



Rzeczpospolita
Polska

Unia Europejska
Europejski Fundusz Społeczny



Projekt „Stawiamy na rozwój UKW”
współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego



instytut
informatyki

dr inż. Janusz Dorożyński

Przedmiot **Systemy informatyczne**

Studia I stopnia, tryb stacjonarny i niestacjonarny

Instrukcja laboratoryjna cz. 1 w6'1

Encje, notacja Barkera

Laboratorium z przedmiotu: **Systemy Informatyczne**
Studia I stopnia, tryb stacjonarny i niestacjonarny

spis

Encje, notacja Barkera

1. Wprowadzenie

Instrukcja dotyczy jednej z wielu używanych do modelowania notacji graficznego przedstawiania encji – notacji Barkera. Jest ona stosowana m.in. przez firmę Oracle, w takich narzędziach do modelowania, jak Oracle Designer (obecnie już nie wspierany, ale nadal używany) czy Oracle SQL Developer Data Modeler (SDDM).

1.1. Funkcje i informacje

Informatyka to nauka zajmująca się przetwarzaniem informacji za pomocą programów komputerowych. W skrócie, chodzi o to, że programy służą do przetwarzania danych, czyli informacji zawartych w różnych formach, na przykład w tekstach, obrazach czy dźwiękach.

1.2. Encje, atrybuty, powiązania, dziedziczenie opisowo

Oprogramowanie systemu informatycznego działa na danych opisujących elementy świata, na którym działa. Przykładowo, w przedsiębiorstwie handlowym każdy klient, towar i zamówienie ma swoje atrybuty, takie jak nazwa czy cena. Zestaw atrybutów opisujących daną kategorię obiektów nazywamy encją.

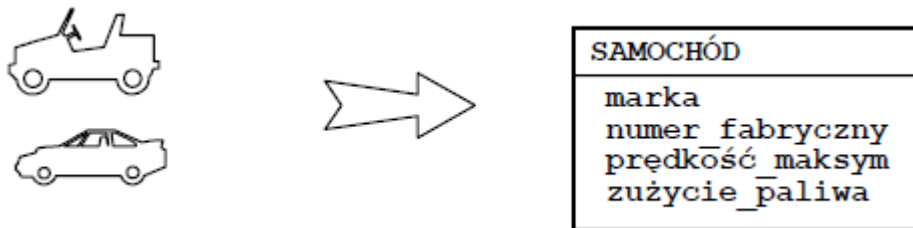
1.3. Definicja encji

Encja to pojęcie stosowane w inżynierii oprogramowania, które reprezentuje wyobrażone lub rzeczywiste obiekty lub grupy obiektów. Przykłady encji to osoby, pojazdy, konta bankowe, konferencje czy wypożyczenia książek. Każda encja jest opisywana przez swoje atrybuty, czyli cechy mające znaczenie dla danej encji, np. imię, nazwisko, numer karty bibliotecznej. Encje mogą być zarówno obiektami fizycznymi, jak i niematerialnymi, takimi jak pojęcia czy wydarzenia. W modelowaniu danych encja jest reprezentowana przez tabelę, a jej wystąpienie przez wiersz w tabeli.

1.4. Przykłady encji

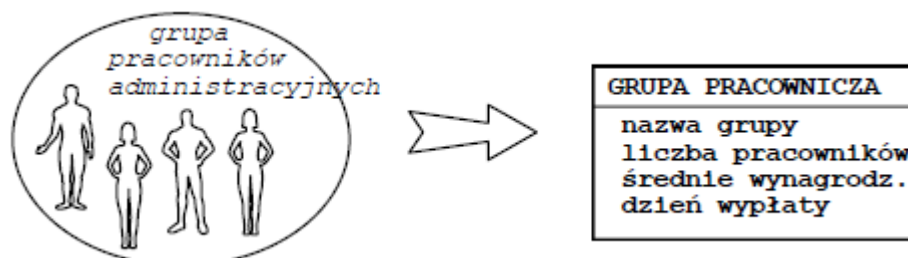
Obiekty materialne

- pracownik, samochód, budynek, produkt, itp



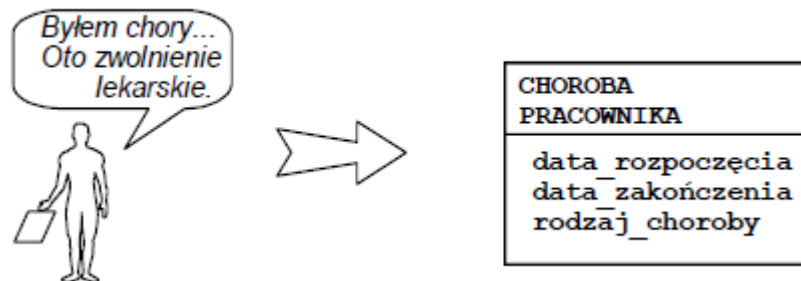
Obiekty niematerialne

- konto bankowe, zamówienie, grupa pracownicza, itp



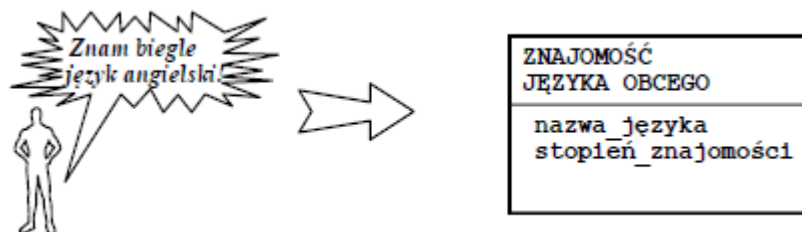
Zdarzenia

- choroba pracownika, przyznanie nagrody, itp.



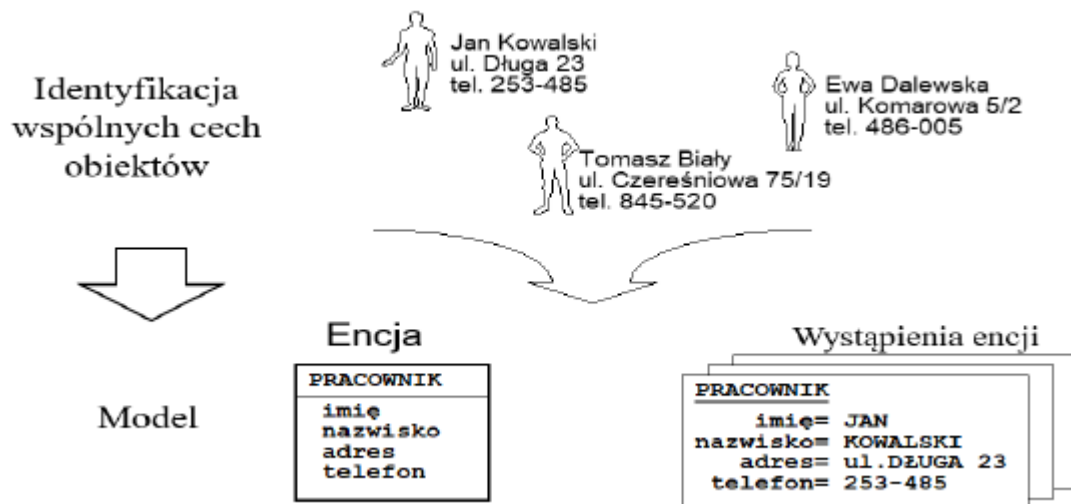
Fakty

- znajomość języka obcego, stan magazynowy produktu, itp.



Przykład #1 (firmowa ewidencja kadrowa)

Firma zatrudnia pracowników, dla których powinna znać dane personalne każdego pracownika, np. imię, nazwisko, adres zamieszkania i numer telefonu.



Przykład #2 (ewidencja towarów firmy handlowej)

Przykładowy opis towarów sprzedawanych w sklepie z oponami samochodowymi jest pokazany na poniższym rysunku.

Numer	Producent	Nazwa	Rozmiar	Cena
1	Michelin	Alpin A3	175/65 R14	315,00
2	Michelin	Alpin A3	195/60 R15	416,00
3	Kleber	Krisalp Hp	195/65 R15	315,00
4	Pirelli	W190 Snowsport	195/60 R15	392,84
5	Uniroyal	MS Plus 55	215/65 R15	424,56

Towar
numer
producent
nazwa
rozmiar
cena

2. Modelowanie

2.1. Modelowanie danych

Model danych jest graficzną reprezentacją rozwiązania problemu informatycznego.

2.2. Diagram związków encji DZE (Entity Relationship Diagrams ERD)

Jest to diagram logiczny reprezentujący strukturę bazy danych z zastosowaniem modelu relacyjnego.

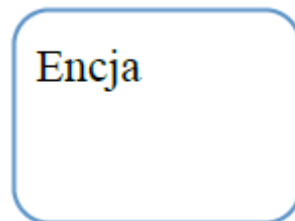
2.3. Ogólne zasady ERD

1. Należy zebrać wszystkie wymagane (mające znaczenie) informacje
2. Informacje występują tylko raz
3. Nie modeluje się informacji możliwych do wyprowadzenia z innych informacji już zamodelowanych.
4. Informacje umieszcza się w przewidywalnym i logicznym miejscu.

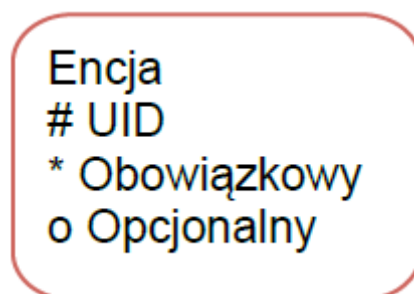
2.4. Konstrukcje ERD

Modelowanie ERD (Entity-Relationship Diagram) to proces projektowania relacyjnych baz danych za pomocą graficznych symboli. Podczas modelowania ERD stosuje się cztery główne konstrukcje: encję, atrybut, związek oraz klucz podstawowy.

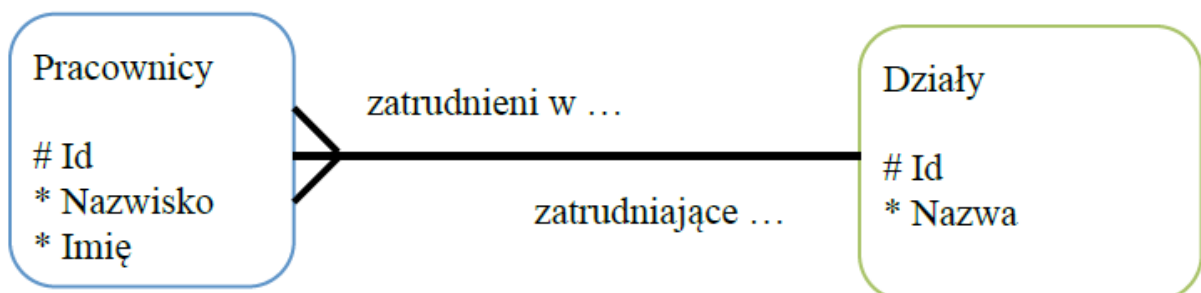
Encja to obiekt informacyjny, który ma znaczenie dla projektantów, użytkowników i projektowanego systemu informatycznego. Encja składa się z wielu wystąpień i po przekształceniu w bazę danych tworzy tabelę (relację), a jej wystąpienie to wiersz (rekord) tabeli.



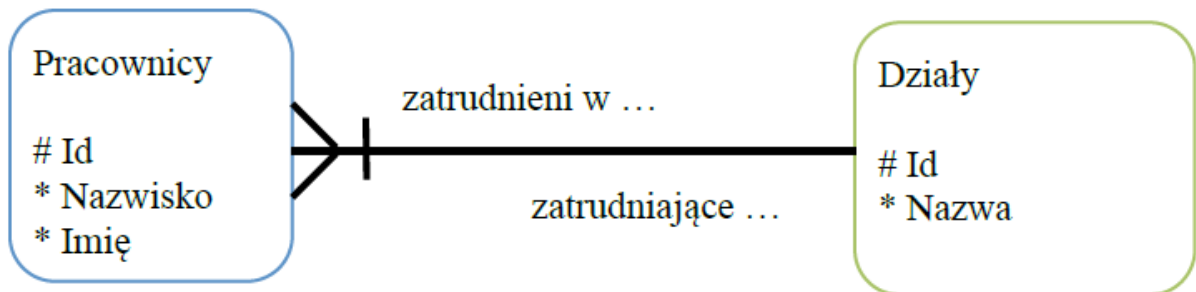
Atrybut to właściwość (cecha) informacyjna encji, która wskazuje istotne dla wszystkich jej wystąpień informacje. Atrybut może być unikalnym identyfikatorem (UID), obligatoryjnym (mandatory) lub opcjonalnym (optional). Podczas przekształcania encji w bazę danych atrybuty stają się kolumnami tabeli.



Związek to powiązanie lub zależność między dwoma encjami. Związek jest implementowany w bazie danych za pomocą kluczy obcych (foreign keys). Na modelu ERD związek jest przedstawiany w postaci linii prostej, a jego rodzaj jest określany przez symbol (np. 1:1, 1:M).



Klucz podstawowy (KP) to unikalny identyfikator (UID), który jednoznacznie identyfikuje każde wystąpienie encji w bazie danych. Klucz podstawowy jest implementowany w bazie danych jako klucz podstawowy (PK) i nie może być pusty (typu null).



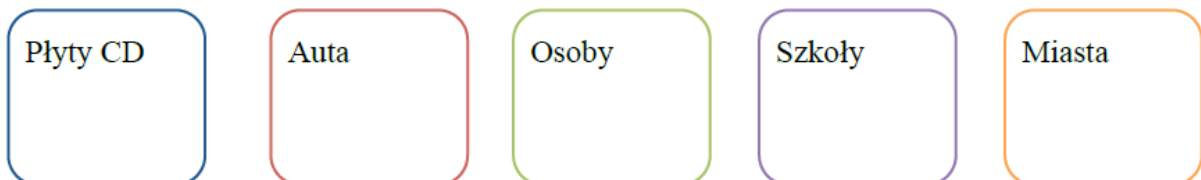
Modelowanie ERD pozwala na stworzenie wizualnej reprezentacji projektowanej bazy danych, która ułatwia zrozumienie relacji między encjami, atrybutami i związkami.

2.5. Graficzne modelowanie encji

Podczas rysowania encji przy użyciu notacji Barkera należy stosować następujące zasady:

1. Encja przedstawiana jest jako prostokąt z zaokrąglonymi narożnikami.
2. Encja musi mieć nazwę i nazwa musi być umieszczona wewnątrz kształtu encji, w górnej jego części.
3. Nazwa encji powinna być pisana dużymi literami.
4. Nazwa encji powinna być w liczbie mnogiej.

Przykład:

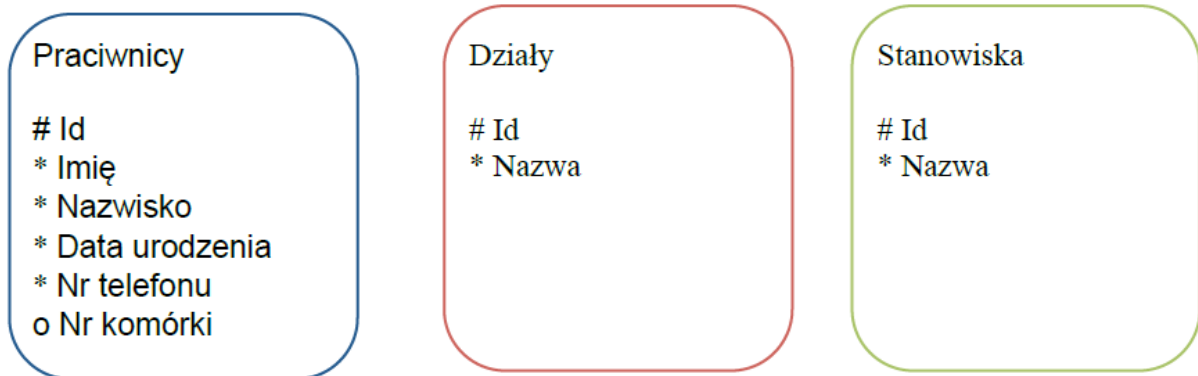


2.6. Graficzne modelowanie atrybutów

Podczas rysowania atrybutów przy użyciu notacja Barkera należy stosować następujące zasady:

1. Atrybuty powinny być opisane tak, aby wszystkie osoby – a nie tylko informatycy – mogły je zrozumieć.
2. Nazwa atrybutu powinna być napisana od wielkiej litery.
3. Obok nazwy każdego atrybutu, z jej lewej strony, powinien być umieszczony symbol wskazany powyżej i określający jego typ (#, *, o).

Przykład:



2.7. Graficzne modelowanie związków

2.7.1. Modalność (połówki) związku – kształty prezentacji

Rodzaje danej cechy:

1. Obowiązkowość: określa, że każde pojedyncze wystąpienie danej encji (czyli tej, z której „wychodzi” dana perspektywa powiązania) musi być związane z odpowiednią liczbą (definiowana przez kardynalność) wystąpień drugiej encji.

Graficznie przedstawiany to jako odcinek linii ciągłej.



2. Opcjonalność: określa, że każde pojedyncze wystąpienie danej encji może być związane z odpowiednią liczbą (definiowana przez kardynalność) wystąpień drugiej encji.

Graficznie przedstawiany to odcinek linii przerywanej.



2.7.2. Kardynalność (połówki) powiązania – kształty prezentacji

Rodzaje danej cechy:

W modelowaniu diagramów relacyjnych występują dwa rodzaje kardynalności: jeden i wiele.

1. Jeden:

Kardynalność jeden dotyczy encji, która może wystąpić tylko raz lub wcale.

Wizualizacja kardynalności jednego to odcinek linii ciągłej.



2. Wiele:

Kardynalność wiele odnosi się do encji, która może wystąpić wiele razy lub raz.

Wizualizacja kardynalności wielu to krótka łapka lub stopa.



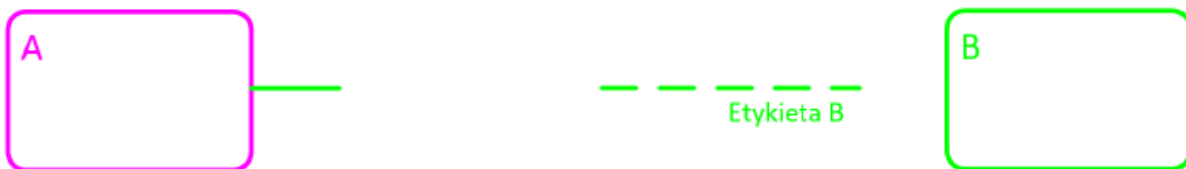
2.7.3. Perspektywy powiązania

W modelowaniu diagramów relacyjnych powiązanie między dwiema encjami jest przedstawiane jako dwie perspektywy: pierwsza i druga.

Pierwsza perspektywa ($A \rightarrow B$) mówi o tym, że każde wystąpienie encji A musi być połączone z jednym lub wieloma wystąpieniami encji B:



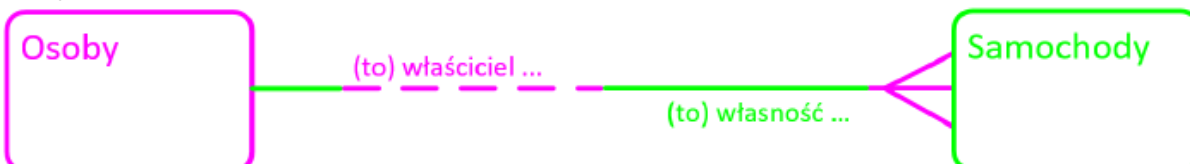
Druga perspektywa ($B \rightarrow A$) mówi o tym, że każde wystąpienie encji B może być połączone z zero lub jednym wystąpieniem encji A:



Powiązanie jest sumą dwóch perspektyw:



Przykład:



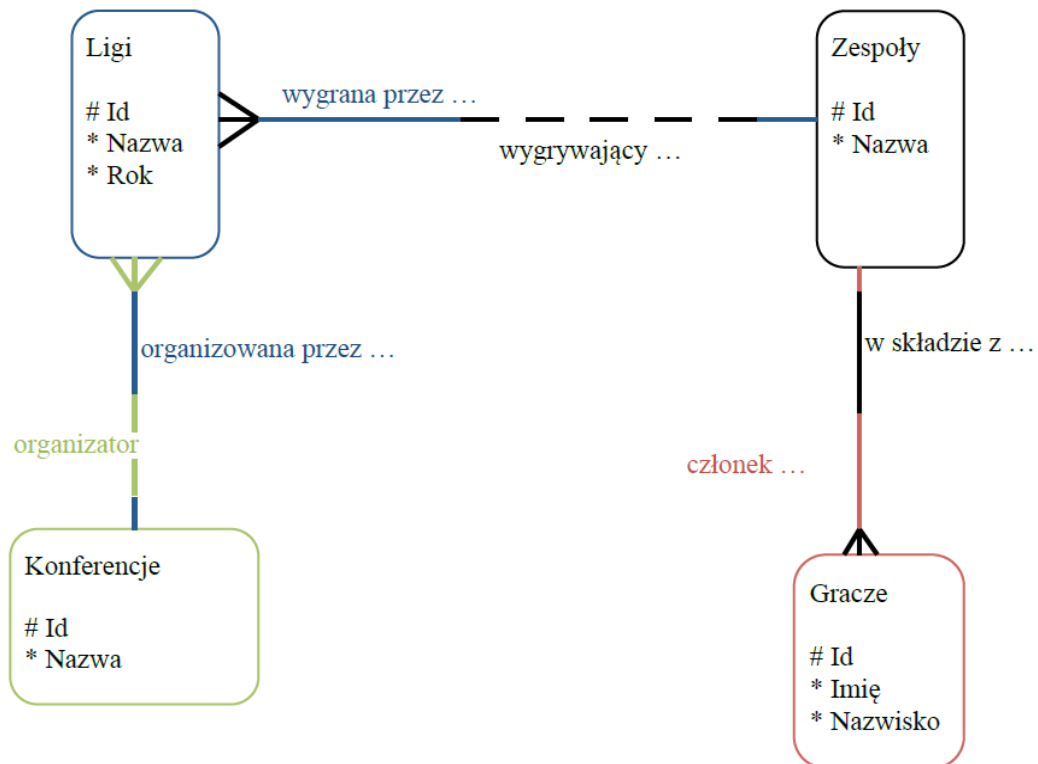
Na przykład, powiązanie między encjami OSOBY i SAMOCHODY przedstawiają dwie perspektywy:

Pierwsza perspektywa (OSOBY → SAMOCHODY) mówi, że każde wystąpienie encji OSOBY może być przypisane jako właściciel zera lub jednego lub wielu wystąpień encji SAMOCHODY

Druga perspektywa (SAMOCHODY → OSOBY) mówi, że każde wystąpienie encji SAMOCHODY musi być przypisane jako własność dokładnie jednego wystąpienia encji OSOBY.

2.7.4. Etykiety (perspektyw) powiązania

Aby utworzyć etykiety dla perspektyw powiązań, należy przestrzegać kilku zasad. Po pierwsze, etykiety powinny być stosowane dla każdej perspektywy. Po drugie, powinny być zapisane słownie i wyrażać rzeczownikową formę, a w przypadku języka angielskiego zaleca się unikanie czasowników i wyrażen czasownikowych na rzecz przysłówkowych. W języku polskim, ze względu na popularność stosowania czasowników i wyrażen czasownikowych do modelowania encji, zaleca się stosowanie rzeczowników. Trzecią zasadą jest dobór odpowiedniej etykiety dla scenariusza projektowanego systemu.




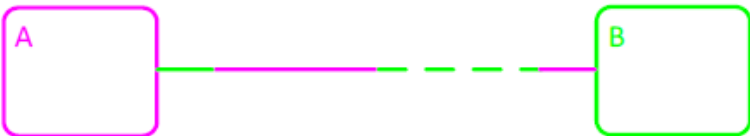


2.7.5. Rodzaje związków

1. Jeden-do-jeden (1:1)

Jeden-do-jeden (1:1) to rodzaj powiązania, w którym każde pojedyncze wystąpienie jednej encji jest powiązane maksymalnie z jednym wystąpieniem drugiej encji, w zależności od modalności perspektywy.

Poniższa tabela przedstawia cztery możliwe kombinacje związane z modalnością, wraz z minimalną i maksymalną liczbą wystąpień encji dla poszczególnych perspektyw.

Laboratorium z przedmiotu: **Systemy Informatyczne**
Studia I stopnia, tryb stacjonarny i niestacjonarny





Nr	Rodzaj związku	Kard. min.	Kard. maks.
1		$A \rightarrow B$ 0 $B \rightarrow A$ 0	$A \rightarrow B$ 1 $B \rightarrow A$ 1
2		$A \rightarrow B$ 1 $B \rightarrow A$ 0	$A \rightarrow B$ 1 $B \rightarrow A$ 1
3		$A \rightarrow B$ 0 $B \rightarrow A$ 1	$A \rightarrow B$ 1 $B \rightarrow A$ 1
4		$A \rightarrow B$ 1 $B \rightarrow A$ 1	$A \rightarrow B$ 1 $B \rightarrow A$ 1

2. Jeden-do-wielu (1:M)

W zależności od modalności perspektywy, każda instancja jednej encji może być powiązana z maksymalnie wieloma instancjami drugiej encji.

Poniższa tabela przedstawia cztery możliwe kombinacje związane z modalnością, a w dwóch ostatnich kolumnach znajduje się informacja o minimalnej i maksymalnej liczbie wystąpień encji dla poszczególnych perspektyw.

Laboratorium z przedmiotu: **Systemy Informatyczne**
Studia I stopnia, tryb stacjonarny i niestacjonarny

#	Rodzaj związku	Kard. min.	Kard. maks.
1		$A \rightarrow B$ 0 $B \rightarrow A$ 0	$A \rightarrow B$ n $B \rightarrow A$ 1
2		$A \rightarrow B$ 1 $B \rightarrow A$ 0	$A \rightarrow B$ n $B \rightarrow A$ 1
3		$A \rightarrow B$ 0 $B \rightarrow A$ 1	$A \rightarrow B$ n $B \rightarrow A$ 1
4		$A \rightarrow B$ 1 $B \rightarrow A$ 1	$A \rightarrow B$ n $B \rightarrow A$ 1

3. Wiele-do-wielu (M:M)

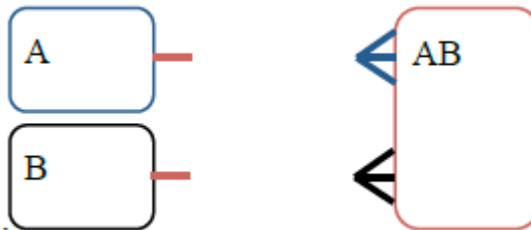
W przypadku tego rodzaju powiązania, które nie może być zrealizowane bezpośrednio w bazie danych relacyjnej, konieczne jest zastosowanie dodatkowej encji oraz dedykowanych powiązań. Rozwiązanie to może być realizowane ręcznie lub automatycznie, jeśli używane oprogramowanie do modelowania (np. SDDM) oferuje taką funkcjonalność.

Podczas rozwiązywania powiązania wiele do wielu dla przykładowych encji A i B, należy wykonać następujące kroki:

1. Tworzymy trzecią encję (pośrednią/haszującą) AB

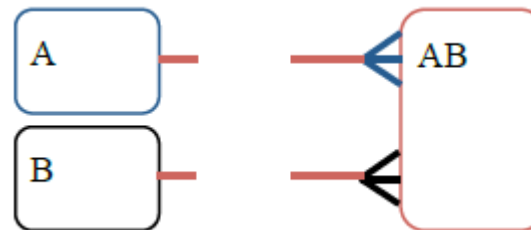


2. Dodajemy dwa powiązania jeden-do-wielu z oryginalnych encji do nowej encji.

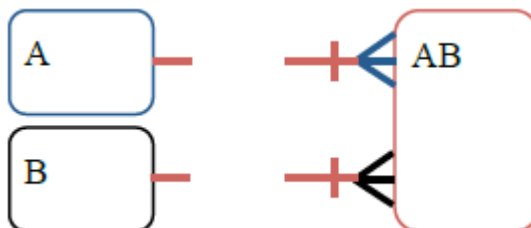


3. Modalność z perspektywy od encji AB powinna być zawsze obowiązkowa.

4. Modalność pozostałych perspektyw jest dziedziczona z modalności źródłowego powiązania wielu do wielu.



5. Paski Uld są dodawane do perspektyw $AB \rightarrow A$ i $AB \rightarrow B$ (wskazują, że klucze obce są również kluczami podstawowymi - patrz następny rozdział Pasek UID).



Przykład:

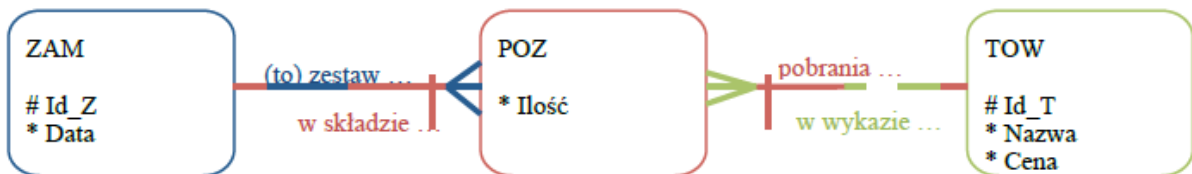
Baza danych przechowuje listę zamówień dziennych. Jedna tabela ZAM jest potrzebna do codziennych zamówień, a inna do różnych dostępnych

przedmiotów TOW. Trzecia tabela (rozwiązująca związek wiele do wielu) POZ zawiera listę przedmiotów sprzedanych w każdym zamówieniu.



Rozwiązaniem jest trzecia tabela (rozwiązująca powiązanie wiele do wielu) POZ

zawiera listę przedmiotów sprzedanych w każdym zamówieniu



Zestawienie rozwiązań:

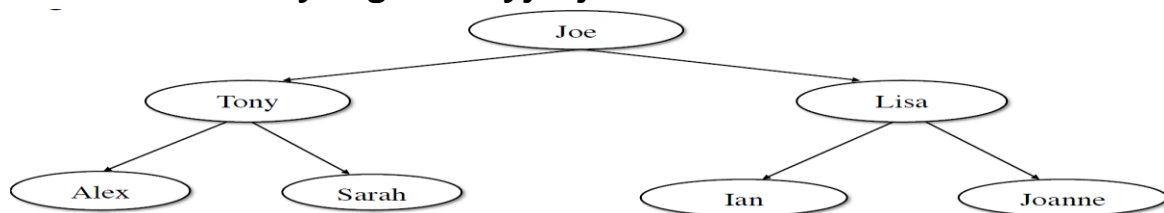
#	Rodzaj związku
1	
2	
3	
4	

2.8. Związek rekursywny

W niektórych sytuacjach związek występuje między tymi samymi encjami, co jest znane jako powiązanie rekurencyjne. Jest on zazwyczaj realizowany za pomocą klucza obcego, który odnosi się do klucza podstawowego tej samej encji. Przyjrzyjmy się poniższemu scenariuszowi:

Firma składa się z wielu pracowników. Joe jest przewodniczącym i nie ma menedżera. Joe jest menedżerem Tony'ego i Lisy. Tony jest menadżerem Alexa i Sarah. Lisa jest menadżerem Iana i Joanne.

Schemat struktury organizacyjnej:



Perspektywy:

Każde wystąpienie encji <Pracownicy> może dotyczyć „kierowania” jednym lub wieloma wystąpieniami encji < Pracownicy >.

Każde wystąpienie encji <Pracownicy> może dotyczyć „bycia kierowanym” przez dokładnie jedno wystąpienie encji <Pracownicy>.

Implementacja w bazie:

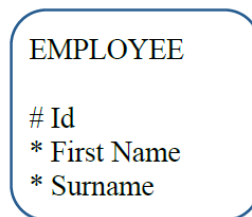
Tabela: Pracownicy		
<u>Id</u>	Imię	Id_kierownika
1	Joe	
2	Tony	1
3	Lisa	1
4	Alex	2
5	Sarah	2
6	Ian	3
7	Joanne	3

2.9. Atrybuty UID

Istnieją cztery typy atrybutów Unique Identifier, które są implementowane jako klucze podstawowe:

1. **Pojedynczy atrybut UID:** Pojedynczy atrybut UID występuje, gdy encja zawiera tylko jeden atrybut UID, który nie jest kluczem obcym.

ERD:

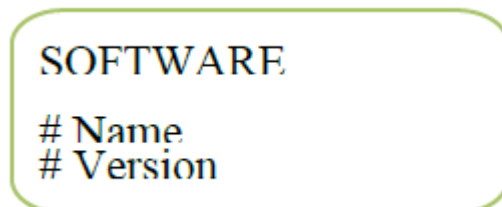


Implementacja w bazie:

Table: Employees		
<u>Id</u>	First Name	Surname
1	Joe	Borg
2	Tony	Galea
3	Lisa	Abela

2. **Wielokrotny atrybut UID:** wielokrotny atrybut UID występuje, gdy klucz podstawowy składa się z wielu atrybutów (złożonego klucza podstawowego), z których wszystkie są kluczami nie będącymi kluczami obcymi.

ERD:

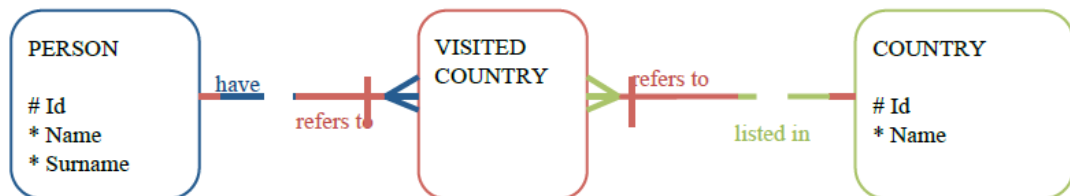


Implementacja w bazie:

Table: Software	
<u>Name</u>	<u>Version</u>
Oracle DBMS	11g
MySQL	5.1
SQL Server	2008

3. **Skomponowany atrybut UID:** Skomponowany atrybut UID występuje, gdy encja ma klucz podstawowy, który jest również kluczem obcym. Są one oznaczone za pomocą paska UID.

ERD:



Implementacja w bazie:

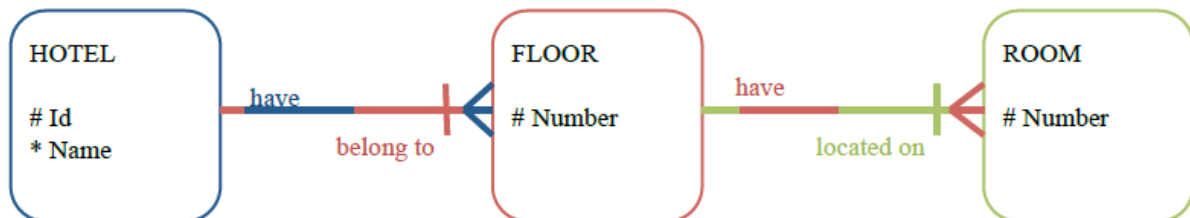
Table: Persons		
<u>Id</u>	Name	Surname
1	Joe	Borg
2	Tony	Galea
3	Lisa	Abela

Table: Visited Countries	
<u>Person ID</u>	<u>Country ID</u>
1	1
1	2
1	3
2	4
3	4

Table: Countries	
<u>Id</u>	Name
1	Italy
2	England
3	France
4	Switzerland

4. **Skomponowany kaskadowy atrybut UID:** Skomponowany kaskadowy atrybut UID występuje wtedy, gdy encja używa kluczy obcych będących kluczami podstawowymi encji ze skomponowanymi atrybutami UID.

ERD:



Implementacja w bazie:

Table: Hotel	
<u>Id</u>	Name
1	Hilton
2	Westin

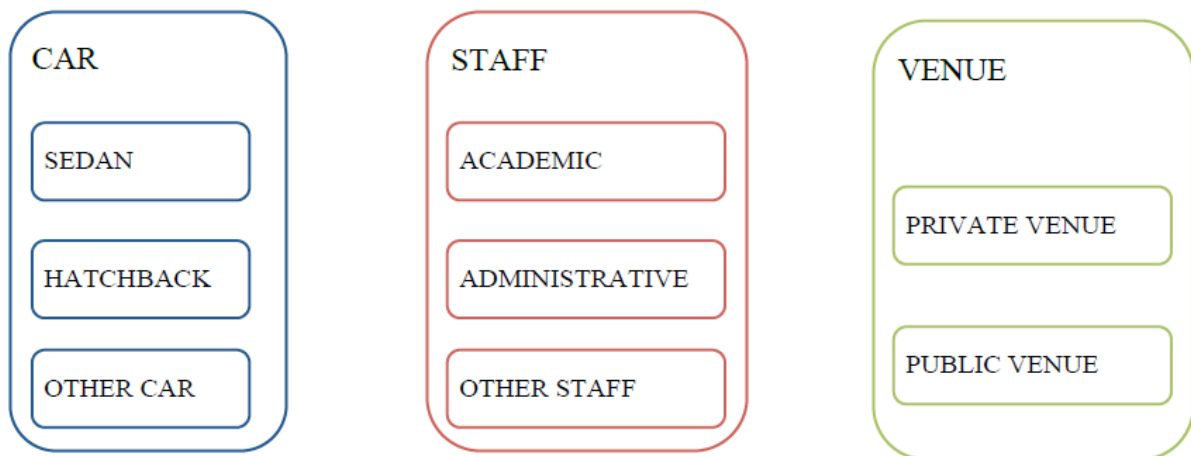
Table: Floor	
<u>Hotel ID</u>	<u>Floor Number</u>
1	1
2	1
2	2

Table: Room		
<u>Hotel Id</u>	<u>Floor Number</u>	<u>Room Number</u>
1	1	1
1	1	2
2	1	1
2	1	2
2	2	1

2.10. Nad typy i podtypy encji

Zdarzają się sytuacje, w których różne encje mogą być zgrupowane w jedną wspólną encję nazywaną supertypem, podczas gdy zagnieżdżone encje są podtypami. Używa się też określeń nadtyp lub generalizacja dla supertypu, oraz określenie specjalizacja dla podtypy encji.

Przykład:



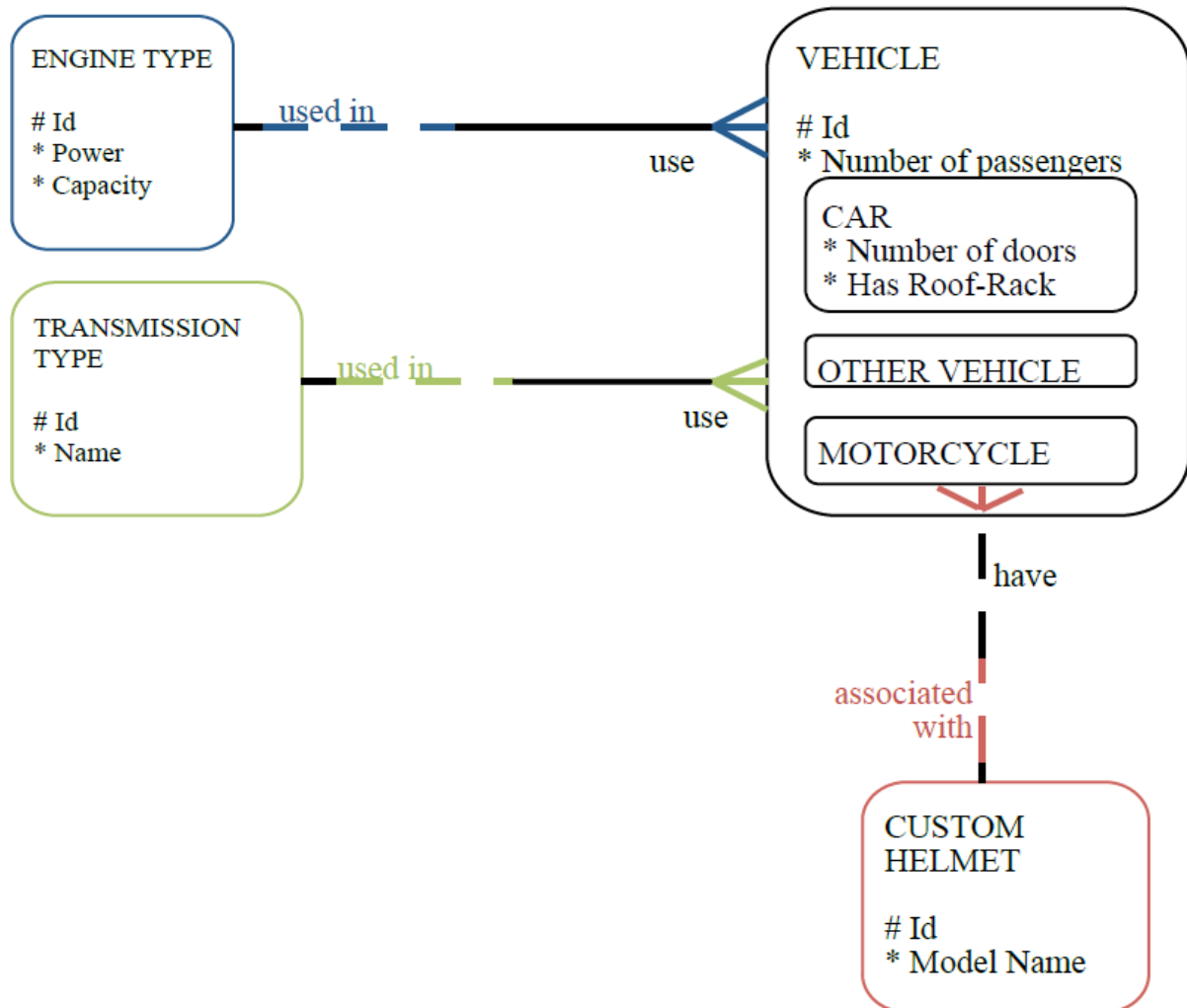
2.10.1. Zasady podtypów encji

1. **Zasada wyczerpania:** każda instancja (wystąpienie) encji supertypu musi być instancją jednego z podtypów.
2. **Zasada wzajemnego wykluczenia:** każda instancja encji supertypu może być instancją tylko jednej instancji podtypu.

Podczas rysowania podtypów encji należy pamiętać o następujących zasadach:

1. Każdy podtyp jest specjalizacją supertypu i dlatego musi być zamknięty w encji supertypu.
2. Typowe atrybuty i powiązania dla wszystkich podtypów muszą być wymienione tylko w supertypie, ale są dziedziczone w każdym podtypie.
3. Podtyp może i ogólnie ma własne atrybuty i powiązania z innymi encjami.
4. Nigdy nie może istnieć tylko jeden podtyp dla danej encji supertypu.

Przykład:

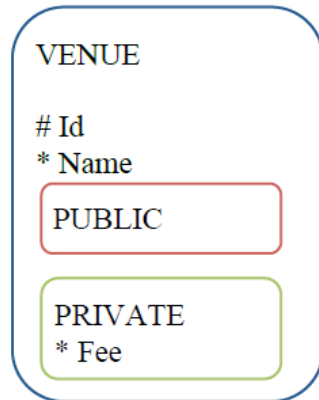


2.10.2. Implementacja podtypów

Istnieją różne metody implementacji podtypu, ale preferowana metoda jest następująca:

1. Tworzymy tabelę dla supertypu ze wszystkimi typowymi atrybutami i powiązaniem.
2. Tworzymy oddzielną tabelę dla każdego podtypu.
3. W podtypach znajdują się określone atrybuty i powiązania.
4. Tworzymy klucz obcy w supertypie dla każdego podtypu.
5. Nakładamy ograniczenia, że tylko jeden klucz obcy nie może mieć wartości pustej (typu NULL).
6. Nakładamy, że przynajmniej jeden klucz obcy nie może mieć wartości pustej (typu NULL).

ERD:



Implementacja w bazie:

Table: Venues			
<u>Vid</u>	Name	Pub_id	Prv_id
1	Floriana Granaries	1	
2	Villa Arrigo		1
3	MFCC		2

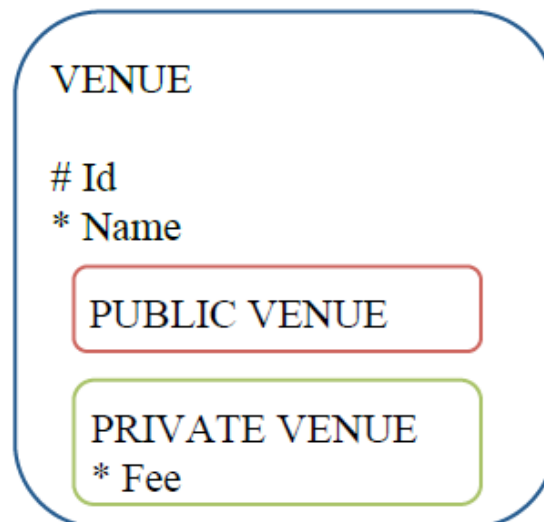
Table: Public Venues	
<u>Pub_id</u>	
1	

Table: Private Venues	
<u>Prv_id</u>	Fee
1	500
2	600

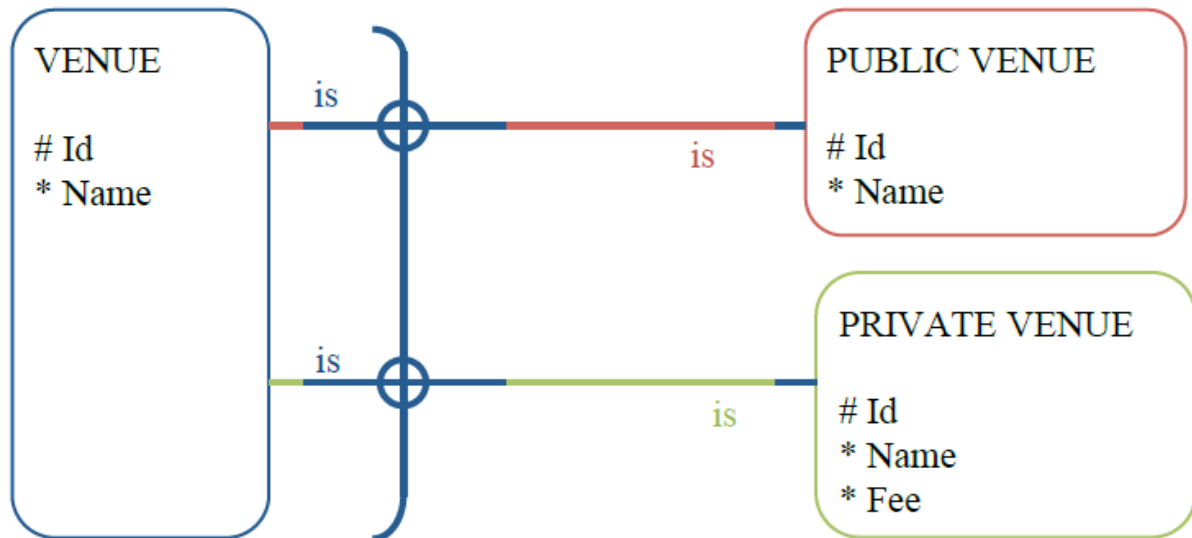
2.11. Wyłączające powiązanie Łuk (Arc)

Mogą istnieć sytuacje, w których encja jest powiązana z jedną lub inną encją, ale nie z obydwooma. Ilekroć konieczne jest dokonanie selekcji między powiązaniami, tworzony jest wyłączający łuk.

Przywołajmy scenariusz podtypów:



Powyższa encja z odpowiednimi podtypami może zasadniczo być reprezentowana w następujący sposób:

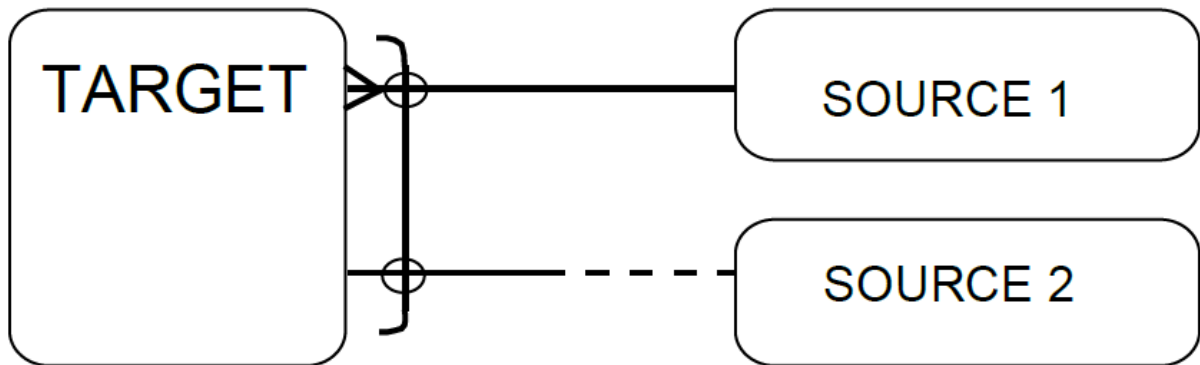


- Każde pojedyncze wystąpienie encji <VENUE> musi być dokładnie jednym wystąpieniem encji <PUBLIC VENUE> lub <PRIVATE VENUE>.
- Każde pojedyncze wystąpienie encji < PUBLIC VENUE> musi być dokładnie jednym wystąpieniem encji <VENUE>.
- Każde pojedyncze wystąpienie encji < PRIVATE VENUE> musi być dokładnie jednym wystąpieniem encji <VENUE>.

Przykład:

Baza danych jest używana do przechowywania listy pasów startowych (encja <RUNWAY>). Pas startowy może być własnością lotniska (encja <AIRPORT>), takiego jak Międzynarodowy Port Lotniczy Malta. Lotnisko może posiadać co najmniej jeden pas startowy, ale może posiadać więcej. Oprócz tego pas startowy może być własnością klubu (encja <CLUB>), takiego jak klub Motor Racing, który jest właścicielem pasa startowego w Ħal-Far, lub innego klubu Remote Control Club, który jest właścicielem pasa startowego w Ta 'Qali. Załóżmy, że prawo w naszym kraju nakłada na klub obowiązek posiadania tylko jednego pasa startowego. Ponadto prawo nie zezwala na to, by były bezpieczeństwa pasy startowe.

2.11.1. Zasady wyłączającego powiązania Łuk (Arc)



Podczas rysowania łuku powiązania wyłączności należy zastosować następujące reguły:

1. Łuk powiązania może być zastosowany tylko do jednej encji, zwanej encją docelową.
2. Łuk powiązania musi być zastosowany do minimalnej liczby dwóch powiązań wychodzących z encji docelowej.
3. Encja docelowa będzie zawierała klucze obce powiązań zależnych od łuku.
4. Modalność powiązania objętego łukiem musi być taka sama dla każdej z perspektyw encji docelowej.
5. Modalność powiązania objętej łukiem może dla perspektyw powiązań pozostałych encji być inna niż dla ww. modalności..
6. Powiązania objęte łukiem mogą mieć różne kardynalności (liczności) – jeden-do-jeden, jeden-do-wielu.

2.11.2. Rysowanie wyłączającego powiązania Łuk (Arc)

Aby zamodelować wyłączające powiązanie Łuk, należy wykonać następujące kroki:

1. Określamy właściciela powiązania – encję docelową.
2. Tworzymy łuk obejmujący powiązania wychodzące z encji właściciela.
3. Zakreślamy kółkiem skrzyżowane linie (powiązania i łuku) – lub pozostawiamy to narzędziu do modelowania

2.11.3. Implementowanie wyłączającego powiązania Łuk (Arc)

Aby zaimplementować takie powiązanie, należy wykonać następujące kroki:

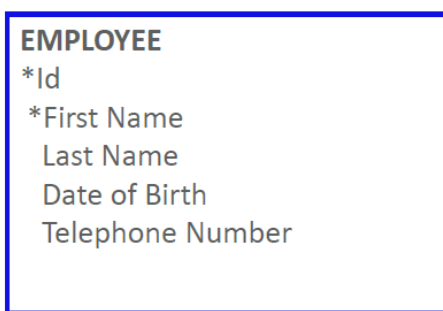
1. Tworzymy klucz obcy dla każdego powiązania, na który ma wpływ łuk. Obce klucze muszą znajdować się w encji, do którego należy łuk.
2. Implementujemy ograniczenie, które pozwala, aby tylko jedna wartość klucza obcego nie była pusta (typu NULL).

3. Inne notacje

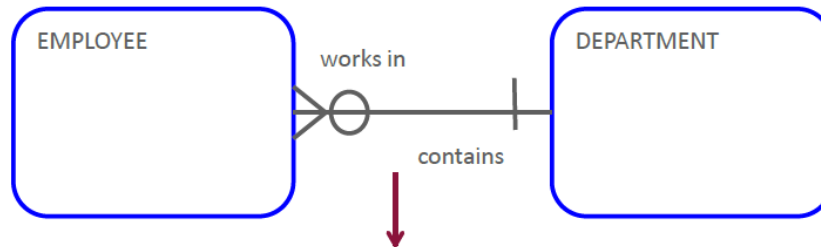
Firma Oracle, np. w programie SQL Developer Data Modeler umożliwia stosowanie dwóch innych notacji: Bachmana i Information Engineering.

Bachman Notation

- Entity (represented by a box)
- Attributes
- Relationship lines



Information Engineering Notation



An EMPLOYEE works only in one DEPARTMENT.
A DEPARTMENT contains zero or more EMPLOYEES.

zero or more ———— ○⋈

zero or one ———— ○⊥

one or more ———— ⊥⋈

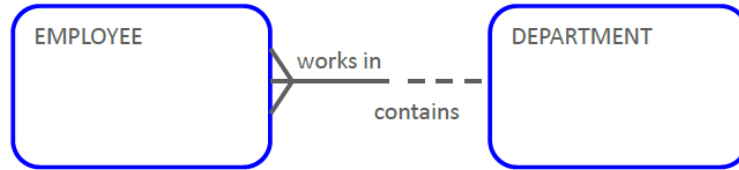
only one ———— ⊥⊥

Data Model Notations

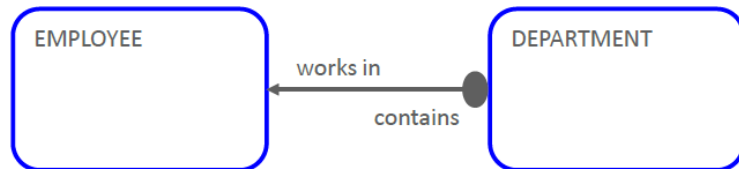
Notation	Barker Notation	Bachman Notation	Information Engineering
Zero or one	——— - □		——— ○⊥ □
One only	——— □	● ——— ●	——— ⊥⊥ □
Zero or more	——— ▷ □	○ ——— → ○	——— ○⋈ □
One or more	——— ▷ □	● ——— → ●	——— ⊥⋈ □
Primary Key/Unique key	#	P	

Data Model Notations: Examples

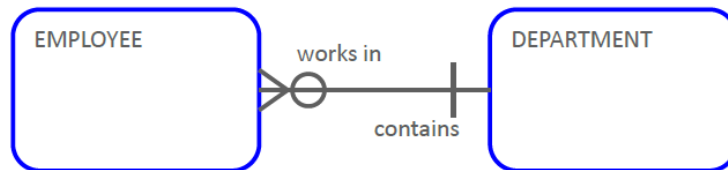
Barker
Notation



Bachman
Notation



Information
Engineering
Notation



Laboratorium z przedmiotu: **Systemy Informatyczne**
Studia I stopnia, tryb stacjonarny i niestacjonarny
