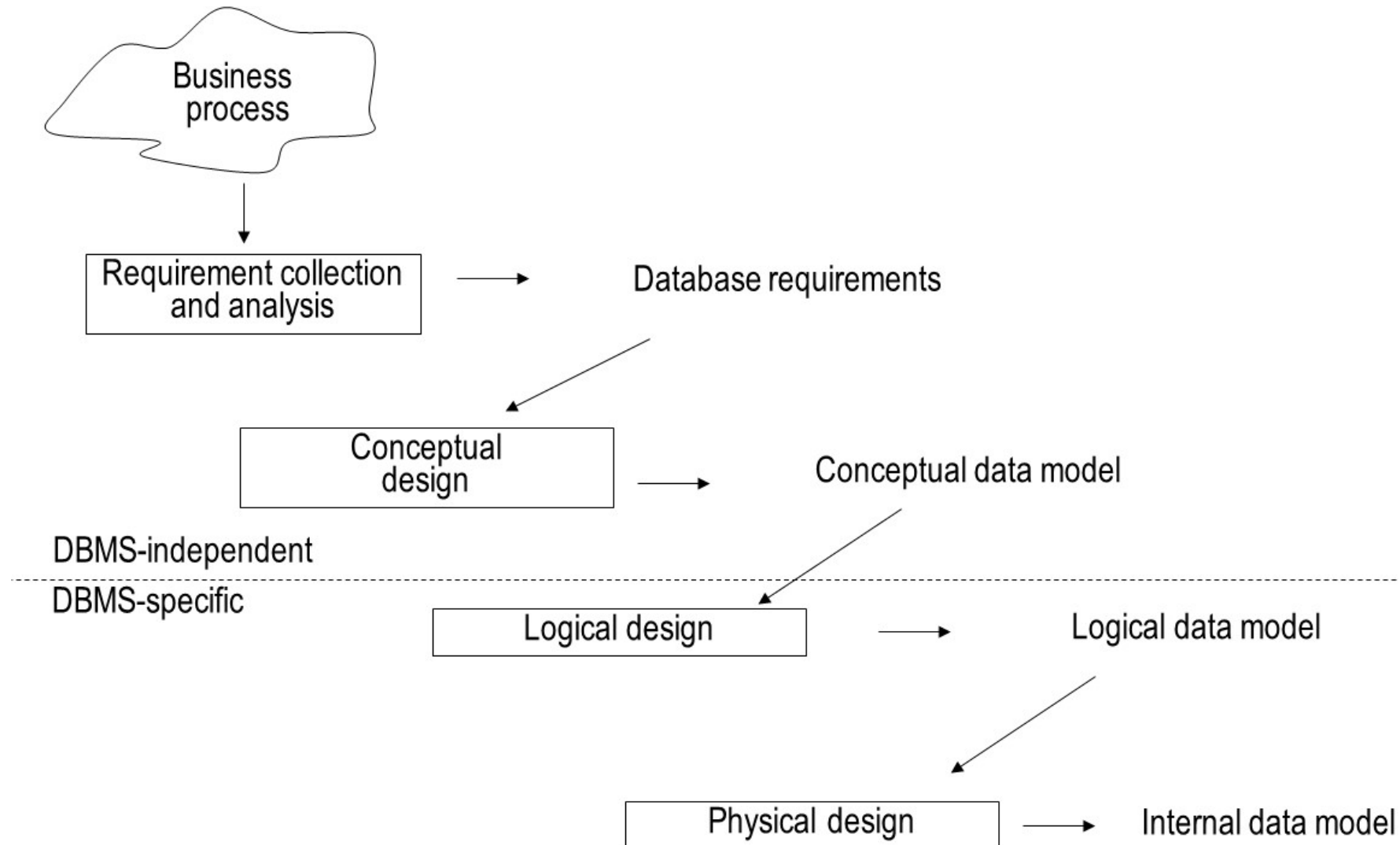


Zagadnienia

- Fazy projektowania bazy danych
- Model związków encji (Entity Relationship - ER)
- Rozszerzony model związków (Enhanced Entity Relationship - EER)
- UML

Fazy projektowania bazy danych



Model związków encji (Entity Relationship - ER)

- Typy encji
- Typy atrybutów
- Typy związków
- Słabe encje
- Związki ternarne (trójskładnikowe)
- Przykłady modelu ER
- Ograniczenia modelu ER

Peter Chen “The Entity–Relationship Model: Toward A Unified View of Data”, 1975, *ACM Transactions on Database Systems*.

- uważany za jeden z najbardziej wpływowych artykułów dziedzinie oprogramowania komputerowego

Typy encji

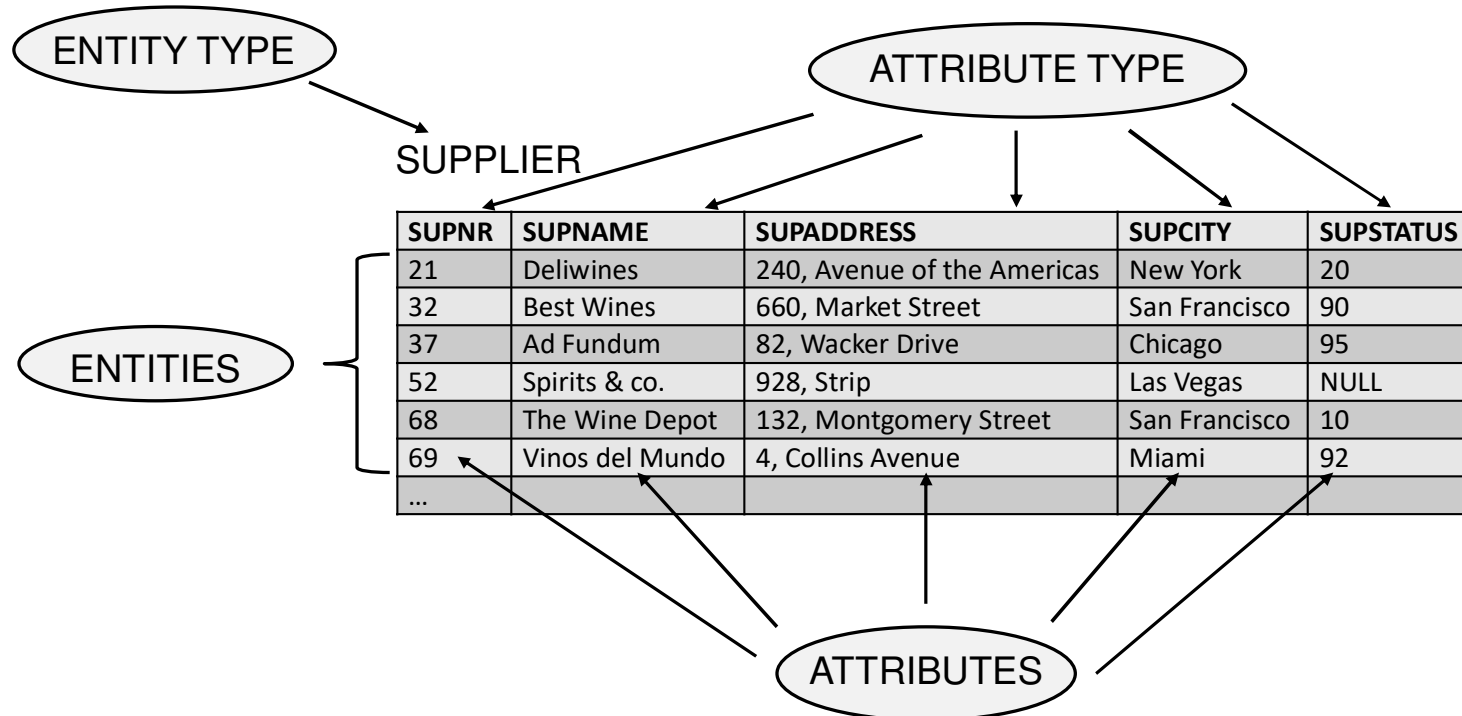
- Typ encji reprezentuje pojęcie biznesowe o jednoznacznym znaczeniu dla określonej grupy użytkowników
 - np. dostawca, student, produkt, pracownik...
- Encja to jedno określone wystąpienie lub instancja typu encji
 - np. Deliwines, Best Wines i Ad Fundum to encje typu encji dostawca



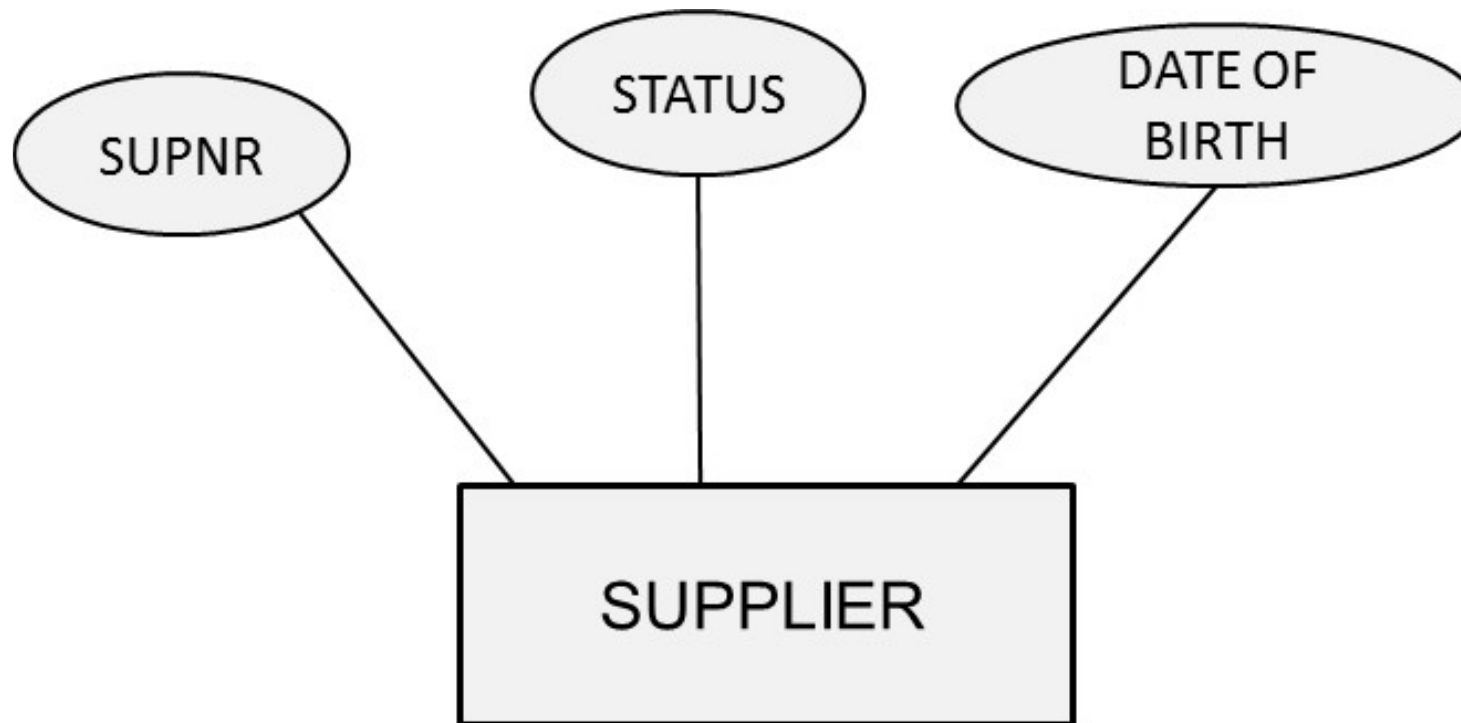
SUPPLIER

Typy atrybutów

- Typ atrybutu reprezentuje właściwość typu encji
 - np. nazwa i adres są typami atrybutu typu encji dostawca
- Atrybut to instancja typu atrybutu



Typy atrybutów



Typy atrybutów

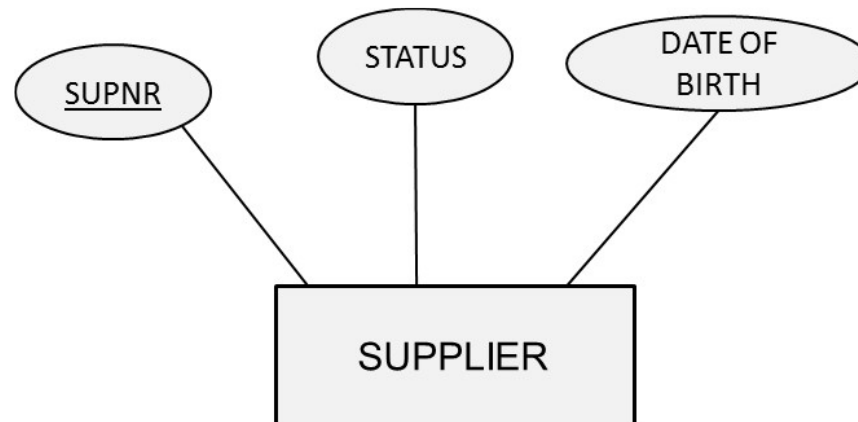
- Dziedziny
- Atrybuty klucza
- Proste a złożone atrybuty
- Jednowartościowe a wielowartościowe atrybuty
- Atrybuty pochodne

Dziedziny

- Dziedzina określa zbiór wartości, które mogą być przypisane do atrybutu każdej pojedynczej encji
 - np. płeć: kobieta i mężczyzna
- Dziedzina może również zawierać wartości *null*
 - *null*: 1) wartość nieznana 2) wartość istnieje, ale nie jest dostępna 3) atrybut nie dotyczy tej krotki
- Dziedziny nie są przedstawiane w modelu ER

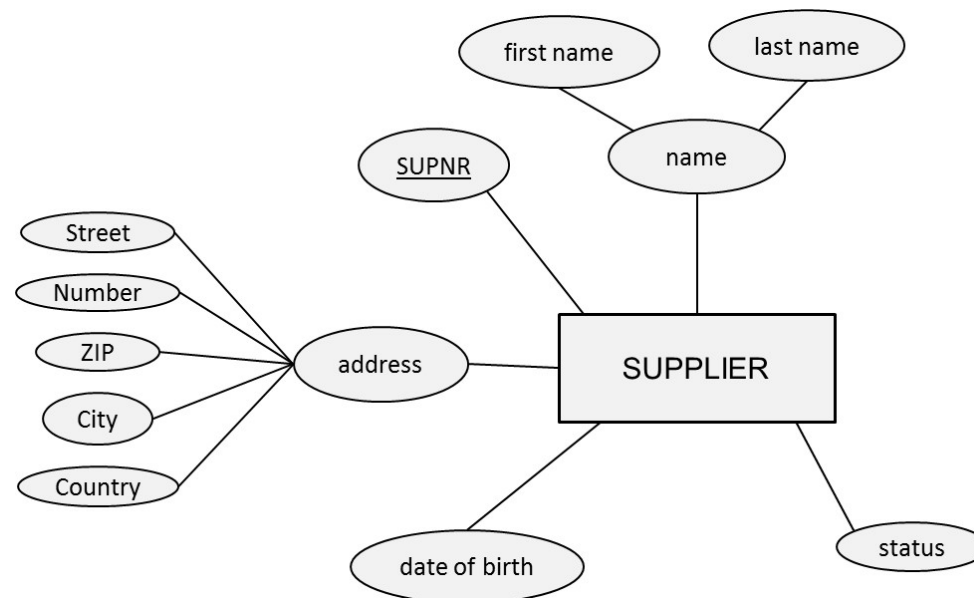
Atrybut klucza

- Atrybut, którego wartości są różne dla każdej pojedynczej encji
 - np.: identyfikator dostawcy, numer produktu, numer ubezpieczenia
- Atrybut klucza może być kombinacją atrybutów
 - np.: kombinacja numeru lotu i daty wylotu



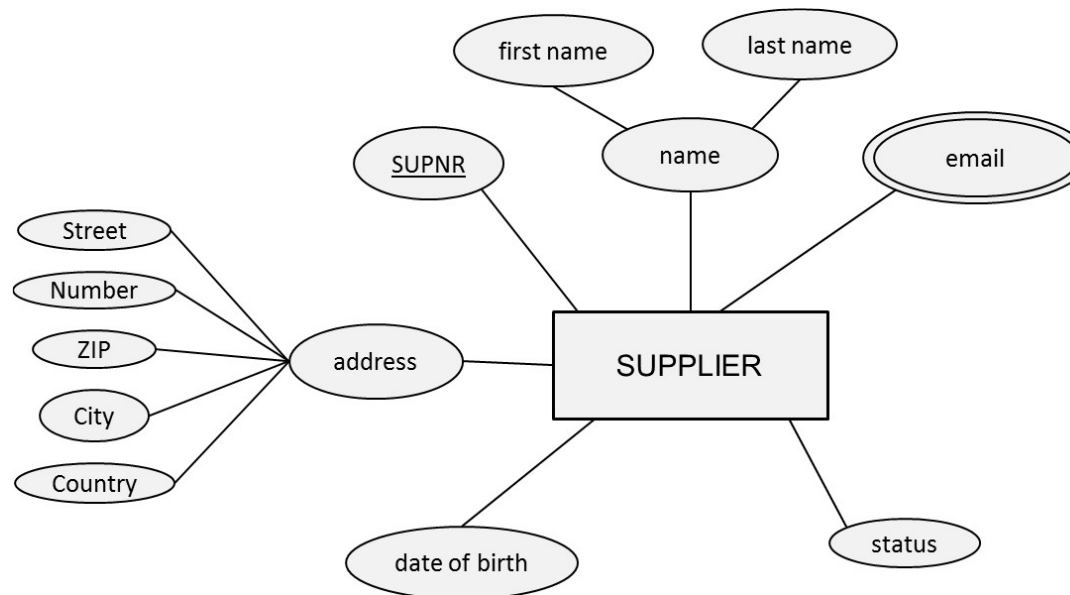
Proste a złożone atrybuty

- Proste (atomowe) atrybuty nie mogą być dalej dzielone na części
 - np. numer dostawcy, status dostawcy
- Złożone atrybuty można rozłożyć na inne znaczące atrybuty
 - np. adres, nazwa



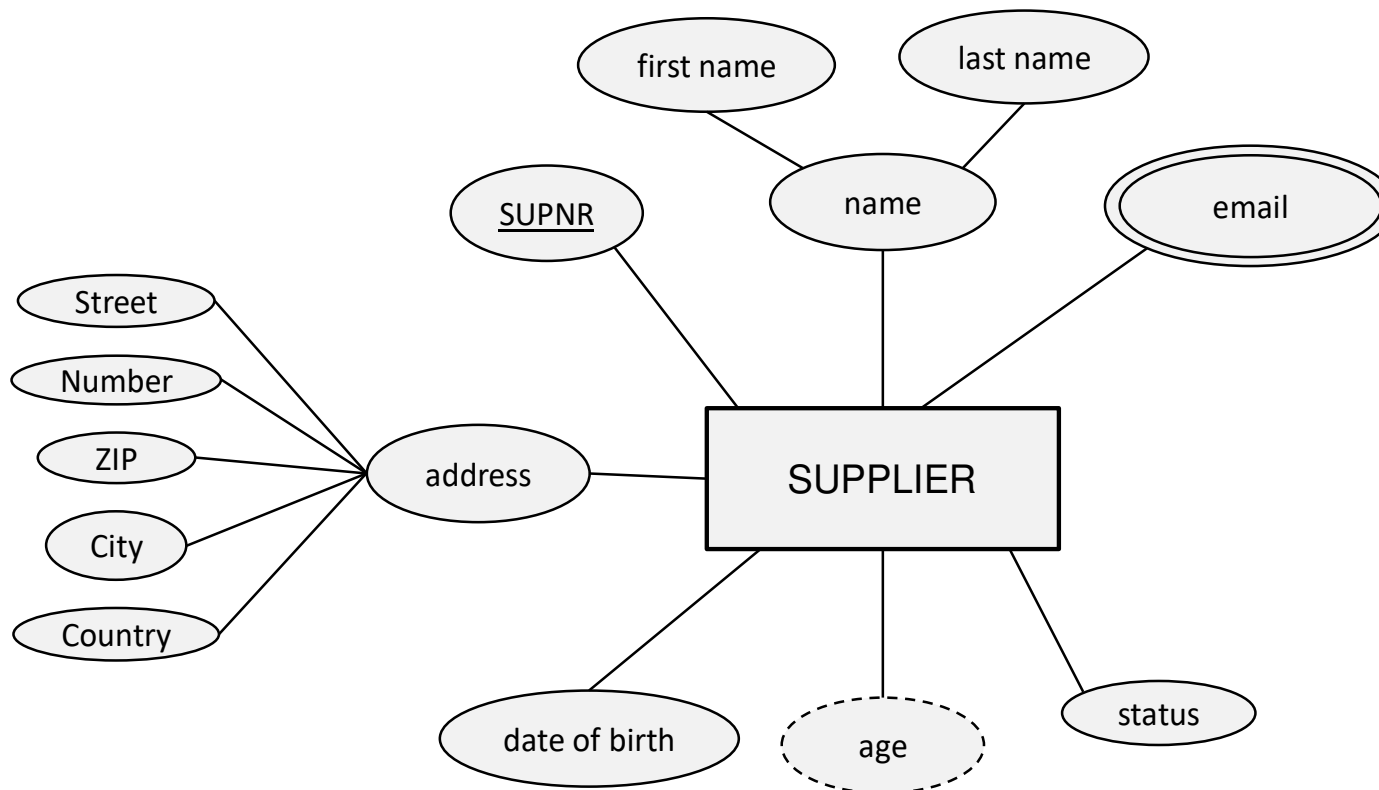
Jednowartościowe a wielowartościowe atrybuty

- Jednowartościowy atrybut ma tylko jedną wartość dla określonej encji
 - np. numer produktu, nazwa produktu
- Wielowartościowy atrybut może mieć wiele wartości
 - np. adresy e-mailowe



Pochodny atrybut

- Można wyprowadzić z innego typu atrybutu
– np. wiek

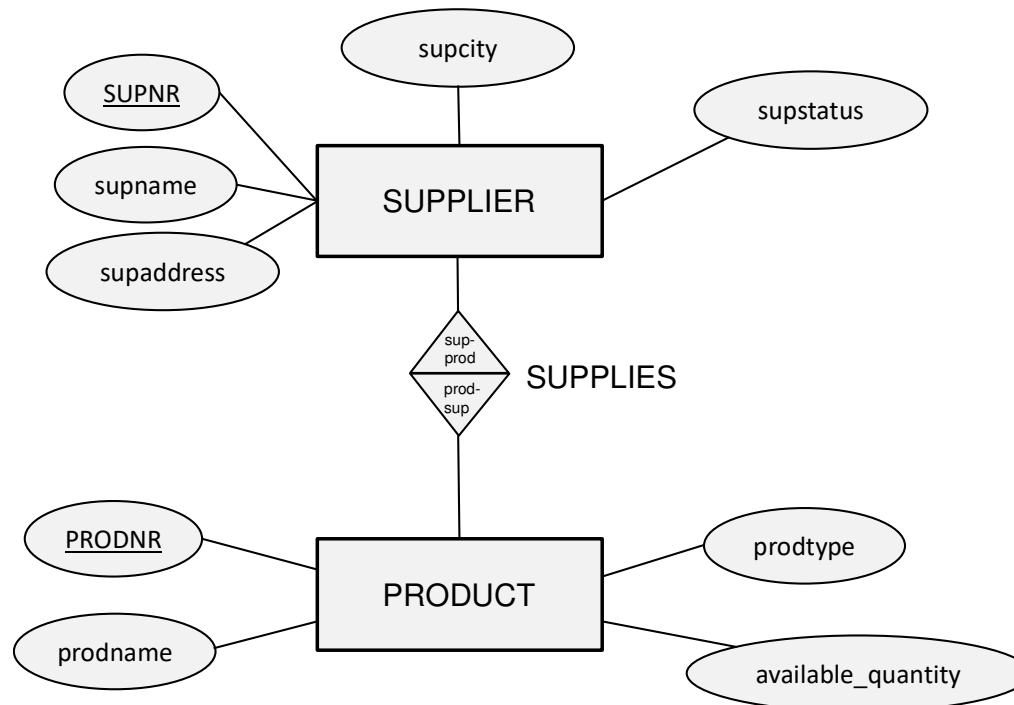


Typy związków

- Definicja
- Stopień i role
- Ograniczenia typu związku
- Atrybuty związków

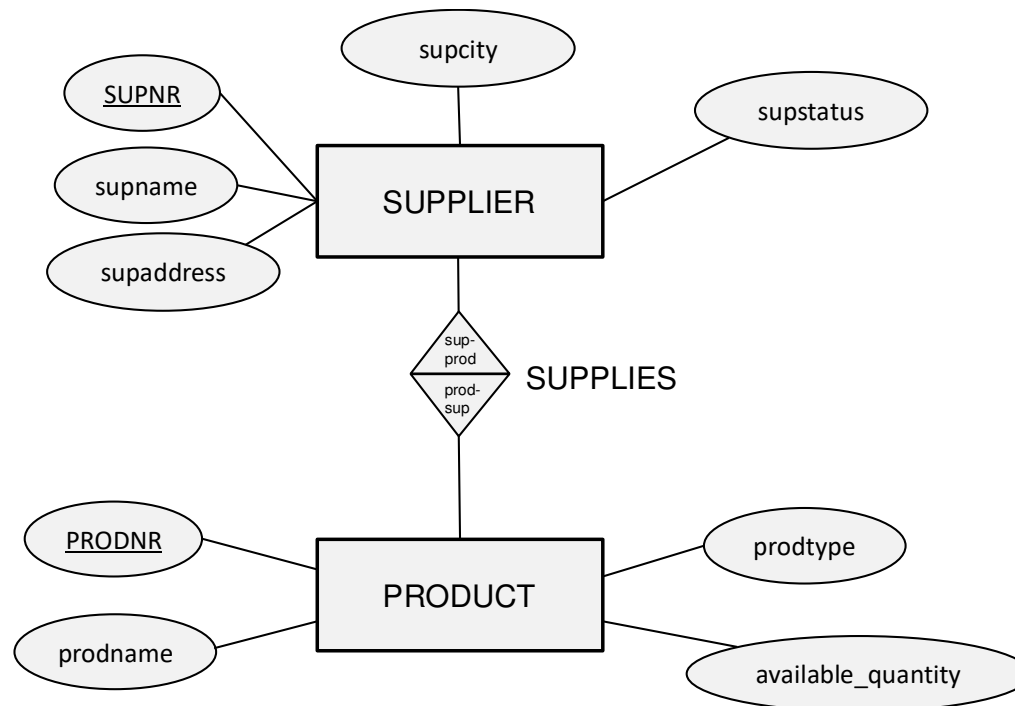
Definicja

- Związek reprezentuje zależność między dwiema lub większą liczbą encji
- Typ związku definiuje zbiór związków między instancjami jednej, dwóch lub większej liczby typów encji

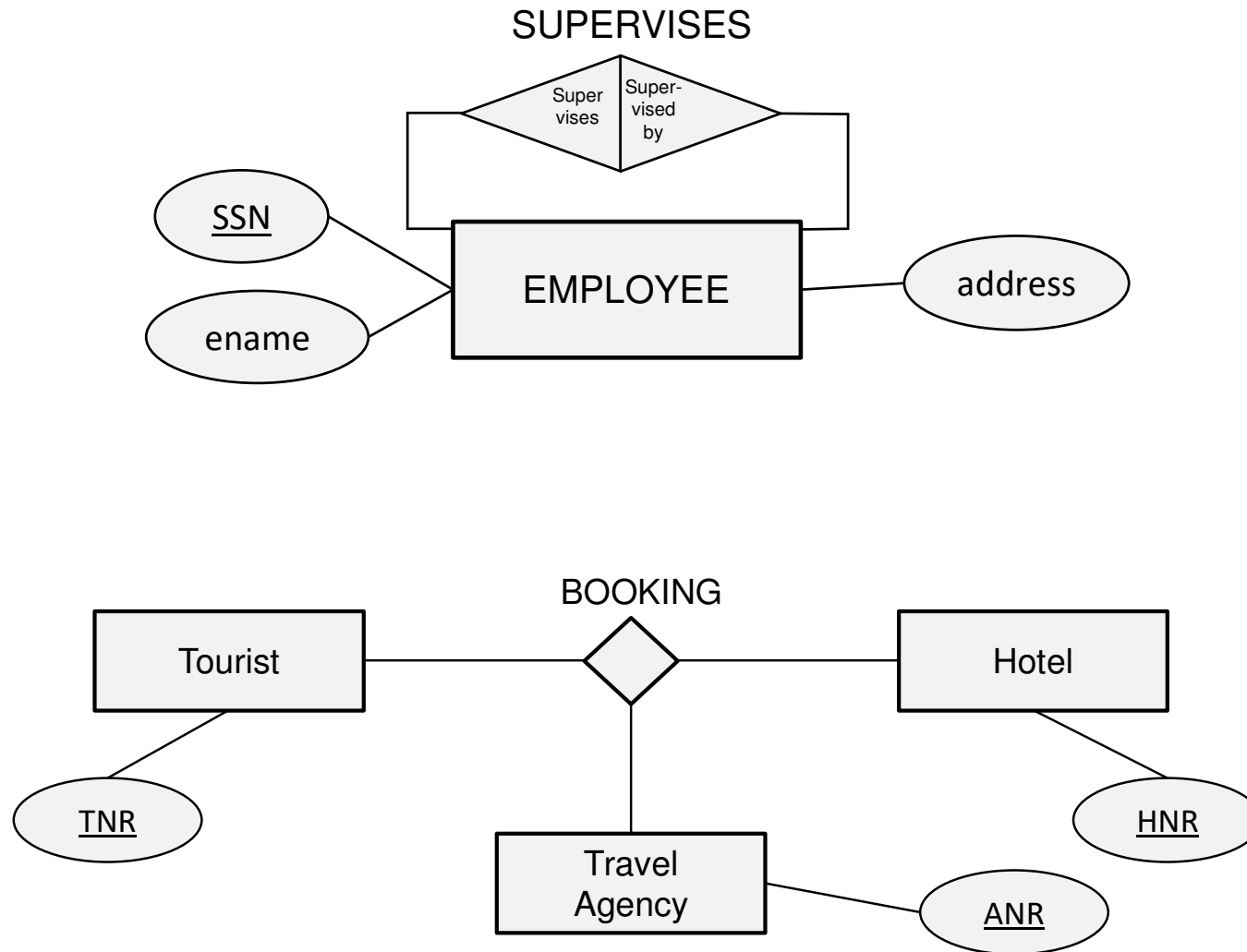


Stopień i role

- Stopień typu związków odpowiada liczbie typów encji uczestniczących w typie związków
 - Unarny: stopień 1, binarny: stopień 2, ternarny: stopień 3
- Role typu związku wskazują różne kierunki, które można wykorzystać do jego interpretacji



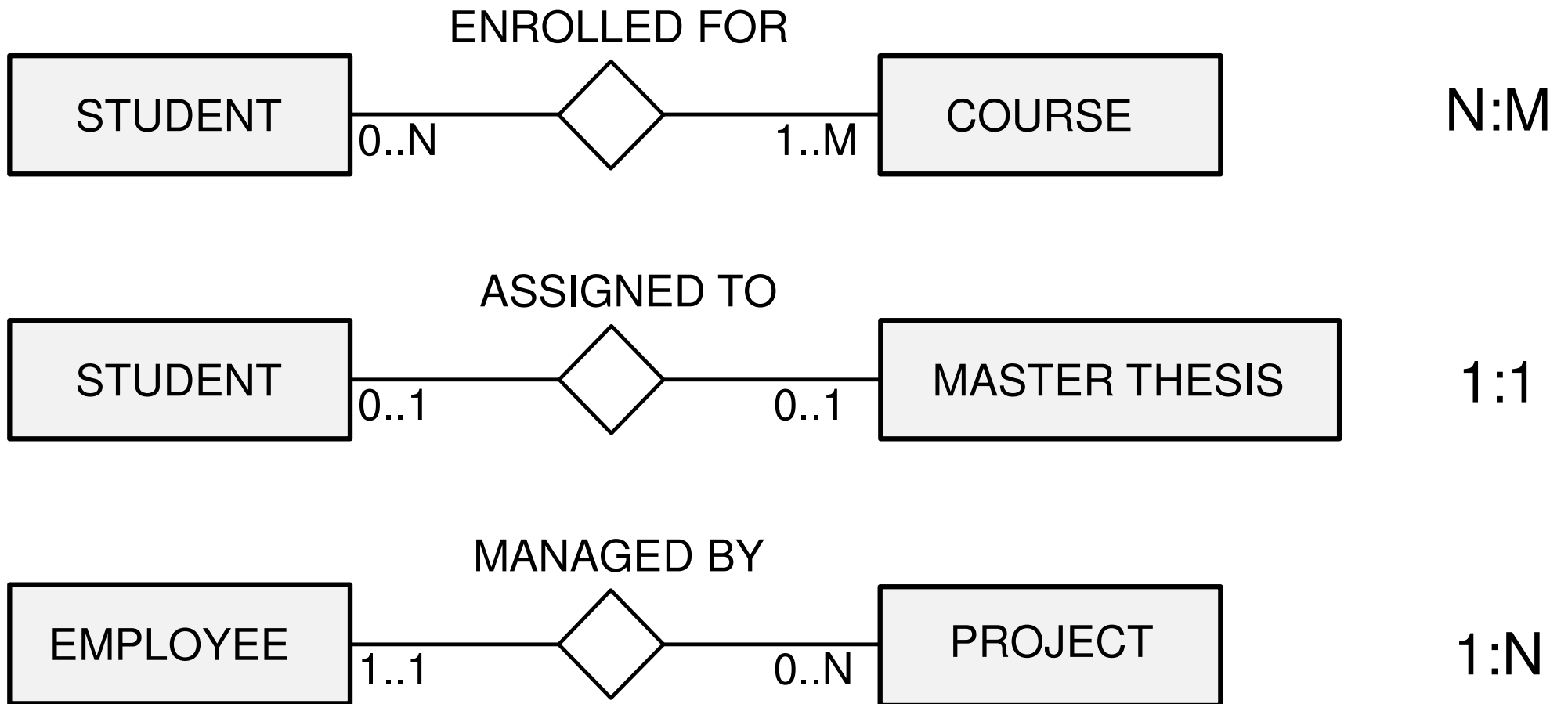
Stopień i role



Współczynnik liczności (cardinality)

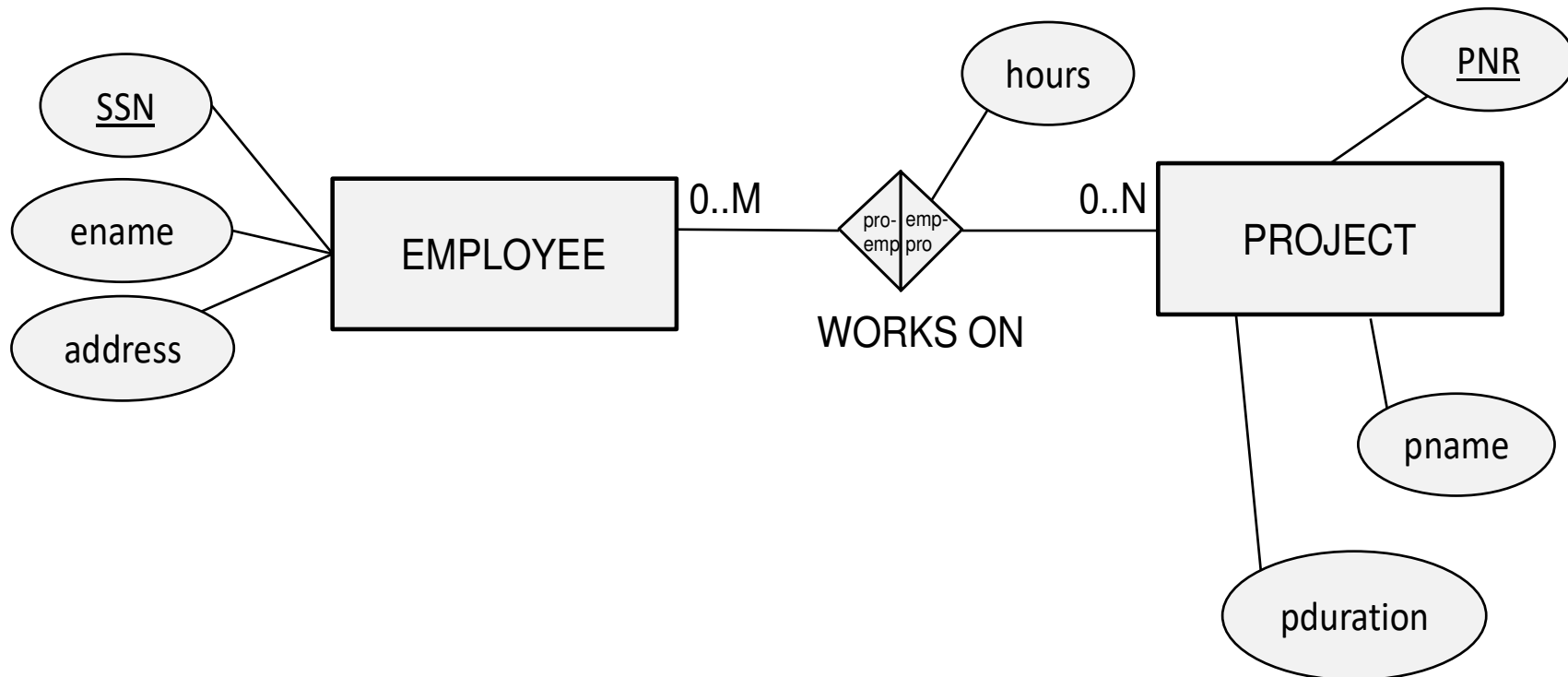
- Każdy typ związku można scharakteryzować pod względem współczynnika liczności, który określa minimalną lub maksymalną liczbę wystąpień związku, w których może uczestniczyć pojedyncza encja
- Minimalny współczynnik liczności może być 0 lub 1
 - jeżeli 0: częściowy udział
 - jeżeli 1: całkowity udział lub zależność istnienia
- Maksymalny współczynnik liczności: 1 lub N
- Typy związków zazwyczaj charakteryzuje się maksymalnym współczynnikiem liczności
 - 4 opcje dla binarnych typów związków: 1:1, 1:N, N:1 i M:N.

Współczynnik liczności



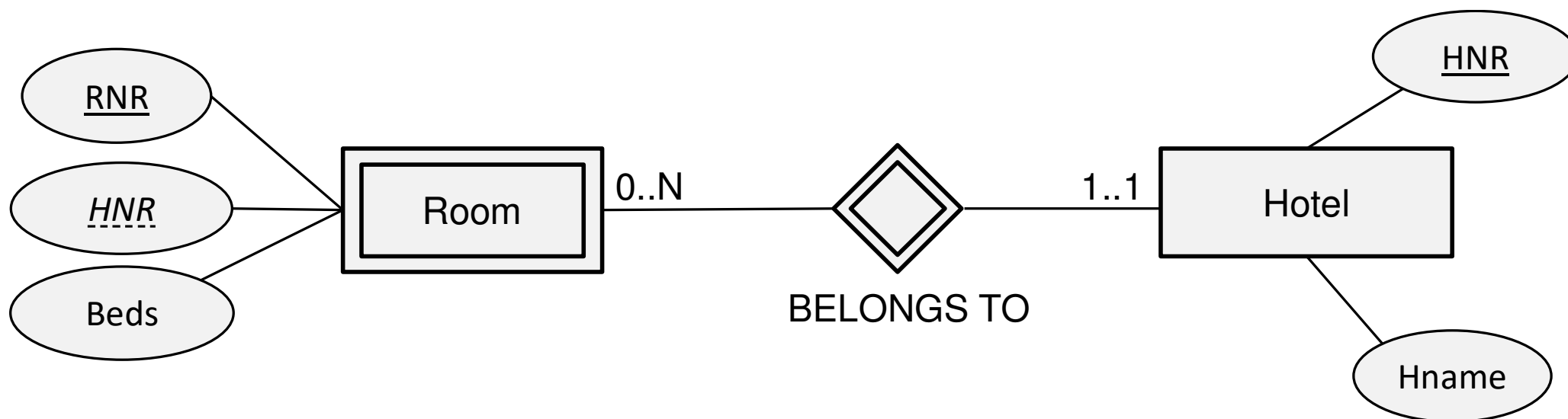
Atrybuty typów związków

- Typy związków mogą również mieć atrybuty
- Te atrybuty można przypisać do jednego z uczestniczących typów encji w przypadku typu relacji 1:1 lub 1:N



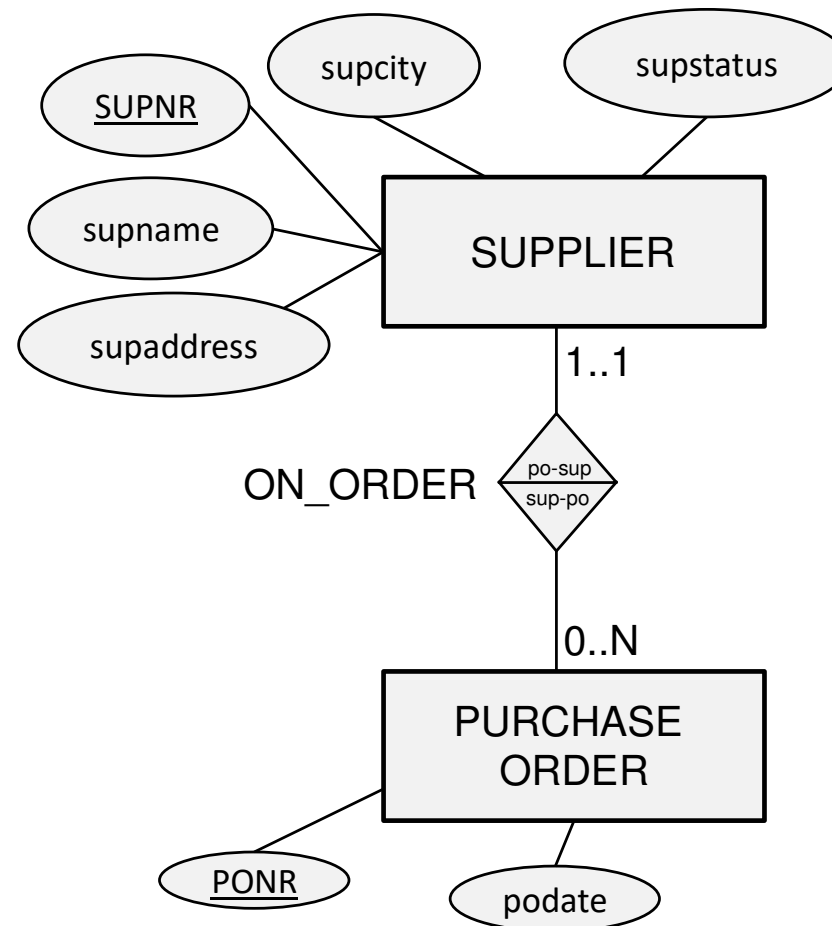
Słabe typy encji

- Silny typ encji posiada atrybut klucza
- Słaby typ encji nie ma własnego atrybuta klucza
 - powiązane z typem encji właściciela, od którego pożyczają atrybut, aby utworzyć atrybutu klucza



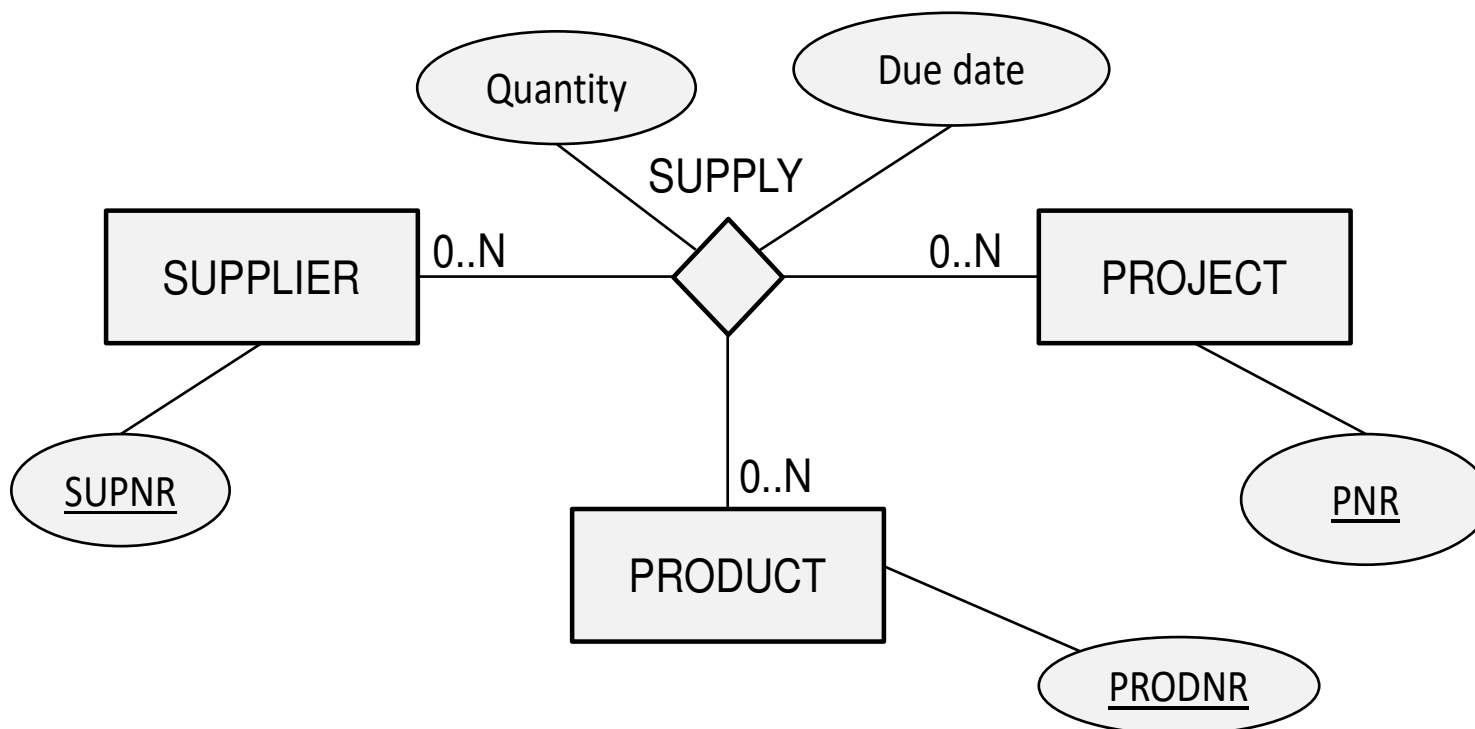
Słaby typ encji

- Słaby typ encji jest zawsze zależny od typu encji właściciela (nie odwrotnie)

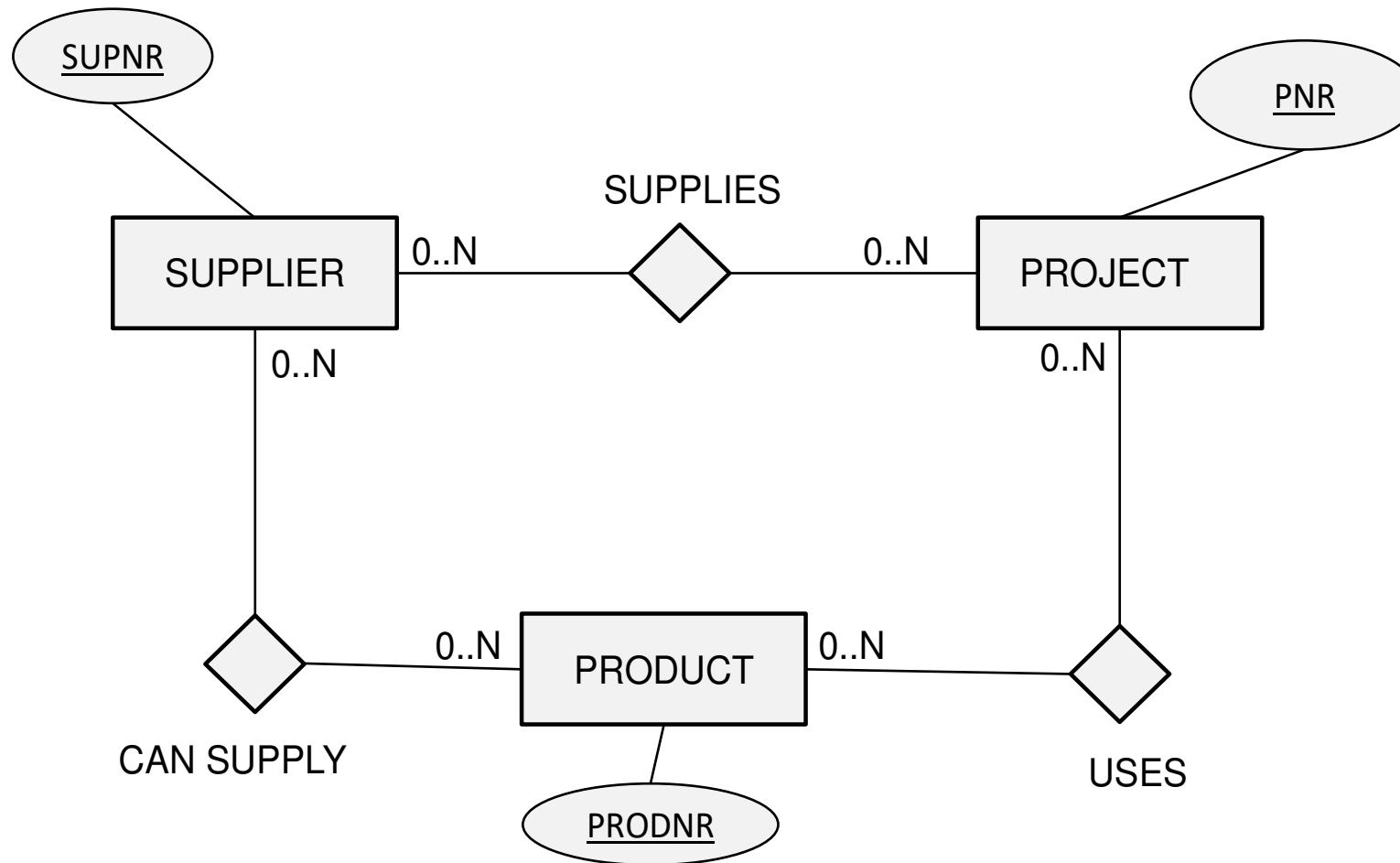


Trójskładnikowe (ternarne) typy związków

- Załóżmy, że mamy sytuację, w której dostawcy mogą dostarczać produkty do projektów. Dostawca może dostarczyć konkretny produkt do wielu projektów. Produkt do konkretnego projektu może być dostarczany przez wielu dostawców. W ramach projektu określony dostawca może dostarczać wiele produktów. Model musi również zawierać ilość i termin dostarczenia określonego produktu do konkretnego projektu przez konkretnego dostawcę.



Ternarne typy związków



Utrata semantyki

Ternarne typy związków

- Załóżmy, że mamy dwa projekty: projekt 1 używa ołówka i pióra, a projekt 2 używa pióra. Dostawca Peters dostarcza ołówek do projektu 1 i pióro do projektu 2, podczas gdy dostawca Johnson dostarcza pióro do projektu 1.

SUPPLY

Supplier	Product	Project
Peters	Pencil	Project 1
Peters	Pen	Project 2
Johnson	Pen	Project 1

SUPPLIES

Supplier	Project
Peters	Project 1
Peters	Project 2
Johnson	Project 1

USES

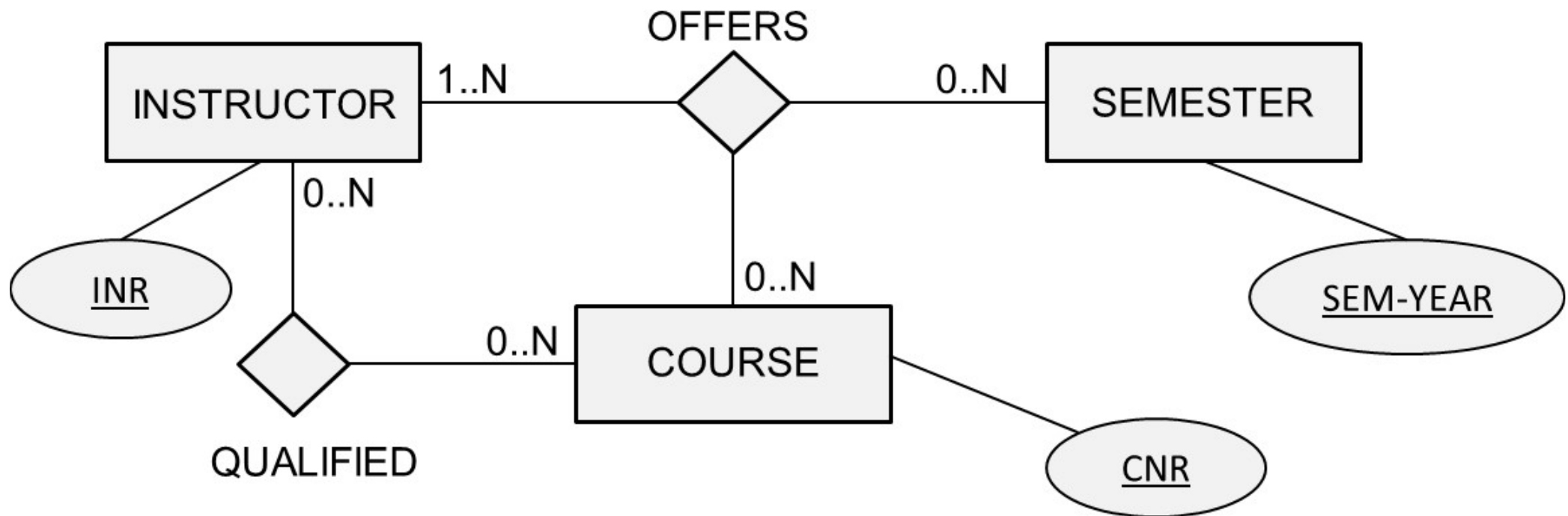
Product	Project
Pencil	Project 1
Pen	Project 1
Pen	Project 2

CAN SUPPLY

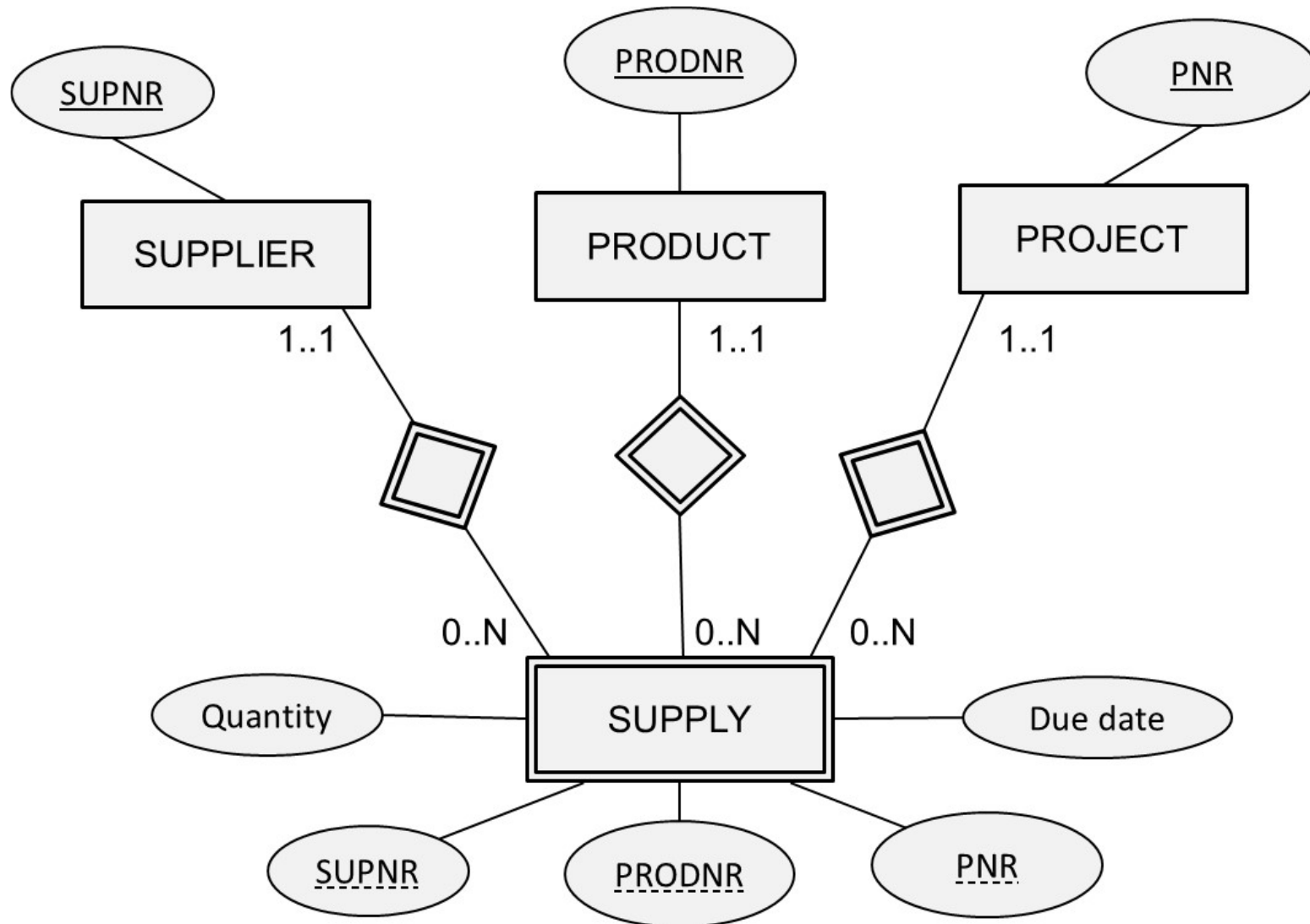
Supplier	Product
Peters	Pencil
Peters	Pen
Johnson	Pen

- Z binarnych typów związków nie jest jasne, kto dostarcza pióro do projektu 1

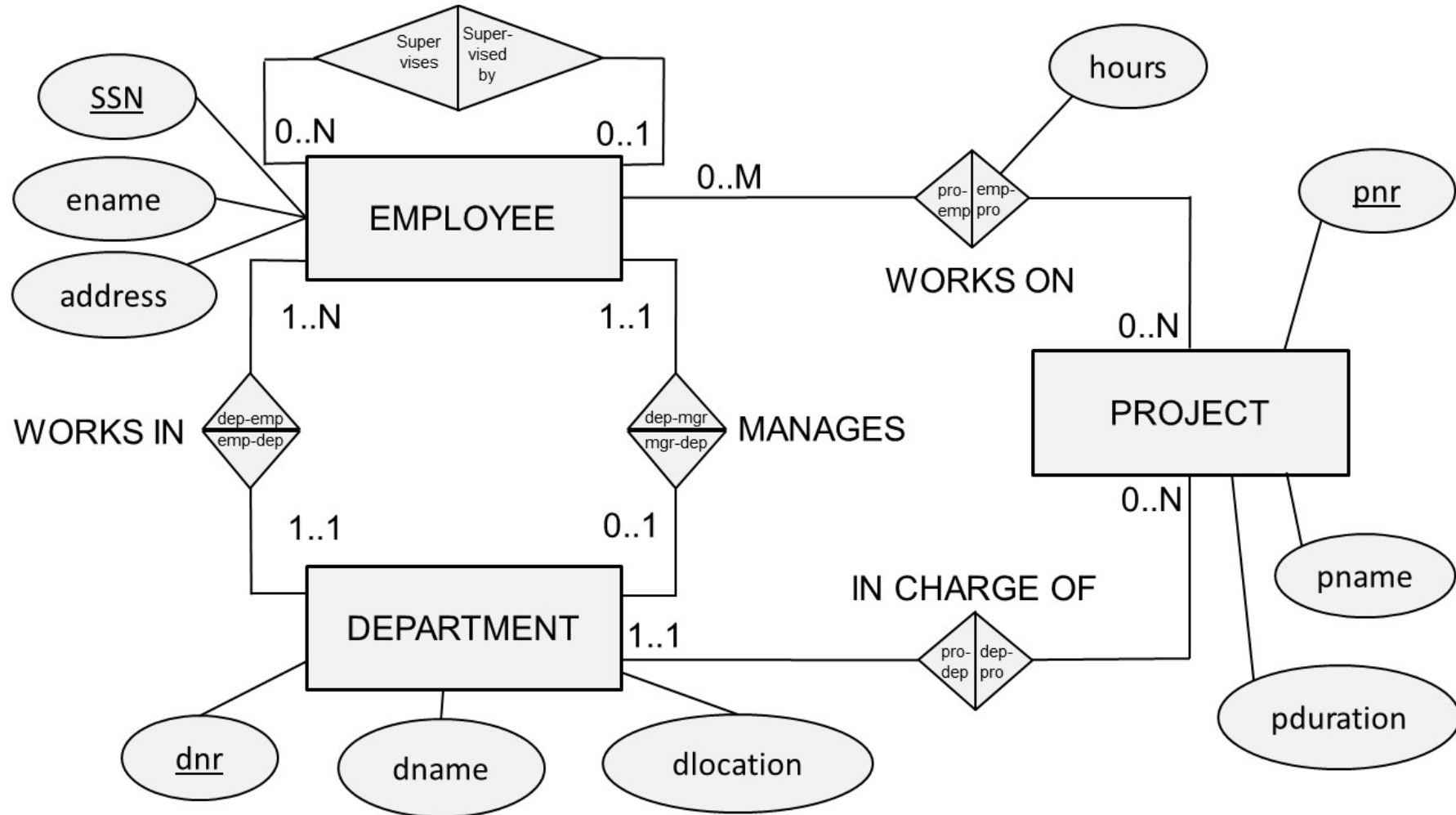
Trójskładnikowe typy związków



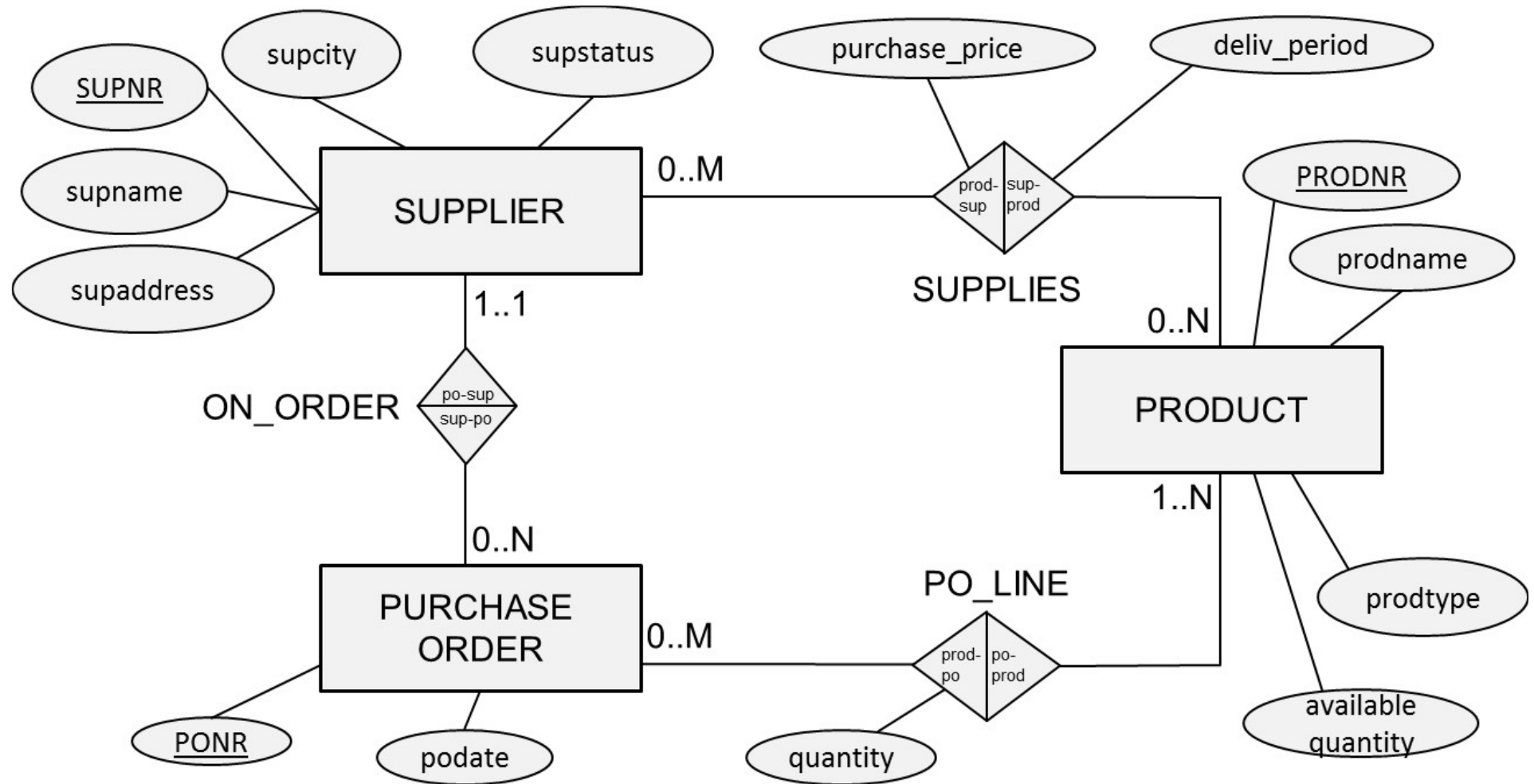
Trójskładnikowe typy związków



Model ER - przykład



Model ER - przykład



Ograniczenia modelu ER

- Model ER przedstawia tymczasową migawkę i nie jest w stanie modelować ograniczeń czasowych
 - np.: projekt musi być przypisany do działu po miesiącu, pracownik nie może wrócić do działu, którego wcześniej był kierownikiem, po dwóch tygodniach musi zostać przydzielone zamówienie do dostawcy itp.
- Model ER nie może zagwarantować spójności w wielu typach związków
 - np.: pracownik powinien pracować w dziale, którym zarządza, pracownicy powinni pracować nad projektami przypisanymi do działów, do których należą, dostawcy mogą być przydzielani tylko do zamówień zakupu produktów, które mogą dostarczyć

Ograniczenia modelu ER

- Dziedziny nie są uwzględnione w modelu ER
 - np.: godziny powinny być dodatnie; prodtype musi być czerwony, biały lub niebieski, supstatus jest liczbą całkowitą od 0 do 100
- Funkcje nie są uwzględnione w modelu ER
 - np.: obliczyć średnią liczbę projektów, nad którymi pracuje pracownik; określić, który dostawca pobiera maksymalną cenę za produkt

Rozszerzony model EER

(Enhanced Entity Relationship)

- Specjalizacja/Generalizacja
- Kategoryzacja
- Agregacja
- Przykłady modelu EER
- Projektowanie modelu EER

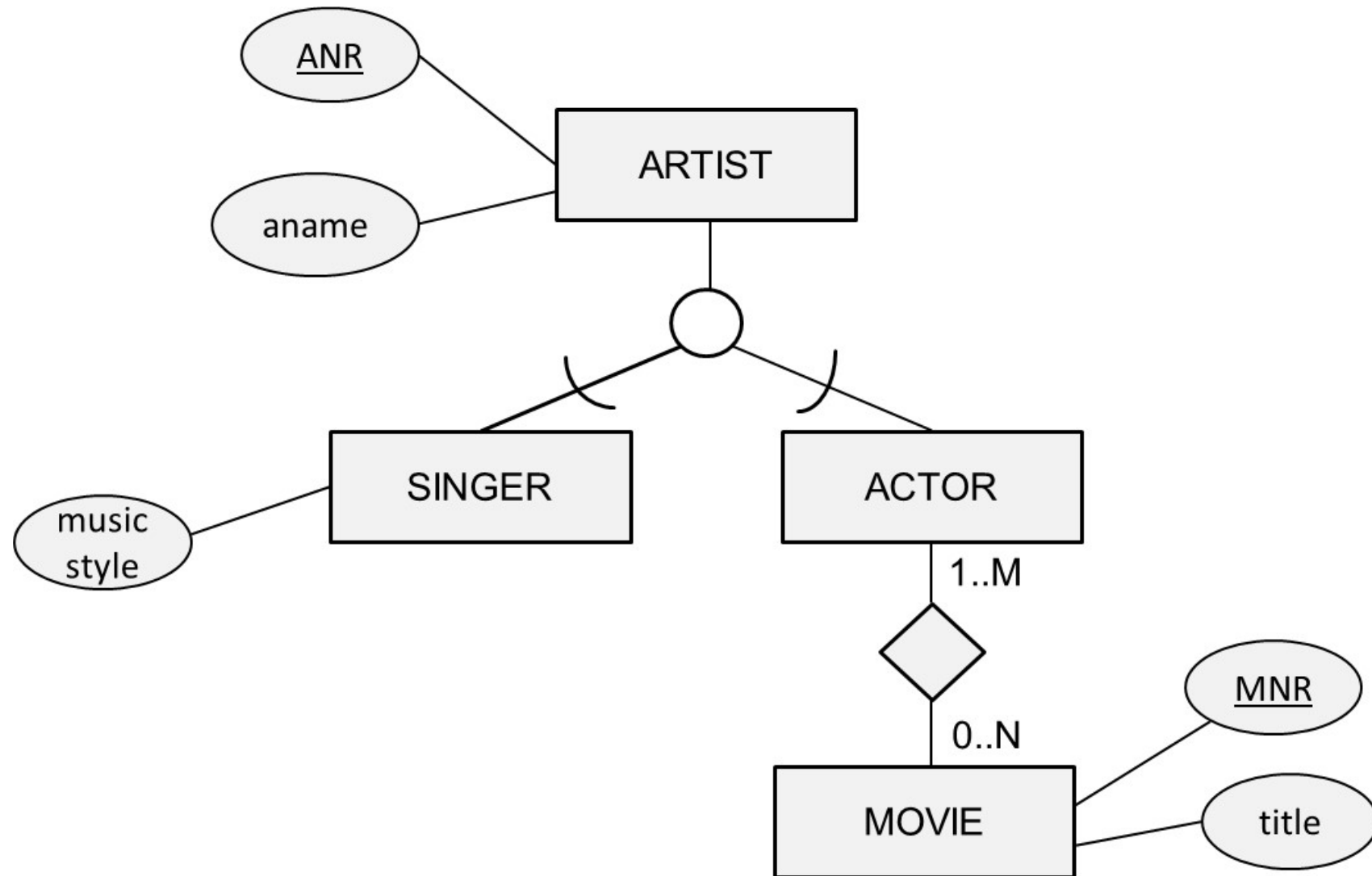
Specjalizacja/Generalizacja

- Specjalizacja odnosi się do procesu definiowania zbioru podklas typu encji
 - Przykład: superklasa ARTIST z podklasami SINGER i ACTOR
- Proces specjalizacji definiuje relację „IS A”
- Specjalizacja może następnie ustalić dodatkowe specyficzne typy atrybutów dla każdej podklasy
 - Przykład: piosenkarz może mieć typ atrybutu stylu muzycznego
- Specjalizacja może również ustanowić dodatkowe specyficzne typy relacji dla każdej podklasy
 - Przykłady: aktor może występować w filmach, piosenkarz może być częścią zespołu
- Podklasa dziedziczy wszystkie typy atrybutów i typy relacji ze swojej nadklasy

Specjalizacja/Generalizacja

- Generalizacja, zwana również abstrakcją, to odwrotny proces specjalizacji
 - Specjalizacja odpowiada procesowi top-down udoskonalania koncepcji (conceptual refinement)
 - Generalizacja odpowiada procesowi bottom-up syntezy pojęciowej (conceptual synthesis)

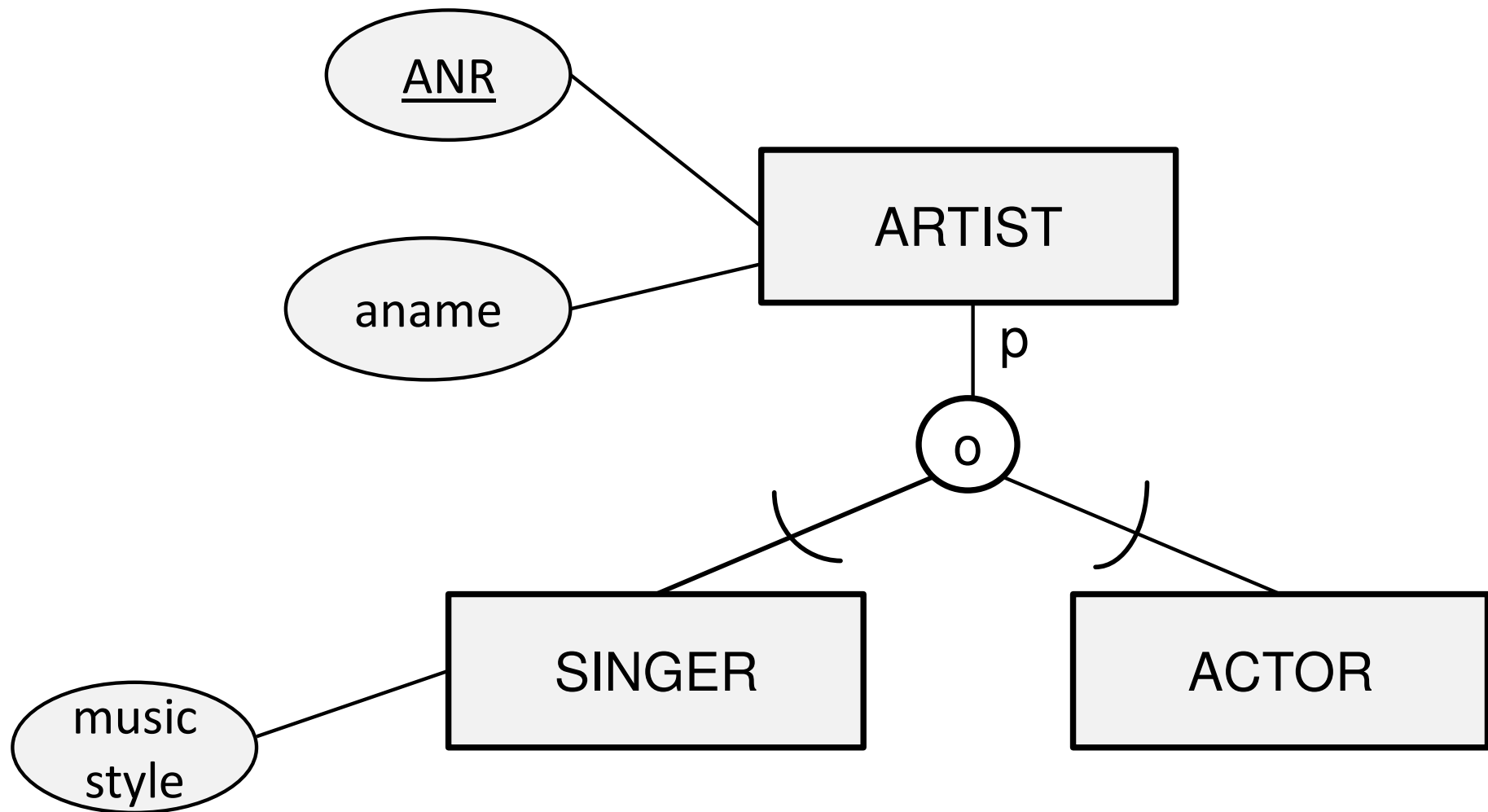
Specjalizacja/Generalizacja



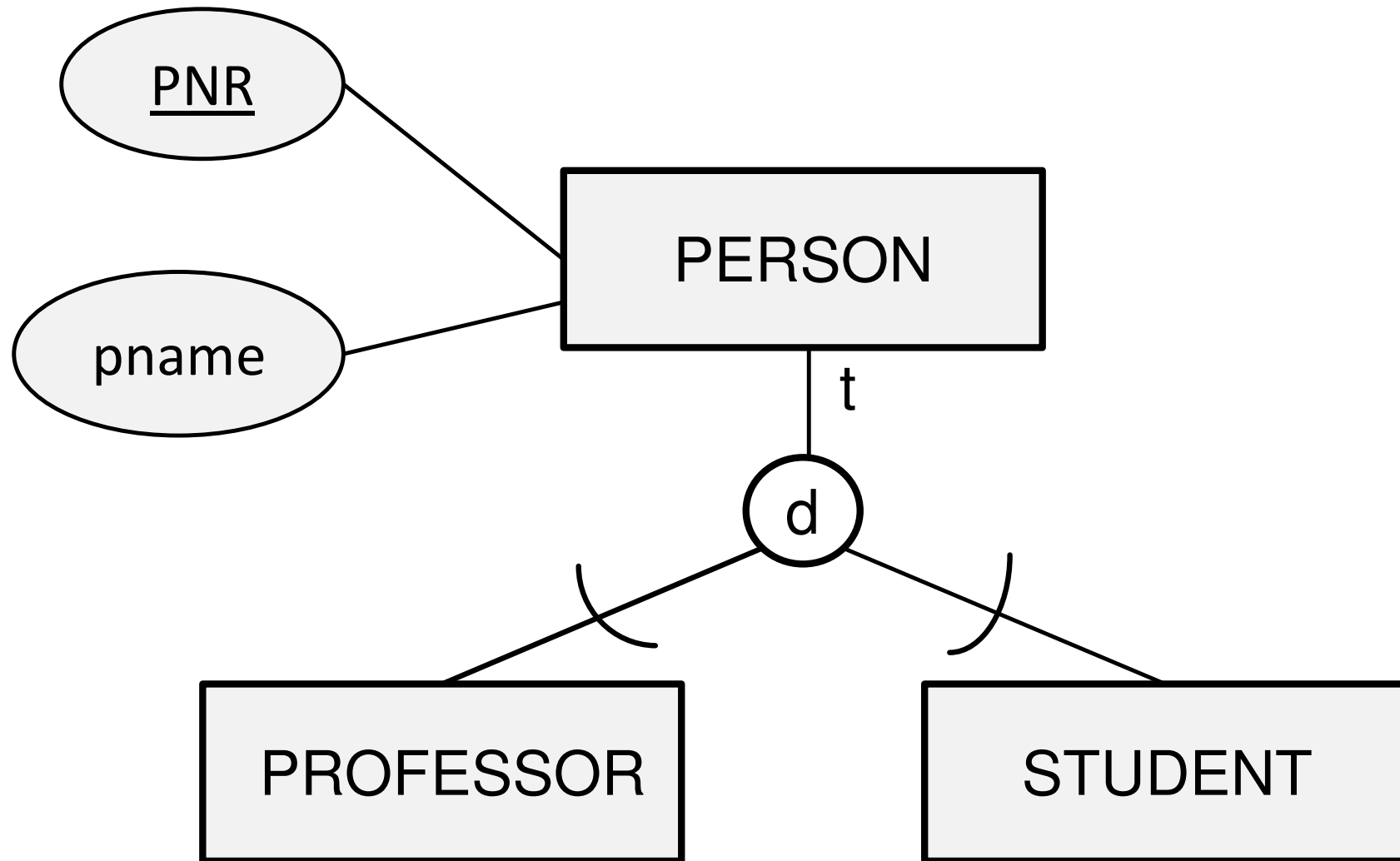
Specjalizacja/Generalizacja

- Ograniczenie rozłączności (disjointness constraint) określa, do jakich podklas może należeć encja nadklasy
 - Specjalizacja rozłączna to specjalizacja, w której jednostka może być członkiem co najwyżej jednej z podklas
 - Specjalizacja nakładająca się to specjalizacja, w której ta sama jednostka może należeć do więcej niż jednej podklasy
- Ograniczenie kompletności (completeness constraint) wskazuje, czy wszystkie encje nadklasy powinny należeć do jednej z podklas, czy nie
 - Całkowita specjalizacja to specjalizacja, w której każda encja w superklasie musi być członkiem jakiejś podklasy
 - Częściowa specjalizacja pozwala encji należeć tylko do nadklasy i do żadnej z podklas

Specjalizacja/Generalizacja

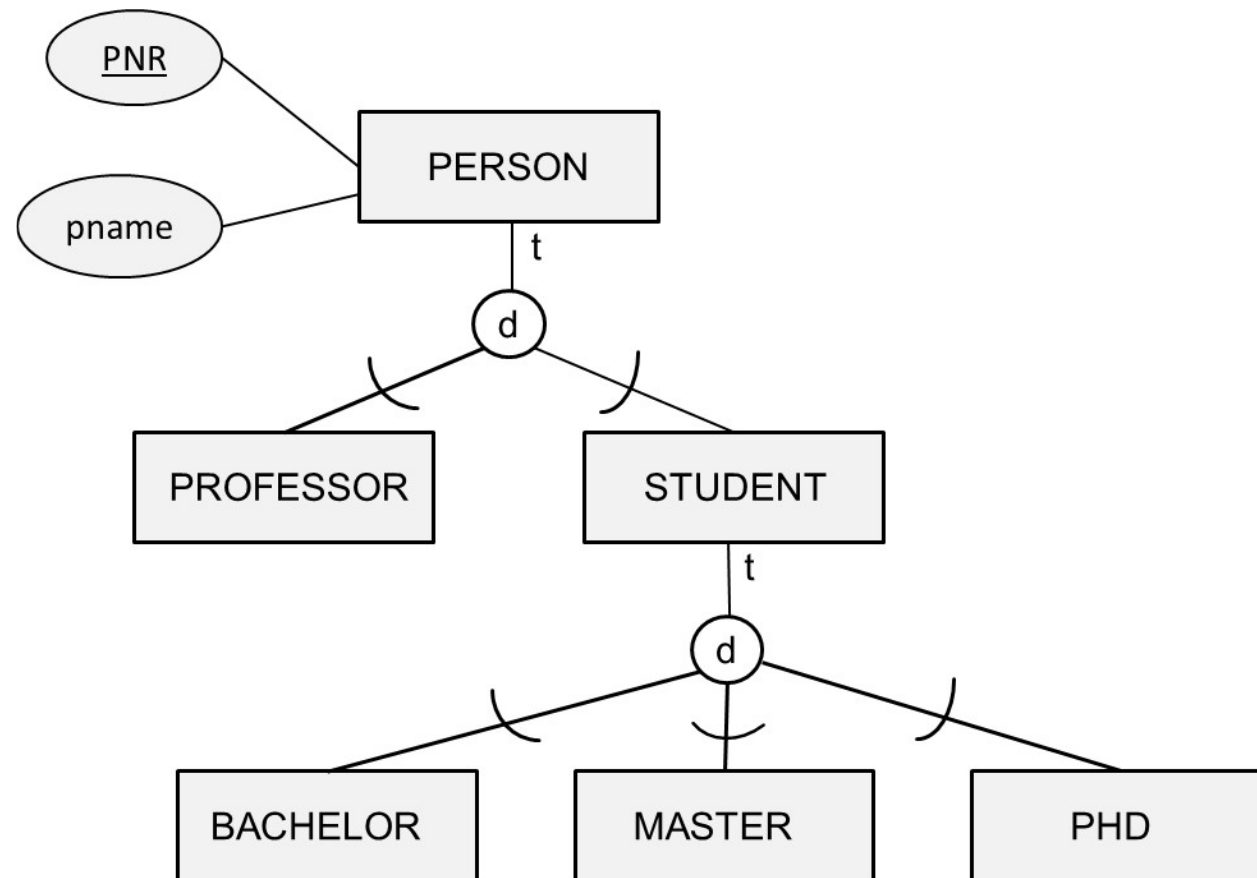


Specjalizacja/Generalizacja



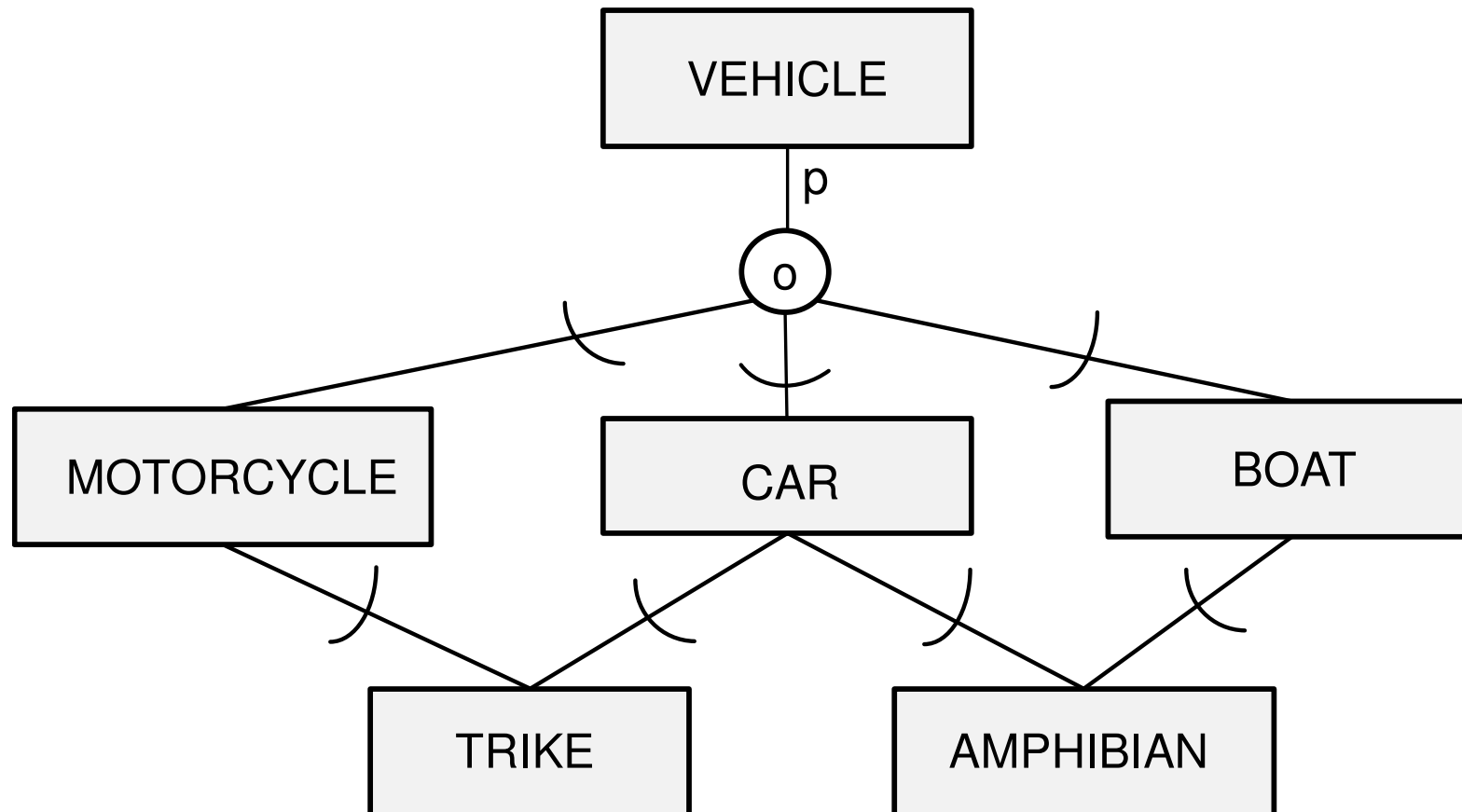
Specjalizacja/Generalizacja

- W hierarchii specjalizacji każda podklasa może mieć tylko jedną nadklasę i dziedziczy typy atrybutów i typy relacji wszystkich poprzedzających ją nadklas aż do korzenia hierarchii



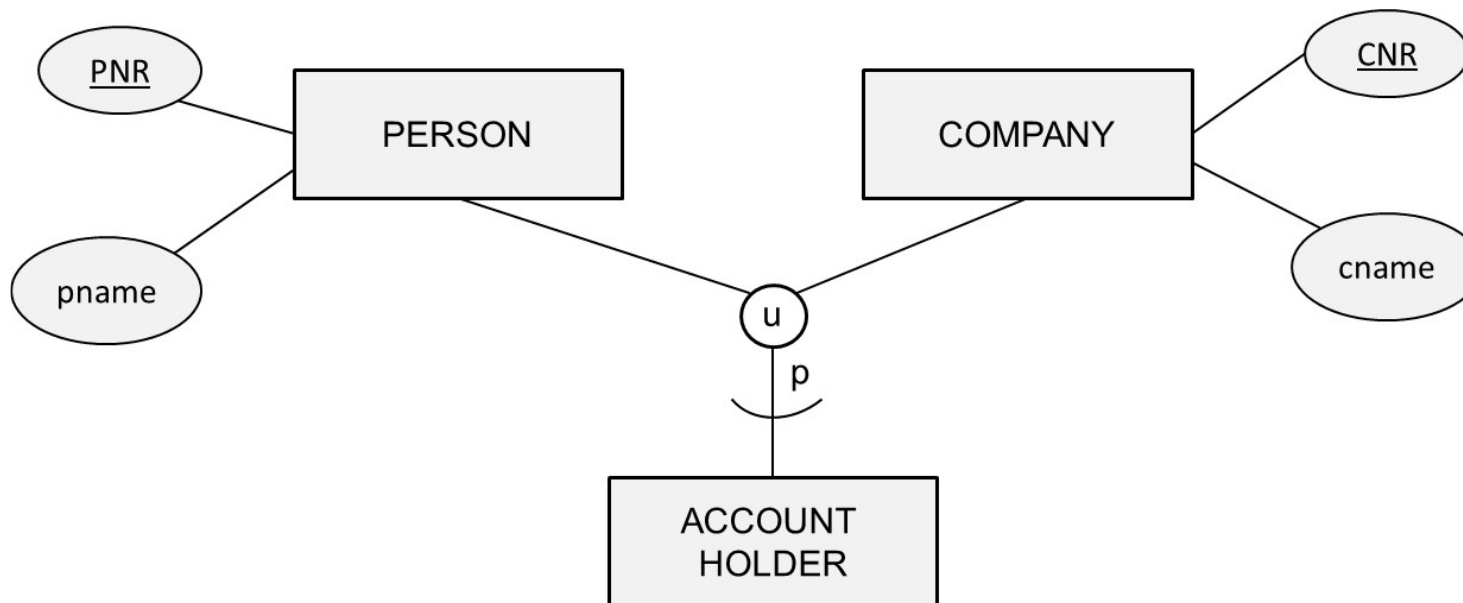
Specjalizacja/Generalizacja

- W sieci specjalizacji podklasa może mieć wiele nadklas (wielokrotne dziedziczenie)



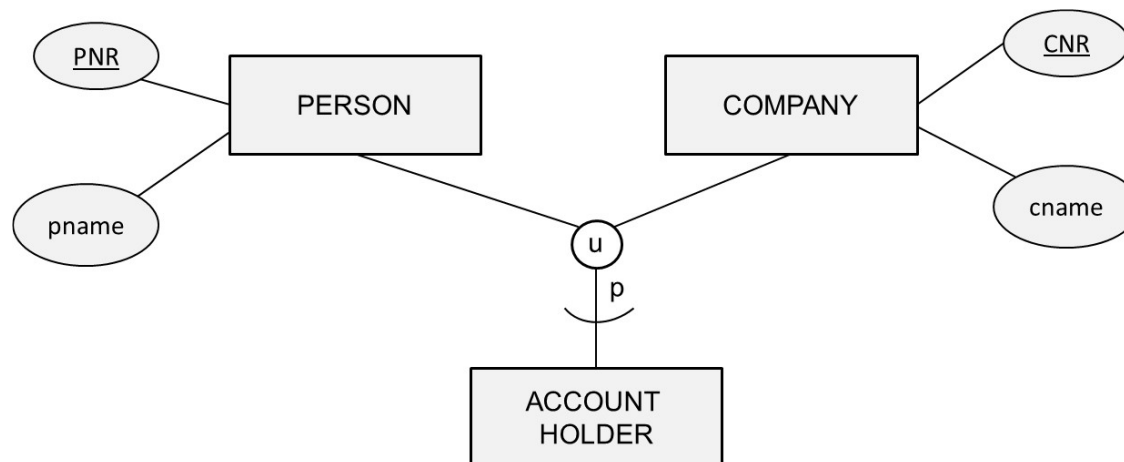
Kategoryzacja

- Kategoria to podklasa, która ma kilka możliwych nadklas
- Każda superklasa reprezentuje inny typ encji
- Kategoria reprezentuje zbiór encji, który jest podzbiorem unii nadklas



Kategoryzacja

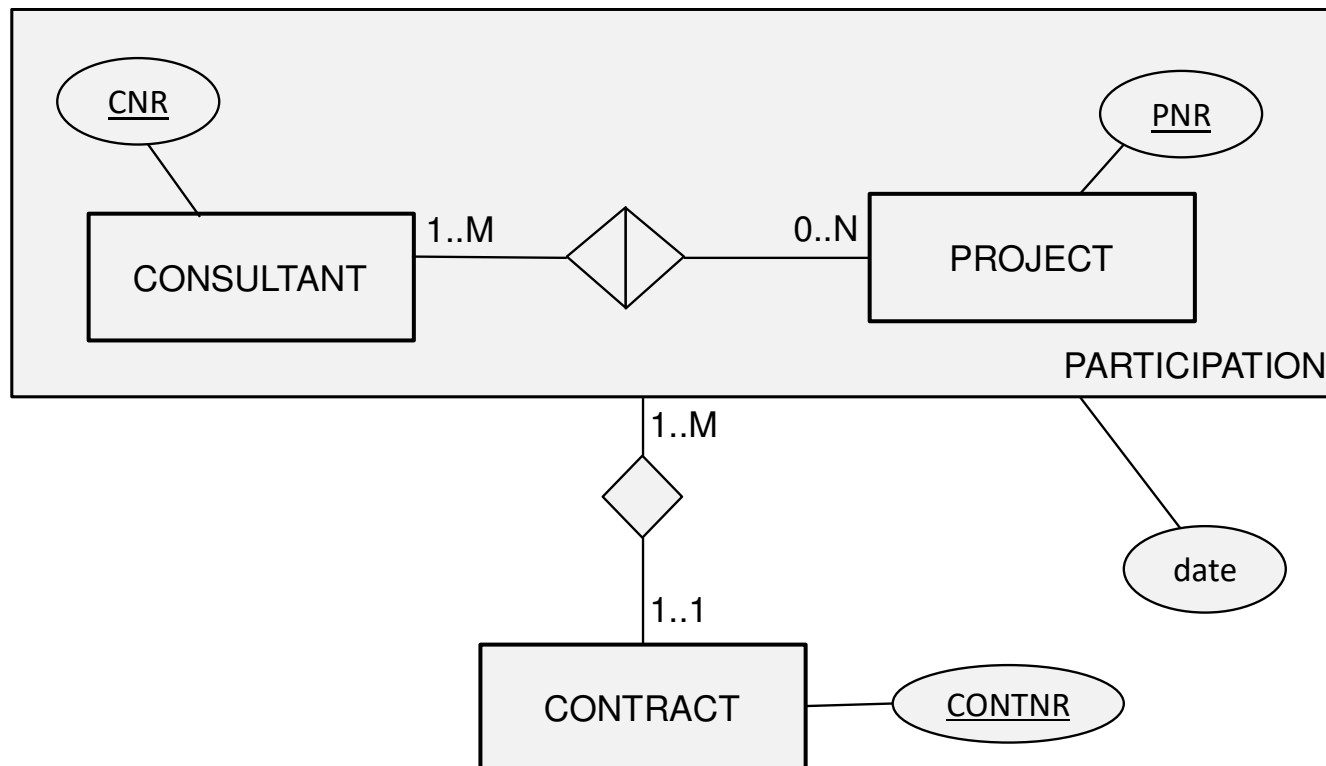
- Dziedziczenie w przypadku kategoryzacji odpowiada encji dziedziczącej tylko atrybuty i relacje tej nadklasy, której jest członkiem (dziedziczenie selektywne)
- Kategoryzacja może być całkowita lub częściowa
 - Całkowita: wszystkie encje nadklas należą do podklasy
 - Częściowa: nie wszystkie encje nadklas należą do podklasy



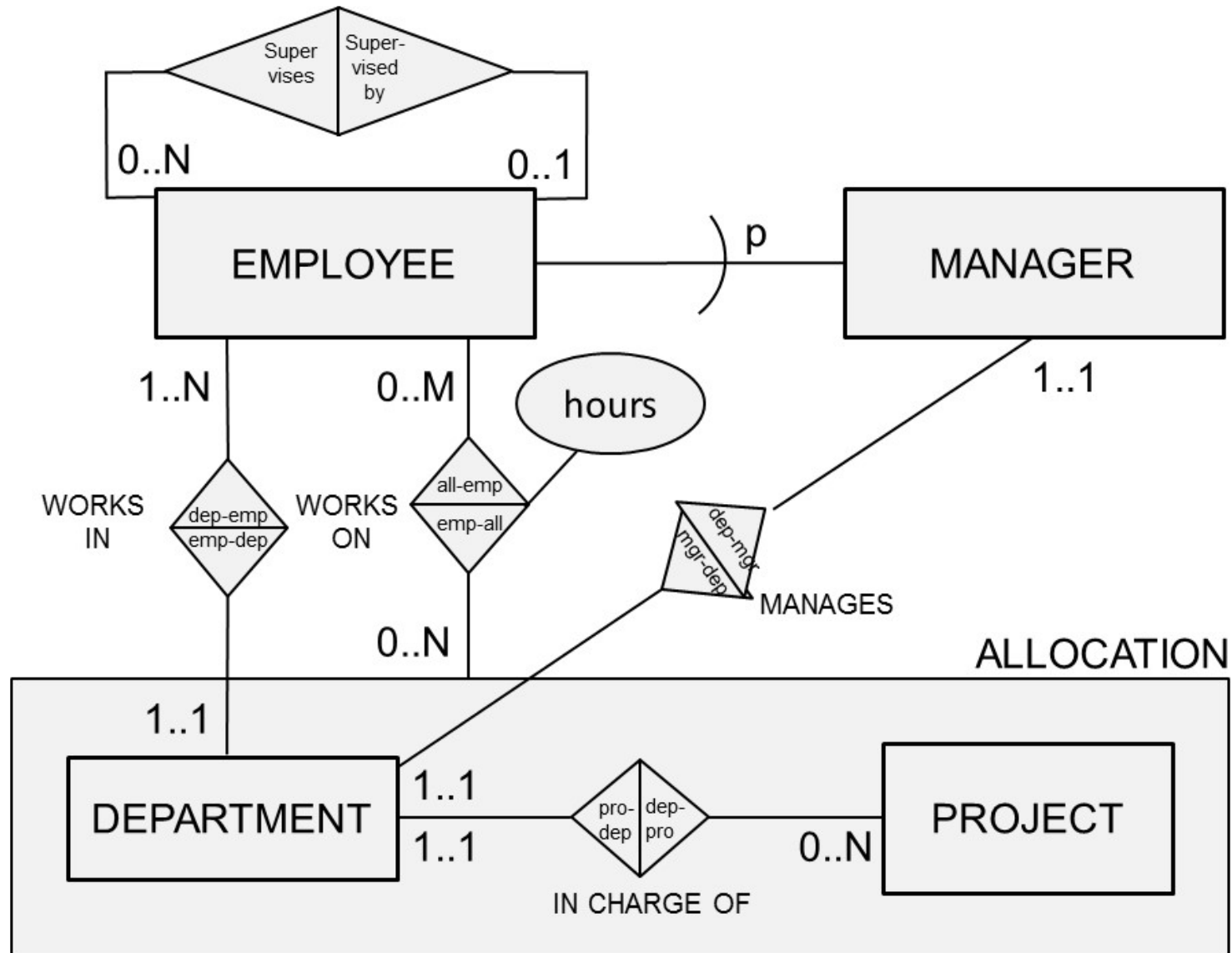
Uwaga: całkowita kategoryzacja może być również reprezentowana jako specjalizacja/generalizacja

Agregacja

- Typy encji, które są powiązane przez określony typ związku, można łączyć lub agregować w zagregowany typ encji wyższego poziomu
- Agregacja jest szczególnie przydatna, gdy typ zagregowanej encji ma swoje własne typy atrybutów i/lub typy związków



Przykład



Projektowanie modelu EER

1. Zidentyfikuj typy encji
2. Zidentyfikuj typy związków i potwierdź ich stopień
3. Podaj współczynniki liczności i ograniczenia uczestnictwa (uczestnictwo całkowite lub częściowe)
4. Zidentyfikuj typy atrybutów i sprawdź, czy są one proste czy złożone, mają jedną lub wiele wartości, są pochodne czy nie
5. Połącz każdy typ atrybutu z typem encji lub typem związku
6. Wskaż typ(y) atrybutów klucza każdego typu encji
7. Zidentyfikuj słabe typy jednostek i ich klucze częściowe
8. Zastosuj abstrakcje, takie jak generalizacja/specjalizacja, kategoryzacja i agregacja
9. Określ cechy każdej abstrakcji, takie jak rozłączne lub nakładające się, całkowite lub częściowe

Narzędzia

- Oracle Data Modeler
- Visual Paradigm
- Vertabelo
- SSMS
- Draw.io
- ...

Geneza UML

- Unified Modeling Language (UML) to język modelowania, który pomaga w specyfikacji, wizualizacji, konstrukcji i dokumentacji artefaktów systemu oprogramowania
- UML został zaakceptowany jako standard przez Object Management Group (OMG) w 1997 roku i zatwierdzony jako standard ISO w 2005 roku
- Najnowsza wersja to UML 2.5, wprowadzona w 2015 roku
- UML oferuje różne diagramy, takie jak diagramy przypadków użycia, diagramy sekwencji, diagramy pakietów, diagramy wdrożeń itp.
- Z punktu widzenia modelowania baz danych najważniejszy jest diagram klas

Obiektowość

- Klasa jest definicją schematu dla zbioru obiektów
 - Podobne do typu encji w ER
- I odwrotnie, obiekt jest instancją klasy
 - Podobne do encji w ER
- Obiekt jest charakteryzowany zarówno poprzez zmienne, jak i metody
 - Zmienne odpowiadają typom atrybutów, a wartości zmiennych atrybutom w ER
 - Brak odpowiednika ER dla metod

Obiektość

- Przykładowa klasa Student
 - Obiekt: Bart, Wilfried, Seppe
 - Przykładowe zmienne: imię i nazwisko ucznia, płeć i data urodzenia
 - Przykładowe metody: calcAge, isBirthday, hasPassed(courseID)
- Ukrywanie informacji (inaczej enkapsulacja) stwierdza, że do zmiennych obiektu można uzyskać dostęp tylko za pomocą metod pobierających (getter) lub ustawiających (setter)
 - metoda pobierająca służy do pobierania wartości zmiennej (getter)
 - metoda ustawiająca przypisuje mu wartość (setter)

Obiektoowość

- Dziedziczenie
 - Nadklasa może mieć jedną lub więcej podklas, które dziedziczą zarówno zmienne, jak i metody z nadklasy
- Przeciążanie metod
 - Różne metody w tej samej klasie mogą mieć tę samą nazwę, ale różną liczbę lub typ argumentów wejściowych