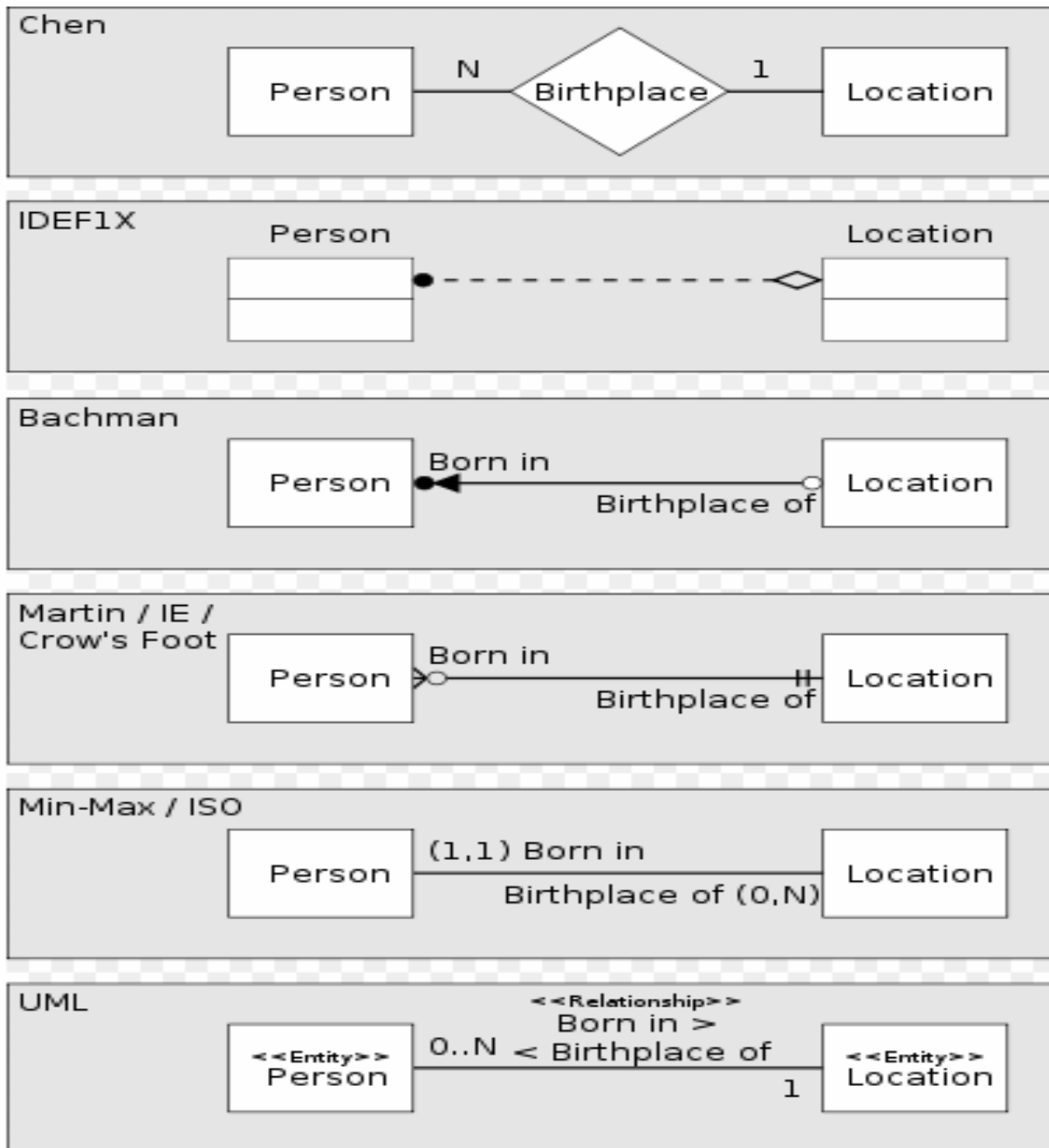
- związki



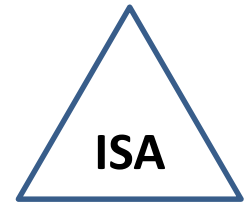
- notacja Barkera nie umożliwia wiązania atrybutów ze związkami → konstrukcją modelującą takie przypadki jest dodatkowa encja pośrednia pomiędzy oryginalnym encjami (oznaczenie związku wchodzącego w skład encji)





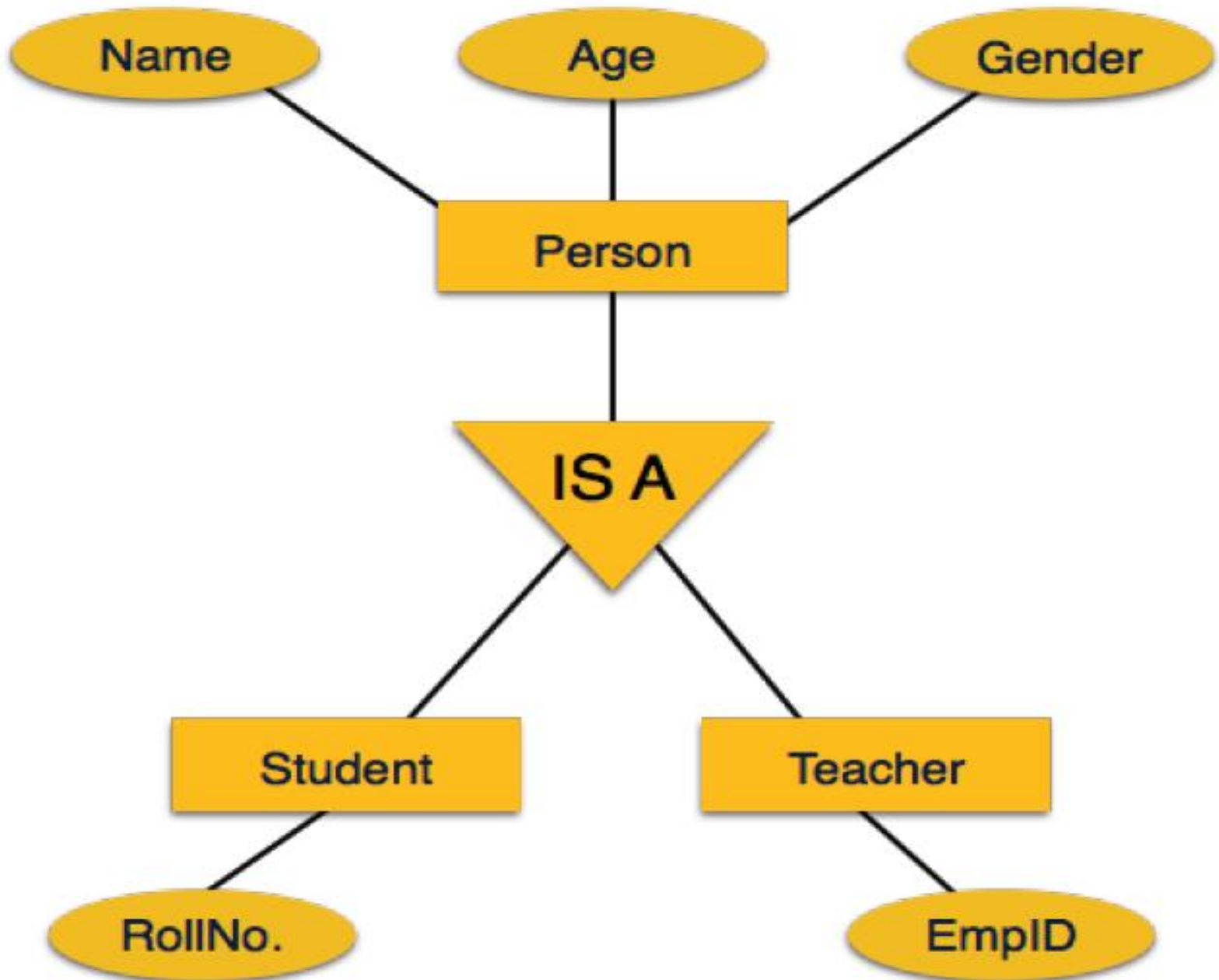


hierarchia encji (generalizacja) w modelu E/R



- model ER ma potencjał wyrażania encji w sposób hierarchiczny
- notacja: związek "isa" np. Student **is a** Person
- im wyżej w hierarchii, tym większe uogólnienie (generalizacja) encji, im niżej (specjalizacja), tym większy podział na podgrupy, więcej szczegółów encji
- podobieństwo do obiektowości:
dziedziczenie przez encje "niższe" cech encji "wyższych", szczegóły encji są ukryte przed użytkownikiem (abstrakcja)
- generalizacja: np. Student → Osoba
- specjalizacja: np. Osoba → Wykładowca

- podencje dziedziczą wszystkie atrybuty swojej nadencji
- podencje mogą mieć dodatkowe atrybuty i/lub związki, które nie przysługują wszystkim encjom zbioru wyjściowego (nadklasa)
- identyfikator nadencji jest wspólny dla wszystkich jej podencji (podencje nie mają swoich identyfikatorów)
- każde wystąpienie nadencji jest zawsze wystąpieniem jednej podencji
- każde wystąpienie podencji jest wystąpieniem nadencji



encja słaba (weak entity)

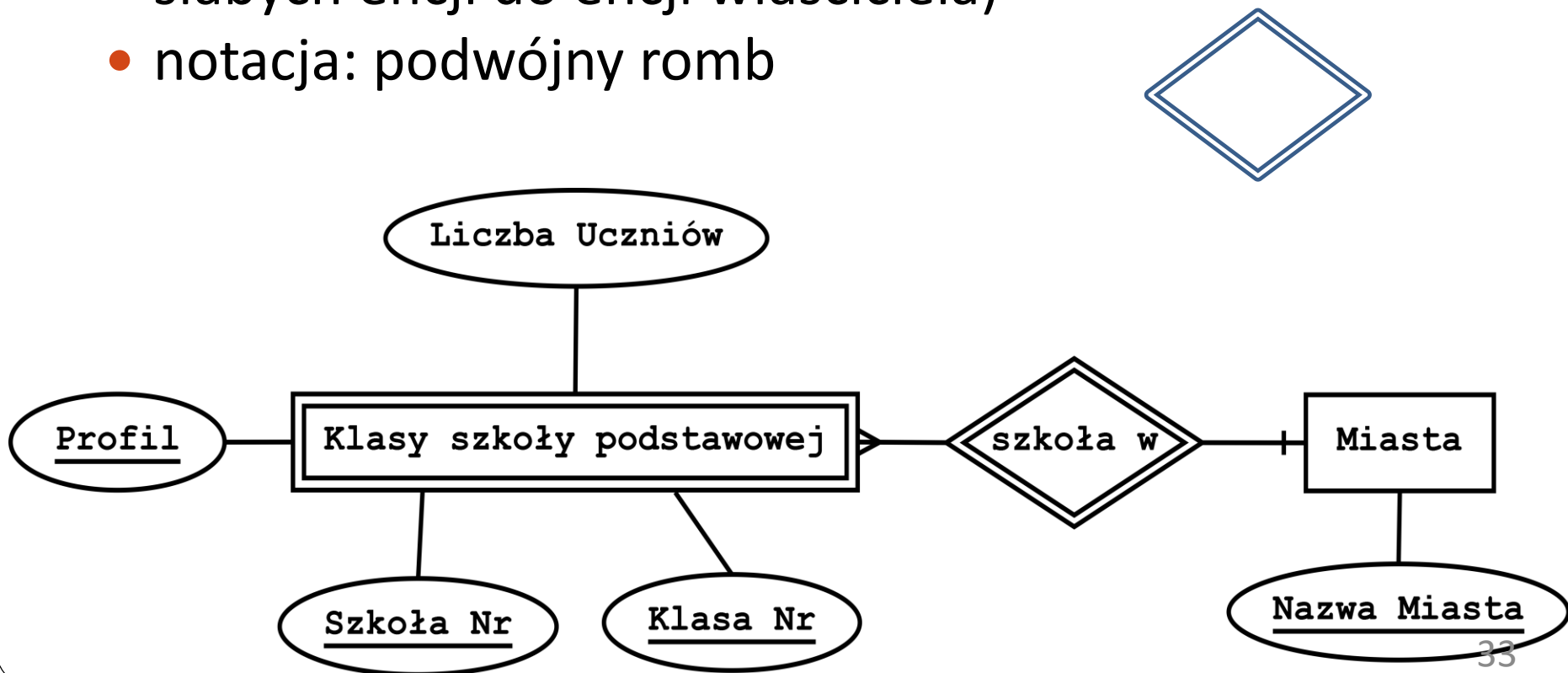
- encja, której istnienie zależy od istnienia innych encji (występuje jako elementy struktury wyższego rzędu)
- przykład: encja Dziecko w bazie danych korporacji ma sens tylko gdy istnieje encja Pracownik, będąca rodzicem dziecka (encja Pracownik jest właścicielem encji Dziecko)
- przykład: encja Realizacja o właścicielach: Pracownik, Projekt
- notacja: prostokąt z podwójną ramką, klucz częściowy = podkreślenie linią przerywaną



- nie ma własnego atrybutu kluczowego
- identyfikacja odbywa się przez kombinację wyróżnionego atrybutu (tzw. klucza częściowego) z kluczem głównym właściciela
np. $\text{klucz(Dziecko)} = (\text{IdPracownika}, \text{ImieDziecka})$ lub utworzony związek wspierający (albo dodatkową encję) np. $\text{Realizacja}(\text{IdPracownik}, \text{IdProjekt})$ (jeśli jest to związek wspierający to notacja: krótka linia prostopadła do związku umieszczona obok końca związku)
- krotność po stronie przeciwnej słabej encji wynosi dokładnie jeden - w przeciwnym razie mogłaby wystąpić sytuacja, w której słaba encja nie posiadałaby jednoznacznego identyfikatora

związek wspierający (identyfikujący)

- wskazuje zbiory encji, których klucze wykorzystywane są przez zbiory słabych encji
- związek między encją słabą, a encjami jej właścicieli (binarny i typu wiele-do-jeden ze zbioru słabych encji do encji właściciela)
- notacja: podwójny romb



związek unarny (binarny rekursywny)

- określa powiązanie pomiędzy wystąpieniem encji a innym wystąpieniem tej samej encji
- związek modelujący zależności hierarchiczne musi być opcjonalnym z obu stron - w przeciwnym przypadku, powstałaby hierarchia nieskończona
- przykłady:

encja: Osoba, związek: mąż/żona (Małżeństwo)

encja: Pracownik, związek: podlega/kieruje
(Zależność Służbowa)

encja: Podzespół, związek:
składa się/jest częścią
(Część Główna-Część Składowa)

