

Wprowadzenie w problematykę baz danych

Zastosowania, terminologia, przegląd
zagadnień

Tradycyjne zastosowanie baz danych

- » Kluczowe narzędzie współczesnego społeczeństwa, np.:
 - wpłacanie/wypłacanie pieniędzy w banku
 - rezerwacja miejsc w hotelu/biletów lotniczych
 - zakupy w sklepie internetowym
 - zakupy w supermarkecie
- » Dane tekstowe lub numeryczne

Nowe zastosowania baz danych

- » Systemy składowania *Big data/Systemy NoSQL*
 - serwisy społecznościowe FB, Twitter...
 - zarządzanie danymi w chmurze
- » Multimedialne bazy danych *YouTube, Spotify*
 - cyfrowe zapisywanie fotografii, wideo, audio
- » Geograficzne systemy informacyjne (GIS) *Google Maps*
 - przechowywanie i analiza map, zdjęć pogodowych, satelitarnych
- » Hurtownie danych. systemy OLAP
 - analiza danych do podejmowania decyzji

Nowe zastosowania baz danych c.d.

- » Bazy danych czasu rzeczywistego i aktywne bazy danych
 - sterowanie kontrolą i procesami w produkcji przemysłowej
- » Internet rzeczy (IoT)

Bazy danych odgrywają główną rolę we wszystkich obszarach wykorzystywania komputerów

Kluczowe definicje

- » *Baza danych* – zbiór powiązanych danych
 - *dane*: znane fakty, które można rejestrować i które mają konkretne znaczenie
- » *Baza danych* – własności:
 - reprezentuje wybrany aspekt rzeczywistego świata
 - mini-świat, dziedzina problemu – zmiany w nim muszą być uwzględniane w BD
 - logicznie koherentny zbiór danych z jakimś spójnym znaczeniem
 - przypadkowy zbiór danych nie jest BD
 - projektowana, konstruowana i wypełniana danymi w określonym celu
 - z przypisaną grupą docelowych użytkowników i z góry przyjęte zastosowania realizowane przez nich

Rozmiary baz danych

- » Dowolne rozmiary i zróżnicowane poziomy złożoności, np.:
 - lista adresowa – kilkaset rekordów o prostej strukturze
 - skomputeryzowany katalog zbiorów wielkiej biblioteki – ponad pół miliona wpisów zorganizowanych w różnych kategoriach, uporządkowane alfabetycznie
 - baza danych utrzymywana przez serwisy społecznościowe – FB: ponad miliard użytkowników – informacja o powiązaniach, posty, komentarze itd..

Przykładowa baza danych *Amazon.com*

- » Dane na temat ponad 150 mln użytkowników
- » Miliony produktów
- » Zajmuje ponad 42 terabajty
- » Przechowywana na setkach serwerów

Bazy mogą być generowane i utrzymywane ręcznie
lub skomputeryzowane

Kluczowe definicje c.d.

- » *System zarządzania bazami danych* (SZBD, DBMS)
 - zbiór programów, które umożliwiają tworzenie i utrzymywanie bazy danych
 - uniwersalny system programowy, ułatwiający definiowanie, konstruowanie, manipulowanie i udostępnianie baz danych różnym użytkownikom i aplikacjom
- » *Definiowanie bazy danych* = określenie typów i struktur danych oraz ograniczeń
- » *SZBD* – przechowuje definicje bazy oraz *metadane*

Kluczowe definicje c.d.

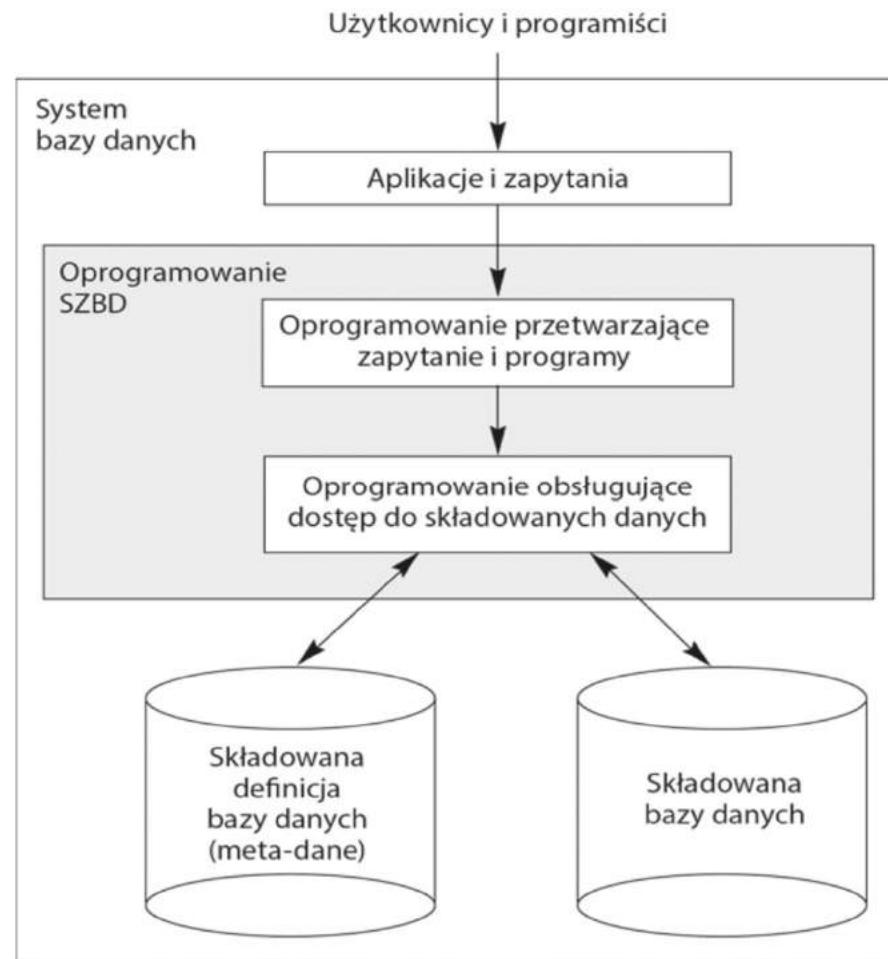
- » *Konstruowanie BD*
 - kontrolowany przez SZBD proces umieszczania w BD właściwych informacji w jakimś repozytorium (medium przechowywania)
- » *Manipulowanie BD*
 - wykonywanie zapytań wyciągających z BD określone informacje, aktualizowanie BD, generowanie raportów
- » *Udostępnianie BD*
 - umożliwia wielu użytkownikom i programom jednoczesne operowanie na danych

Kluczowe definicje c.d.

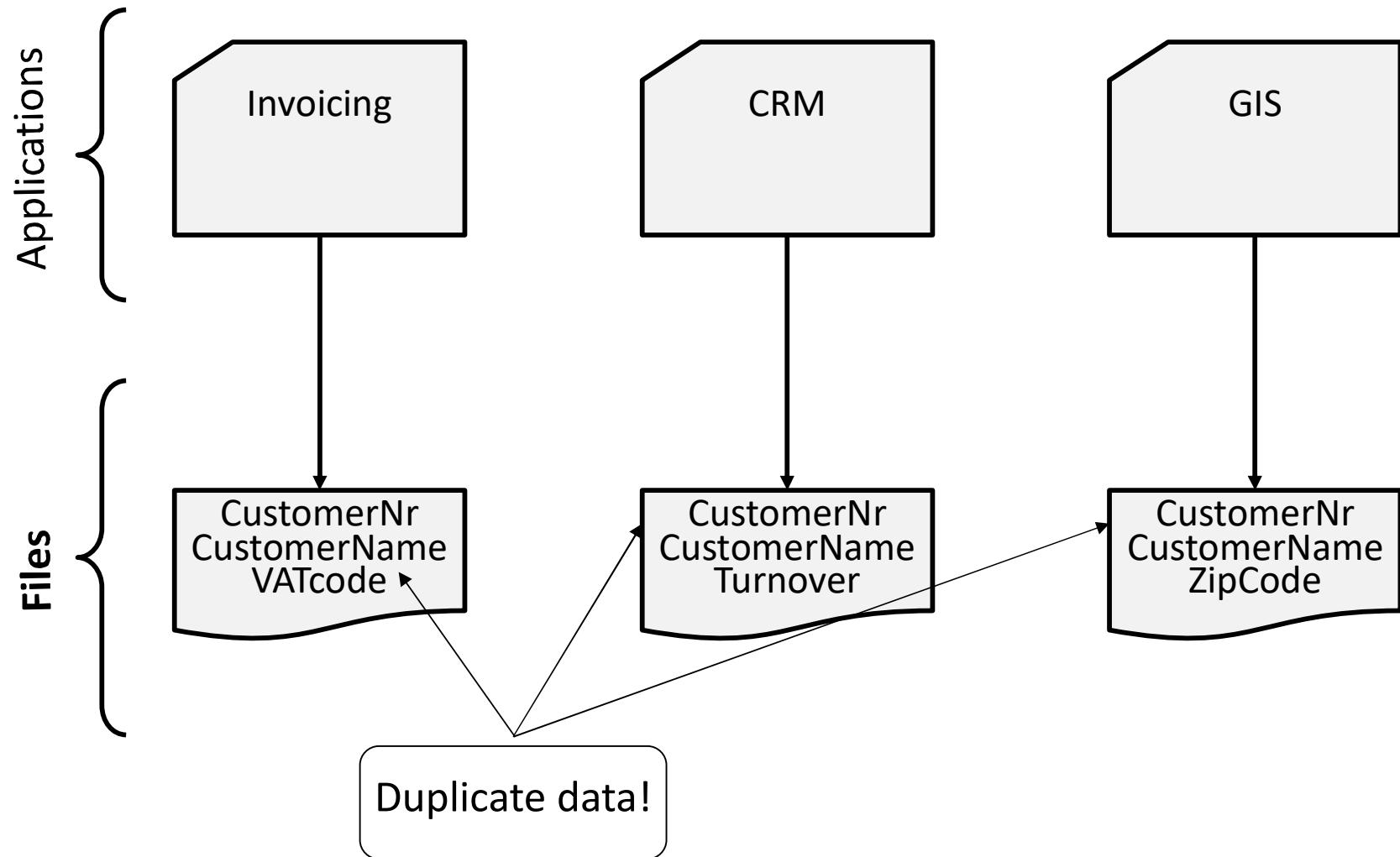
- » Aplikacja uzyskuje *dostęp* do BD wysyłając zapytania lub żądania do SZBD
 - *zapytanie*: pobranie lub modyfikowanie
- » Pozostałe funkcje SZBD:
 - *ochrona BD*
 - *konserwowanie*

System baz danych

» = Baza danych + oprogramowanie SZBD



Podejście: plik vs baza danych do zarządzania danymi





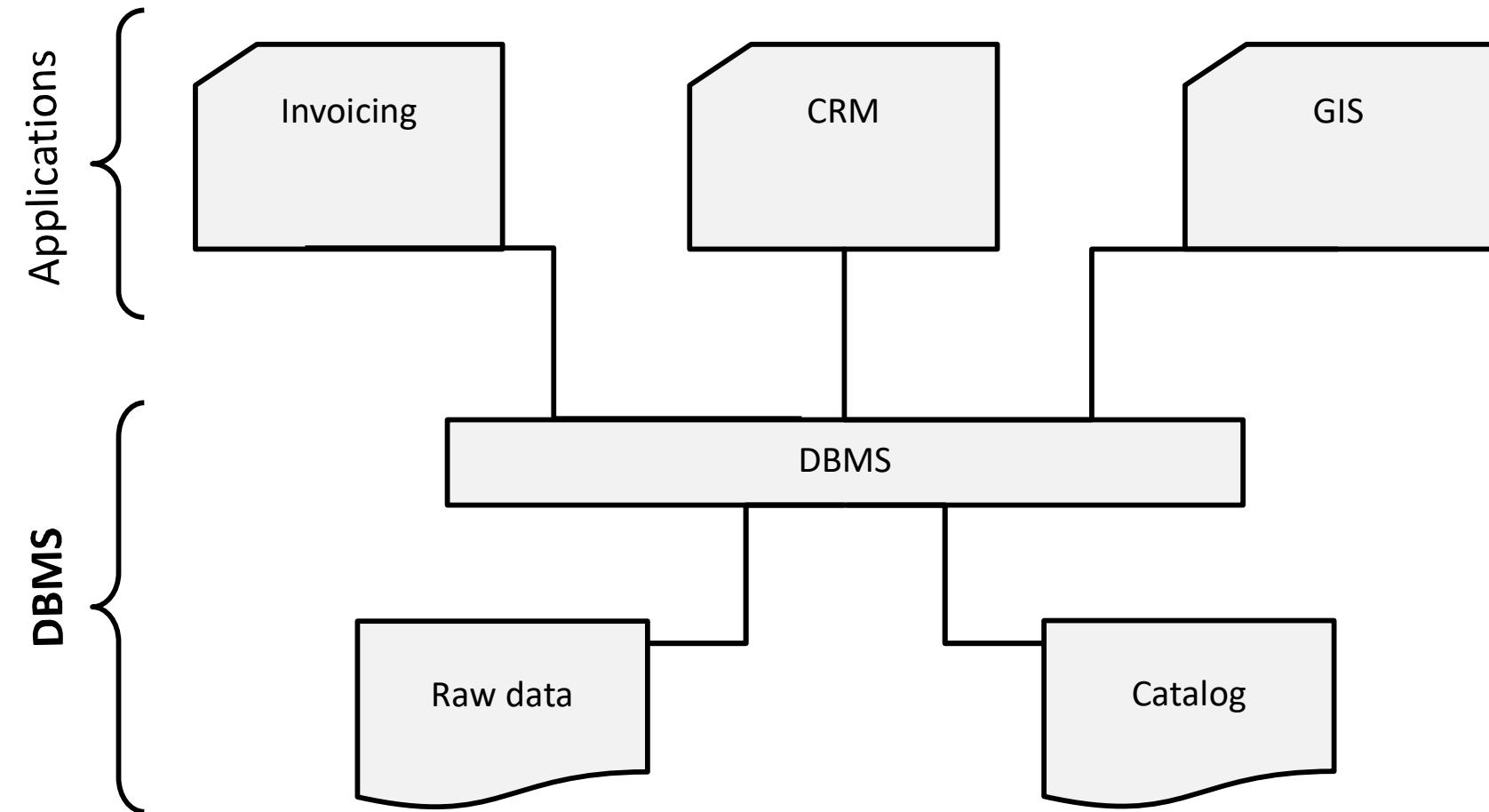
AGH

Podejście: plik vs baza danych do zarządzania danymi c.d.

Początki: aplikacje bazodanowe budowane bezpośrednio na systemach plików, co prowadziło do:

- » Przechowywanie zduplikowanych lub nadmiarowych informacji
- » Niebezpieczeństwo niespójnych danych
- » Silne powiązanie między danymi i aplikacjami
- » Trudno zarządzać kontrolą współbieżności
- » Aplikacje trudne do zintegrowania

Podejście: plik vs baza danych do zarządzania danymi c.d.



Podejście: plik vs baza danych do zarządzania danymi c.d.

» Podejście: plik

```
Procedure FindCustomer;  
begin  
    open file Customer.txt;  
    Read(Customer)  
    While not EOF(Customer)  
        If Customer.name='Bart' then  
            display(Customer);  
        EndIf  
        Read(Customer);  
    EndWhile;  
End;
```

• Podejście: BD (SQL)

```
SELECT *  
FROM Customer  
WHERE  
name = 'Bart'
```

Podejście: plik vs baza danych do zarządzania danymi c.d.

- » *Baza danych:*
 - utrzymywane tylko 1 repozytorium
 - raz utworzone, stosowane przez wielu użytkowników
- » Właściwości podejścia opartego na bazach danych:
 1. samoopisująca natura systemów baz danych
 2. oddzielenie programów od danych oraz abstrakcja danych
 3. obsługa wielu perspektyw dla tych samych danych
 4. współdzielenie danych oraz przetwarzanie transakcji

Ad 1. samoopisująca natura systemów baz danych

- » System bazy danych = baza danych + kompletna definicja lub opis jej struktury i ograniczeń.
 - przechowywana w *katalogu SZBD - metadane*
 - struktura poszczególnych plików, typy i formaty przechowywania poszczególnych elementów danych oraz ograniczeń
 - NoSQL – zamiast metadanych *dane samoopisujące*
 - Katalog wykorzystywany przez SZBD i użytkowników
 - gdy potrzebna informacja odnośnie struktury BD
 - pakiety programowa SZBD – uniwersalne – możliwość obsługi wielu BD
- » Przy przetwarzaniu plików
 - definicja danych - część aplikacji do ich obsługi => programy mogą pracować tylko z 1 BD

Ad 2. oddzielenie programów od danych oraz abstrakcja danych

- » Przy przetwarzaniu plików
 - struktura plików z danymi umieszczana w programach => zmiany struktury wymagają zmiany programów
- » W SZBD
 - struktura plików z danymi przechowywana w katalogu SZBD – oddzielona od programów dostępowych
 - *niezależność programu od danych* – abstrakcja danych

Ad 3. obsługa wielu perspektyw dla tych samych danych

- » Każdy użytkownik BD potrzebuje dostępu do innej perspektywy (view) dla tej BD
 - podzbiór BD albo zawiera dane wirtualne
 - zapewnienie mechanizmów ułatwiających definiowanie różnych perspektyw

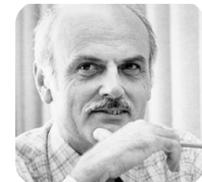
Ad 4. współdzielenie danych oraz przetwarzanie transakcji

- » Oprogramowanie sterowania współbieżnością
 - przetwarzanie transakcji na bieżąco (OLTP)
 - kluczowe pojęcie *transakcji*
 - wykonywany program/proces
 - właściwości: izolacja, niepodzielność

Elementy systemu bazodanowego: Modele danych

- » Zbiór pojęciowych narzędzi do opisu:
 - Danych
 - Zależności między danymi
 - Semantyki danych
 - Ograniczeń dla danych
- » Różnorodność modeli:
 - Relacyjny (Ted Codd, 1970)
 - Związków encji (Entity-Relationship) (głównie do projektowania baz danych)
 - Obiektowy (również obiektowo-relacyjny)
 - Semistrukturalny (XML)
 - Inne starsze modele:
 - Hierarchiczny
 - Sieciowy (raport CODASYL, 1971)

- » Wszystkie dane przechowywane w różnych tabelach
- » Przykład danych tabelarycznych w modelu relacyjnym



Ted Codd

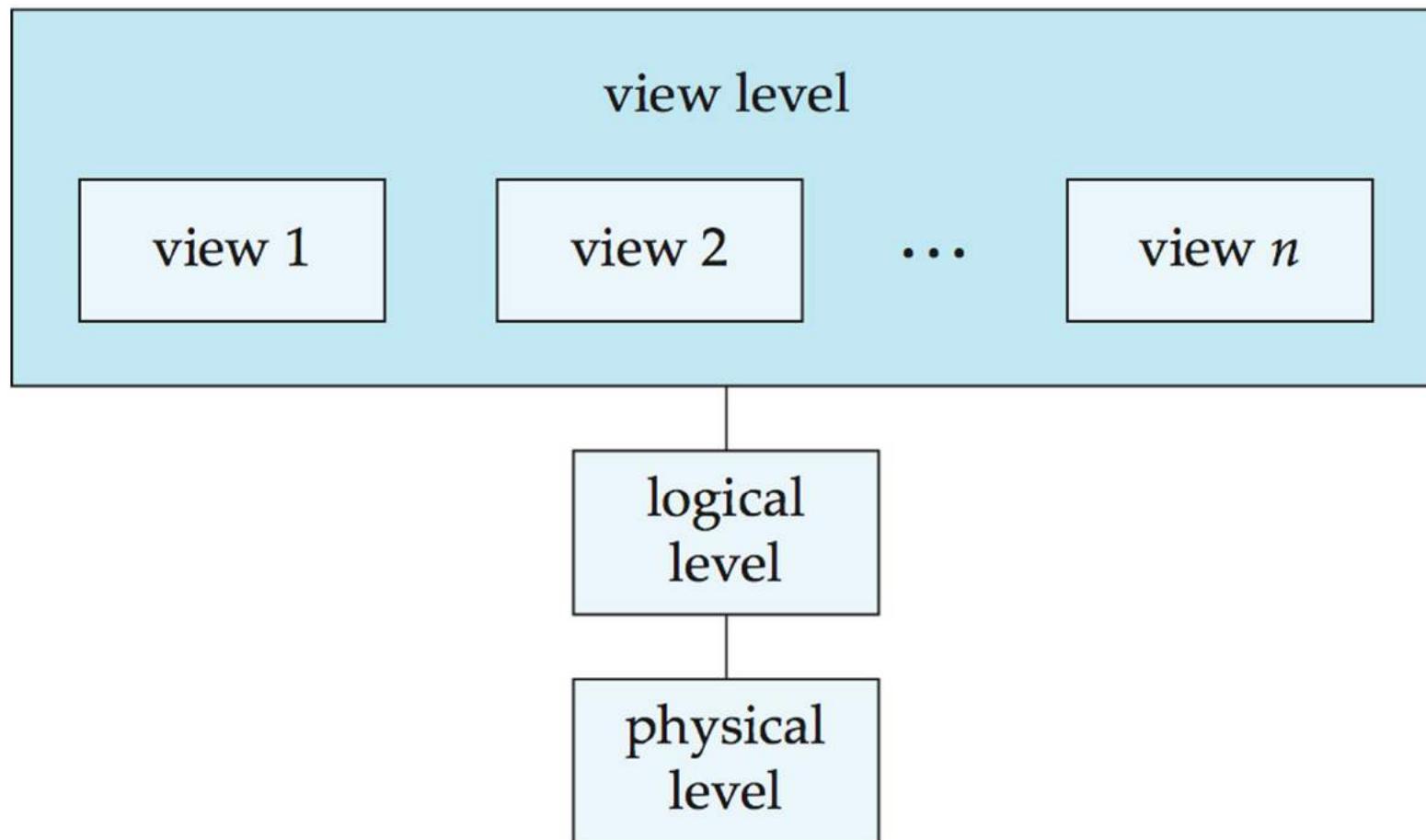
Turing Award 1981

Kolumny

Wiersze

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Trzy poziomy abstrakcji danych



Fizyczna niezależność danych

- » Fizyczna niezależność danych - zdolność do modyfikowania schematu fizycznego bez zmiany logicznego
 - Aplikacje są zależne od logicznego schematu
 - Ogólnie: interfejsy między różnych poziomami i komponentami powinny być dobrze zdefiniowane tak aby zmiany w niektórych częściach nie wpływały znacząco na inne.



AGH

Instancje i schematy

- » **Instancia bazy danych** – aktualna zawartość bazy danych w określonym punkcie w czasie
- » **Schemat bazy danych** – ogólny projekt bazy danych
 - **Schemat fizyczny**: opisuje projekt BD na poziomie fizycznym
 - **Schemat logiczny** – opisuje projekt BD na poziomie logicznym – najważniejszy – bo aplikacje
 - zmieniany rzadko – jeżeli w ogóle
- » **Subschematy** – wiele na poziomie widoków

Instancje i schematy

Schemat BD:

Student (number, name, address, email)

Course (number, name)

Building (number, address)

Instancja BD:

<u>STUDENT</u>			
Number	Name	Address	Email
0165854	Bart Baesens	1040 Market Street, SF	Bart.Baesens@kuleuven.be
0168975	Seppe vanden Broucke	520, Fifth Avenue, NY	Seppe.vandenbroucke@kuleuven.be
0157895	Wilfried Lemahieu	644, Wacker Drive, Chicago	Wilfried.Lemahieu@kuleuven.be

<u>COURSE</u>	
Number	Name
D0I69A	Principles of Database Management
D0R04A	Basic Programming
DOT21A	Big Data & Analytics

<u>BUILDING</u>	
Number	Address
0600	Naamsestraat 69, Leuven
0365	Naamsestraat 78, Leuven
0589	Tiensestraat 115, Leuven

Język definicji danych Data Definition Language (DDL)

- » Specyfikacja notacji do definiowania schematu bazy danych

Przykład:

```
create table instructor (
    ID      char(5),
    name   varchar(20),
    dept_name varchar(20),
    salary  numeric(8,2))
```

- » Kompilator DDL generuje jako wynik zestaw szablonów tabel przechowywanych w **data dictionary** (słowniku danych)
- » Słownik danych zawiera metadane
 - schemat bazy danych
 - warunki integralnościowe
 - klucz główny (ID jednoznacznie identyfikujące instruktorów)
 - autoryzacje
 - kto ma dostęp do czego

Język manipulacji danymi Data Manipulation Language (DML)

- » Język dla dostępu i manipulowania danymi zorganizowanymi zgodnie z odpowiednim modelem danych
 - DML zwany również *językiem zapytań*
- » Dwie klasy języków:
 - **Czyste** - używane do udowodnienia właściwości dotyczących mocy obliczeniowej i optymalizacji
 - Algebra relacji
 - Tuple relational calculus
 - Domain relational calculus
 - **Komercyjne** – używane w komercyjnych systemach
 - SQL najpopularniejszy język zadawania zapytań

Język manipulacji danymi Data Manipulation Language (DML) c.d.

- » Istnieją zasadniczo dwa rodzaje języków manipulacji danymi
 - **Proceduralne DML** – wymagają od użytkownika aby określił, jakie dane są potrzebne i jak je uzyskać
 - **Deklaratywne DML** – wymagają od użytkownika określenia, jakie dane są potrzebne, bez określania sposobu ich uzyskania
- » Deklaratywne DML są zwykle łatwiejsze do nauczenia się i używania niż proceduralne
- » Deklaratywne DML są również nazywane nieproceduralnymi
- » Część DML, która obejmuje wyszukiwanie informacji, zwana jest **językiem** zapytań



AGH

Język zapytań SQL

- » Język zapytań SQL jest nieproceduralny. Zapytanie przyjmuje jako dane wejściowe kilka tabel (być może tylko jedna) i zawsze zwraca jedną tabelę.
- » Przykład zwracając wszystkich nauczycieli z wydziału Informatyki

```
select name  
from instructor  
where dept_name = 'Comp. Sci.'
```

- » Aby móc obliczyć złożone funkcje, SQL jest zwykle osadzony w jakimś języku wyższego poziomu
- » Programy aplikacji zazwyczaj uzyskują dostęp do bazy danych przez:
 - Rozszerzenia języka macierzystego (np. C++), pozwalające na uwzględnienie wstawianie SQL w treść kodu
 - Interfejs programów użytkowych (np. ODBC/JDBC) który dopuszcza wysłanie zapytań SQL do bazy danych

Proceduralny a deklaratywny DML

- » Proceduralny DML
 - DML jawnie określa **jak** się poruszać w bazie danych
 - record-at-a-time DML
 - brak procesora przetwarzania zapytań
- » Deklaratywny DML
 - DML określa, **jakie** dane należy pobrać lub **jakie** zmiany należy wprowadzić
 - set-at-a-time DML
 - procesor przetwarzania zapytań

Projektowanie bazy danych

- » Określanie specyfikacji i analizowanie wymagań – projekt koncepcyjny.
- » Logiczny projekt – Decyzja o schemacie bazy danych. Opracowanie „dobrej” kolekcji schematów relacji.
 - Decyzja biznesowa – które atrybuty należy zapamiętać w bazie danych? (**what?**)
 - Decyzja informatyczna – Jakie schematy relacji powinny być i jak atrybuty powinny być rozłożone między różnymi schematami relacji? (**how?**)
- » Fizyczny projekt – Decyzja o warstwie fizycznej BD

Motor bazy danych

- » System bazodanowy jest podzielony na moduły
- » Funkcjonalne komponenty systemu bazodanowego można podzielić na
 - Menadżer składowania (*storage manager*)
 - Komponenty przetwarzania zapytań (*query processing*)
 - Komponent menadżera transakcji (*transaction manager*)



AGH

Menadżer składowania

- » Moduł programowy - dostarcza interfejs między danymi niskiego poziomu przechowywanymi w bazie danych a programami aplikacji i pytaniami wysyłanymi do systemu.
- » Odpowiedzialny za:
 - Interakcję z menedżerem plików systemu operacyjnego
 - Wydajne przechowywanie, pobieranie i aktualnianie danych
- » Składa się z :
 - Menadżera autoryzacji i integracji
 - Menadżera transakcji
 - Menadżera plików
 - Menadżera buforów



AGH

Menadżer składowania c.d.

» Menedżer składowania implementuje wiele struktur danych w ramach fizycznej implementacji systemu:

- *Pliki danych* - przechowywanie samej bazy danych
- *Słownik danych* - przechowuje metadane dotyczące struktury bazy danych, w szczególności schemat bazy danych
- *Indeksy* - mogą zapewnić szybki dostęp do danych. Indeks bazy danych dostarcza wskaźniki do tych elementów danych, które posiadają określoną wartość



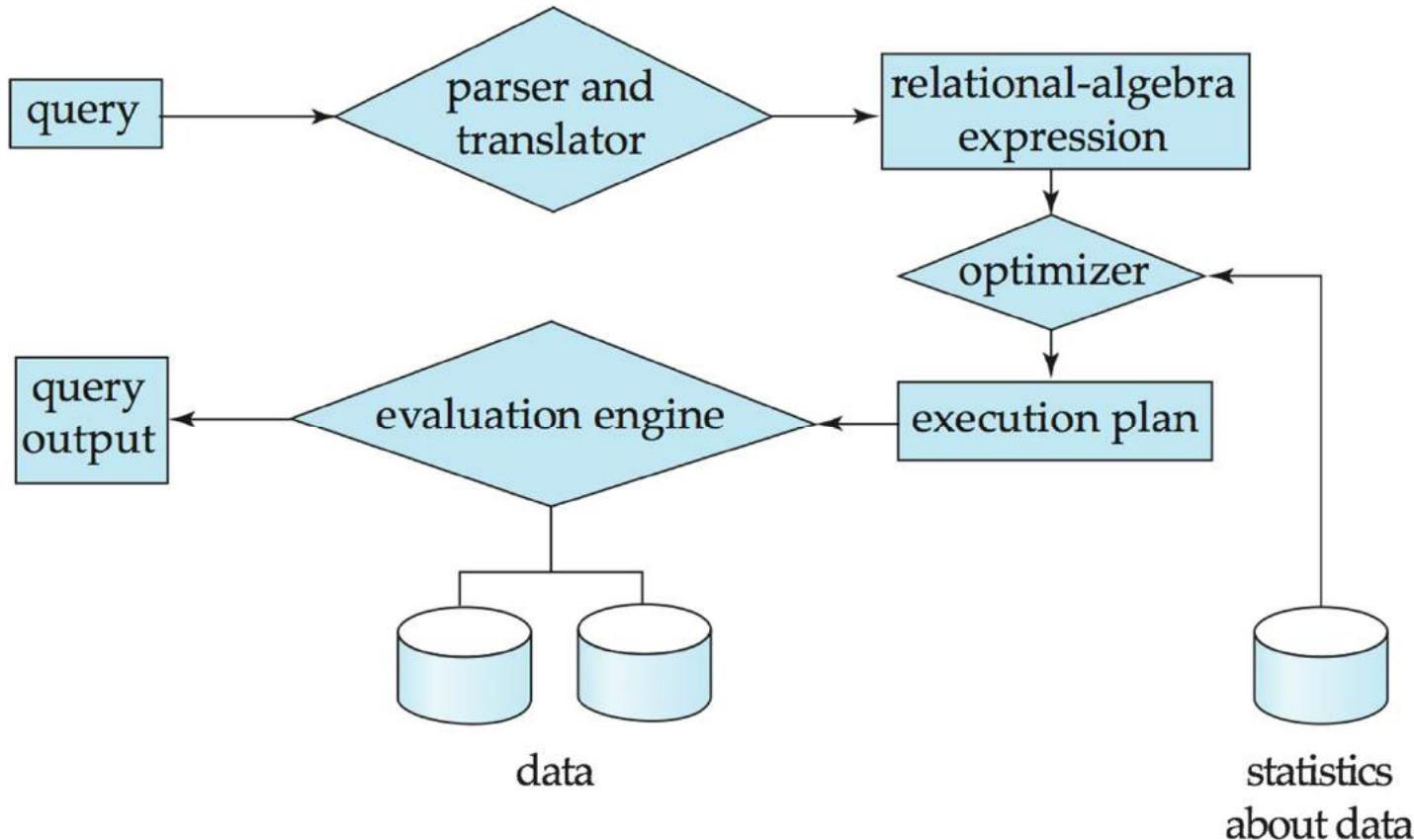
AGH

Procesor przetwarzania zapytań

- » Komponent procesora przetwarzania zapytań zawiera:
 - **interpretator DDL** - interpretuje wyrażenia DDL i zapisuje definicje w słowniku/katalogu danych
 - **kompilator DML** - tłumaczy wyrażenia DML w języku zapytań na plan wykonania składający się z instrukcji niskiego poziomu zrozumiałych dla silnika oceny zapytań
 - Kompilator DML wykonuje optymalizację zapytań: wybiera najtańszy plan wykonania spośród różnych alternatyw
 - **Silnik oceny zapytań** - wykonuje instrukcje niskiego poziomu wygenerowane przez kompilator DML

Przetwarzanie zapytań

1. Parsowanie i translacja
2. Optymalizacja
3. Ocena



Języki zapytań

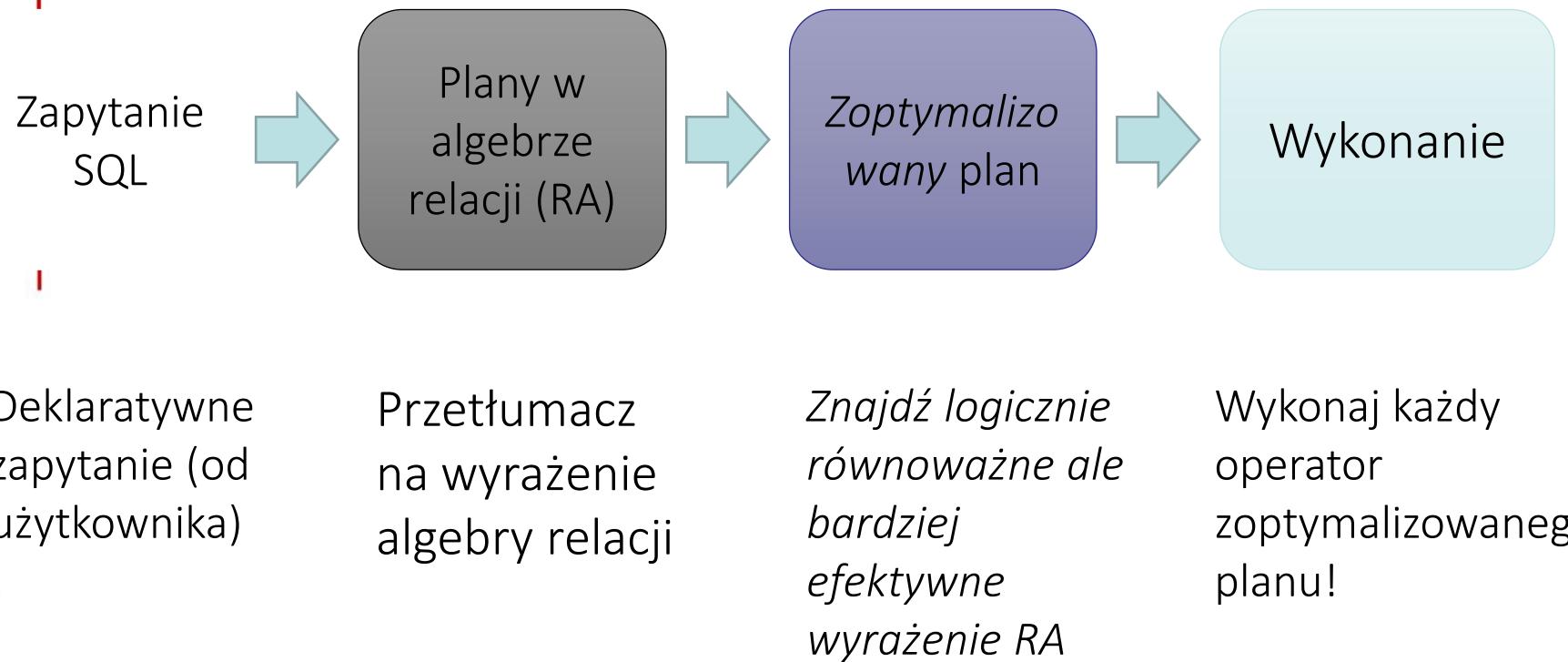
- » Algebra relacji (formalne)

$\pi_{ID} \sigma_{GPA > 3.7 \wedge cName = 'Stanford'}(Student \bowtie Apply)$

- » SQL (implementacyjne)

```
Select student.ID  
From Student, Apply  
where Student.ID=Apply.ID  
And GPA>3.7 and college='Stanford'
```

Jak zapytanie SQL jest przetwarzane



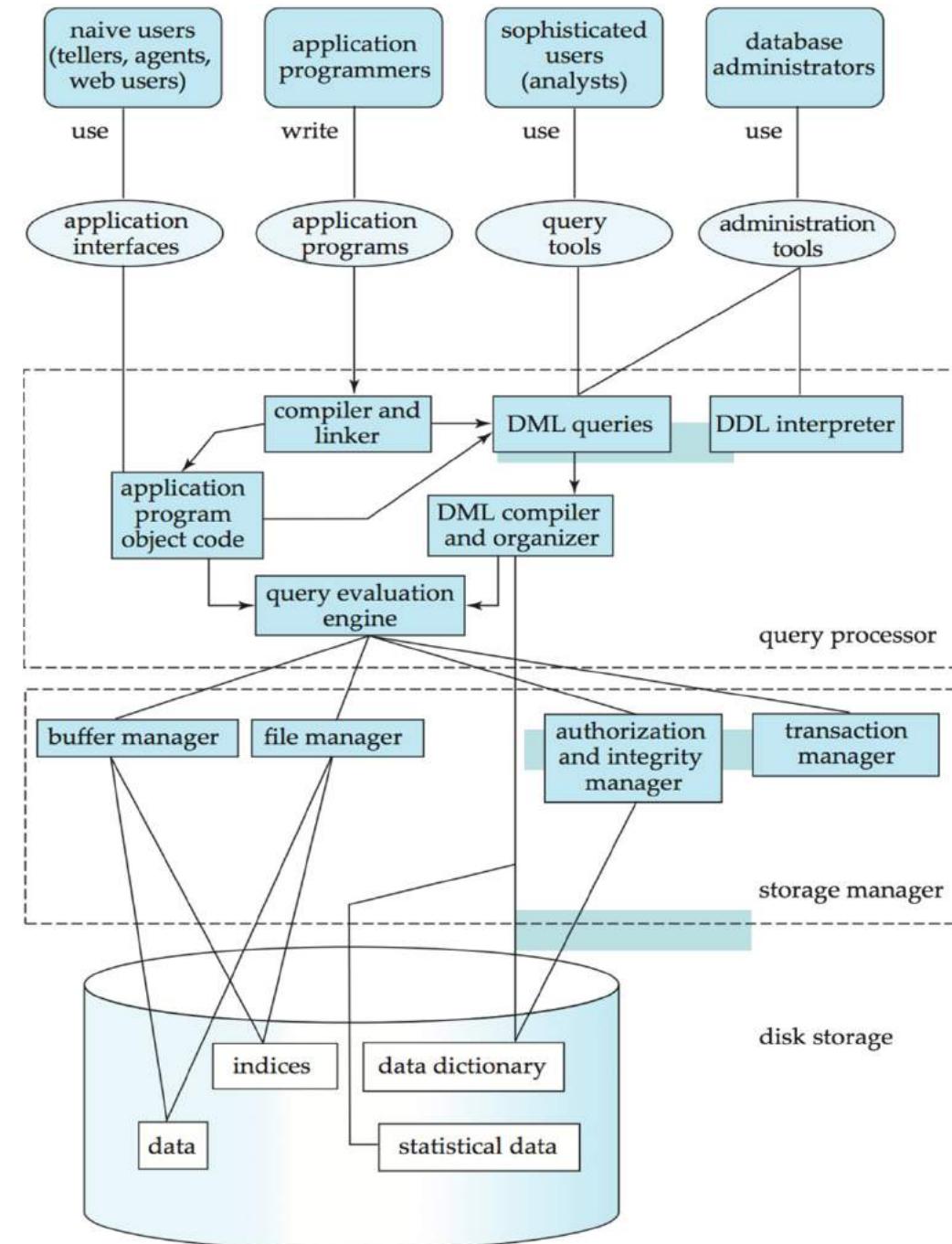


AGH

Zarządzanie transakcjami

- » **Transakcja** - zbiór operacji, który wykonuje się jako pojedyncza logiczna funkcja w aplikacji bazy danych (ACID)
- » **Komponent Menadżera Transakcji** zapewnia, że baza danych pozostaje w spójnym (poprawnym) stanie mimo błędów systemu (np. awarie zasilania, zawieszenie systemu) i niepowodzeń transakcji (np. brak środków na koncie).
- **Moduł kontroli współbieżności** kontroluje interakcje między współbieżnymi transakcjami w celu zapewnienia spójności bazy danych.

Architektura systemu bazodanowego



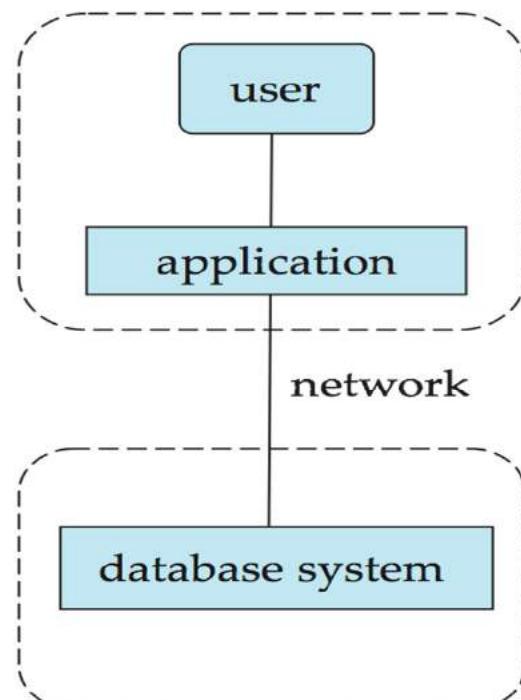
Architektura systemów baz danych

- » Zcentralizowana bazy danych
 - jeden do kilku rdzeni, pamięć współdzielona
- » Klient-serwer
 - jeden serwer wykonuje prace na rzecz wielu maszyn klienckich
- » Równoległe bazy danych
 - wiele rdzeni pamięci współdzielonej
 - dzielony dysk
- » Rozproszone bazy danych
 - geograficzne rozproszenie
 - heterogeniczność schematu/danych

Aplikacje bazodanowe

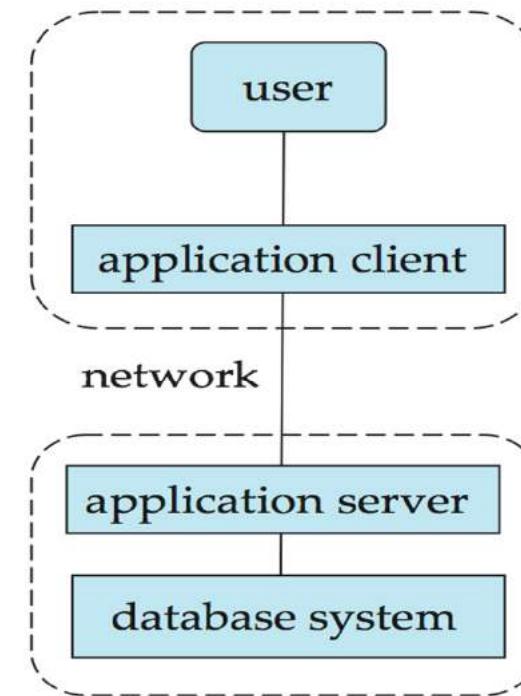
- » Aplikacje bazodanowe są zwykle podzielone na dwie lub trzy części
- **Dwuwarstwowa** architektura - aplikacja znajduje się na komputerze klienckim, gdzie wywołuje funkcjonalność systemu bazy danych na serwerze
- **Trójwarstwowa** architektura – maszyna kliencka działa jako interfejs użytkownika i nie ma żadnych bezpośrednich wywołań bazy danych
 - klient komunikuje się z serwerem aplikacji, zwykle poprzez interfejs formularzy
 - serwer aplikacji z kolei komunikuje się z systemem bazy danych w celu uzyskania dostępu do danych

Architektura dwuwarstwowa i trójwarstwowa



(a) Two-tier architecture

client



(b) Three-tier architecture

server

Rodzaje użytkowników baz danych – aktorzy na scenie

1. Administratorzy bazy danych
 - uwierzytelnianie dostępu do BD
 - koordynowanie i monitorowanie jej wykorzystania
 - pozyskiwanie niezbędnych zasobów programowych i sprzętowych
 - rozwiązywanie problemów z naruszeniem bezpieczeństwa lub zbyt długi czas odpowiedzi
2. Projektanci baz danych
 - odpowiadają za identyfikację danych, które mają być przechowywane w BD
 - wybór właściwych struktur do reprezentacji i przechowywania tych danych

Rodzaje użytkowników baz danych – aktorzy na scenie c.d.

3. Użytkownicy końcowi – ich codzienna praca wymaga dostępu do BD (przetwarzanie zapytań, aktualizacja informacji, generowanie raportów)
 - dorywczy – od czasu do czasu, mogą potrzebować różnych informacji – menadżerowie średniego lub wysokiego szczebla
 - naiwni (parametryczni) – wielokrotne wykonywanie zapytań i aktualizacja
 - doświadczeni – inżynierowie, naukowcy, analitycy
 - samodzielni
4. Analitycy systemowi i programiści aplikacji (inżynierowie oprogramowania)

Rodzaje użytkowników baz danych – aktorzy poza sceną

1. Projektanci i programiści SZBD
2. Programiści narzędzi (pakietów oprogramowania)
3. Operatorzy i personel pomocniczy



AGH

Historia systemów baz danych

lata 50'te i wczesne 60'te

- przetwarzanie danych na taśmach magnetycznych
 - taśmy magnetyczne pozwalały tylko na sekwencyjny dostęp
- karty perforowane jako wejście

» późne lata 60'te i 70'te

- dyski twardy pozwalały na bezpośredni dostęp do danych
- zbieranie danych, tworzenie baz danych, model sieciowy i hierarchiczny powszechnie stosowany
- Tedd Codd definiuje relacyjny model danych
 - prostota i możliwość ukrycia szczegółów implementacyjnych

» późne lata 70'te i 80'te

- IBM Research rozpoczyna tworzenie prototypu *System R*
- UC Berkeley (Michael Stonebraker) rozpoczyna tworzenie prototypu *Ingres*
- *Oracle* realizuje pierwszą komercyjną relacyjną bazę danych
- Wysoka wydajność (jak na te czasy) przetwarzania transakcyjnego

Ewolucja technologii bazodanowej – c.d.

» lata 80'te:

- Badawcze relacyjne prototypy zaczynają ewoluować w stronę komercyjnych systemów
 - SQL staje się standardem przemysłowym
- Równolegle i rozproszone systemy bazodanowe
- Obiektowe systemy baz danych

» lata 90'te:

- duże aplikacje wspomagające podejmowanie decyzji i eksploracji danych
- Ogromne hurtownie danych o pojemności wielu terabajtów
- Pojawienie się handlu internetowego



Ewolucja technologii bazodanowej – c.d.

» Lata 2000:

- Standardy XML i XQuery
- Big data storage systems
 - Google BigTable, Yahoo PNuts, Amazon,
 - Systemy NoSQL.
- Big data analysis
 - Map reduce i pochodne

» Późniejsze 2010

- Ewolucja systemów NoSQL
- Dane w chmurze
- Oprogramowanie jako usługa

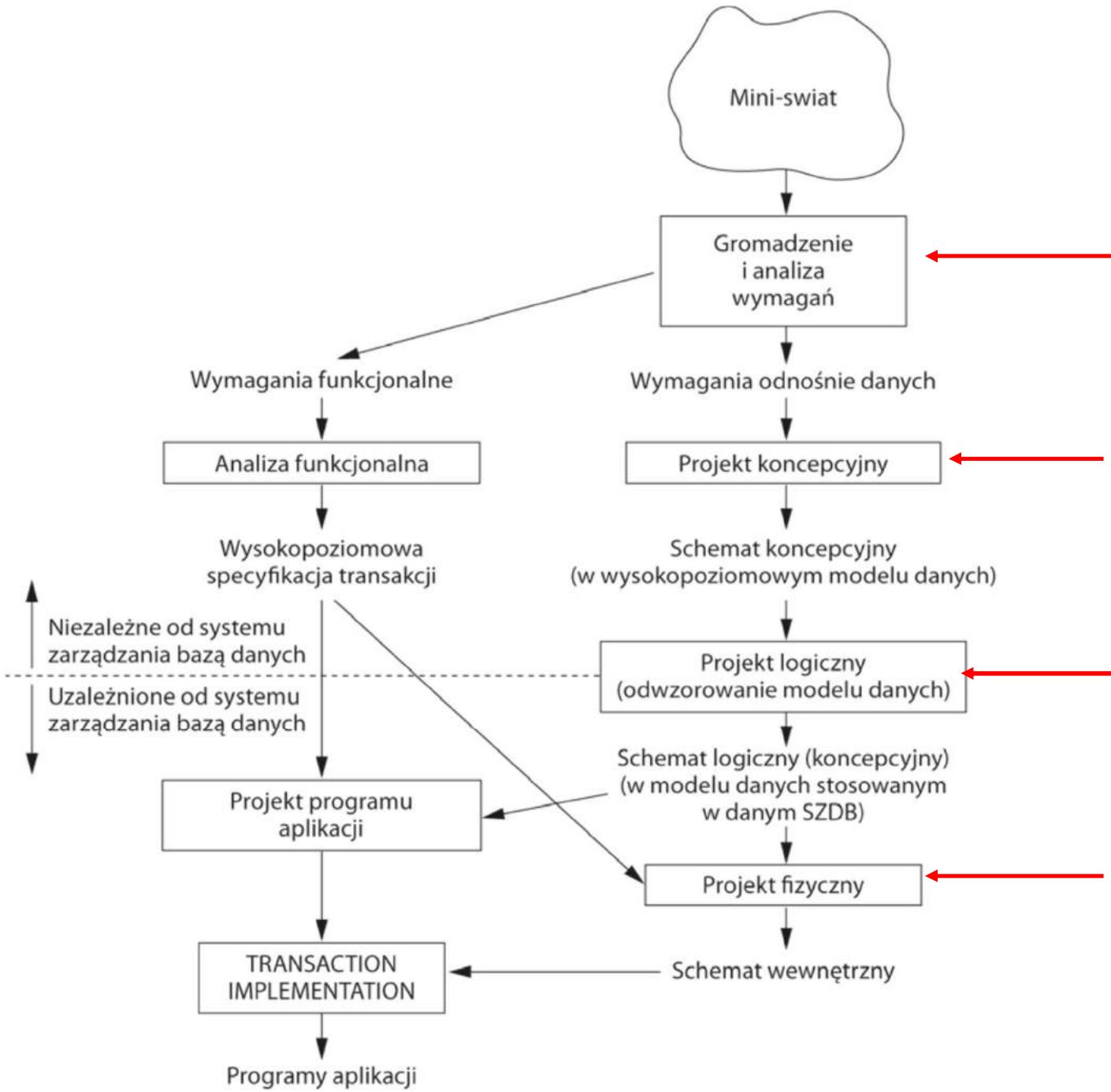


AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Wykład

Proces projektowania baz danych

Model ER i EER



Ad 1. Gromadzenie i analiza wymagań

- » Wywiady z końcowymi użytkownikami
 - aby zrozumieć i udokumentować ich wymagania danych
 - zwięźle zapisany zbiór wymagań użytkowników
- » Wymagania danych a wymagania funkcjonalne dla projektowanej aplikacji
 - określone operacje/transakcje
 - diagramy przepływu danych, sekwencji, scenariusze – inżynieria oprogramowania

Opis mini-świata – zarządzanie danymi pracowników, działów i projektów

Cel: baza ma umożliwić zarządzanie danymi pracowników, działów oraz realizowanych projektów

» Firma jest podzielona na działy. Każdy z działów ma unikatową nazwę, unikatowy numer oraz przydzielonego konkretnego pracownika, który tym działem kieruje. W BD należy utrzymywać datę początkową, od której dany pracownik kieruje wskazanym działem. Każdy dział może być rozproszony i znajdować się w wielu miejscowościach.

» Dział kontroluje wiele projektów, z których każdy ma unikatową nazwę, unikatowy numer oraz jedno miejsce realizacji.

Opis mini-świata – zarządzanie danymi pracowników, działów i projektów c.d.

- » W bazie danych musi być przechowywane nazwisko i Pesel, adres, wysokość pensji, płeć i data urodzenia każdego pracownika firmy. Pracownik musi być przypisany do jednego działu, ale może pracować nad wieloma projektami, które niekoniecznie muszą być kontrolowane przez ten sam dział. W bazie danych będziemy rejestrować liczbę godzin, które pracownicy poświęcają poszczególnym projektom w ciągu tygodnia. Dla każdego pracownika będziemy dodatkowo przechowywali informację o bezpośrednim zwierzchniku

Opis mini-świata – zarządzanie danymi pracowników, działów i projektów c.d.

- » Chcemy przechowywać (np. w celach ubezpieczeniowych) informacje o rodzinach poszczególnych pracowników. Dla każdego członka rodziny pracownika firmy w bazie danych będą utrzymywane następujące elementy danych: imię, płeć, data urodzenia oraz stopień pokrewieństwa z pracownikiem.

Ad 2. Projektowanie koncepcyjne

- » Stworzenie schematu koncepcyjnego w oparciu o wysokopoziomowy, koncepcyjny model danych
- » Schemat koncepcyjny = zwięzły opis wymagań danych
 - szczegółowe opisy typów encji, związków oraz ograniczeń
 - wyrażane za pomocą elementów wysokopoziomowego modelu danych

Ad 3. Projektowanie logiczne (odwzorowanie modelu danych)

- » Implementacja w oparciu o jeden z SZBD
- » Wynik: schemat bazy danych zgodny z modelem implementacyjnym wykorzystywanym w danym SZBD
- » schemat koncepcyjny => przekształcany z wysokopoziomowego modelu danych w implementacyjny model danych



AGH

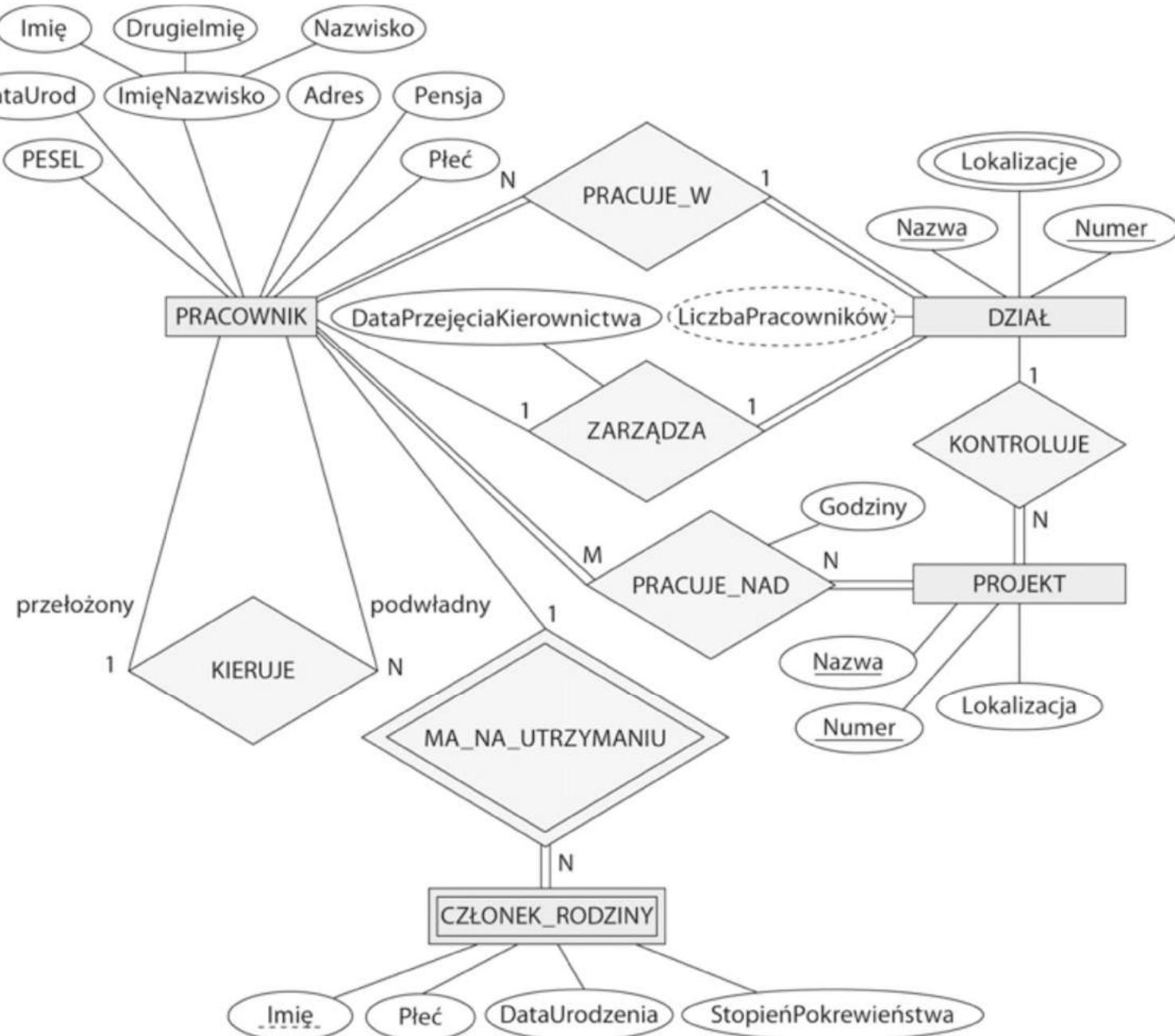
Ad 4. Projektowanie fizyczne

- » Definiuje się:
 - wewnętrzne struktury składowania
 - organizację plików
 - indeksy
 - ścieżki dostępu
 - fizyczne aspekty plików baz danych
- » Również projektowanie i implementowanie programów aplikacji bazy danych

Model związków encji

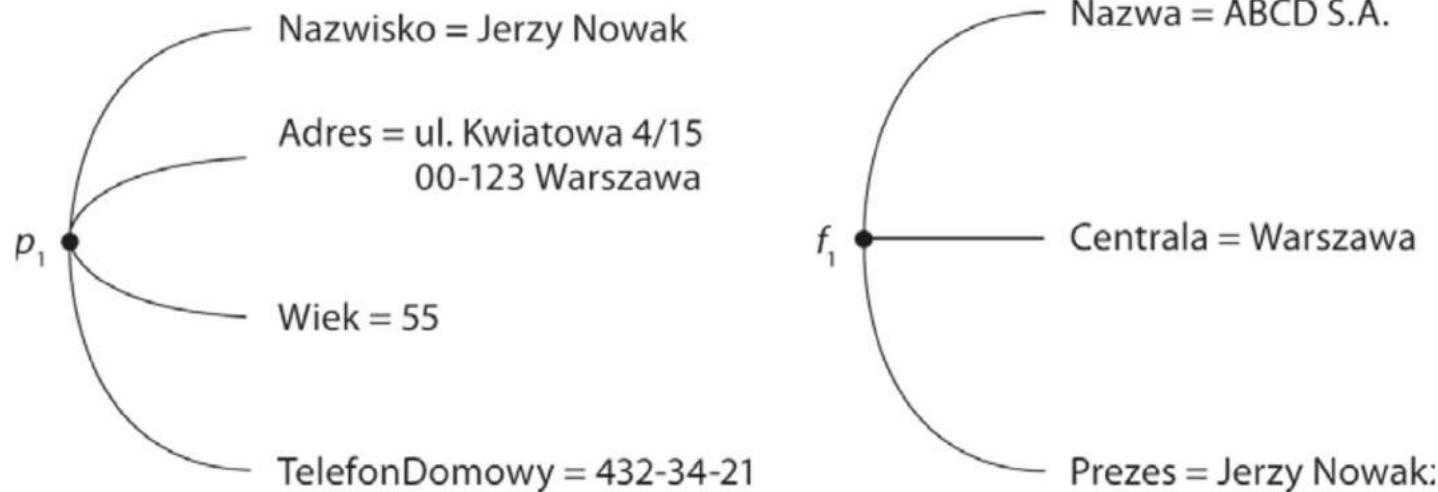
Entity relationship model ER

- » 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 w dziedzinie oprogramowania komputerowego
- » Jego praca zainicjowała badania nad modelowaniem koncepcyjnym
 - jeden z najpopularniejszych modeli danych do koncepcyjnego modelowania.
- » Model ER ma atrakcyjną i przyjazną dla użytkownika notację graficzną.
- » Składa się z trzech podstawowych elementów: typy encji, typy atrybutów i typy związków.



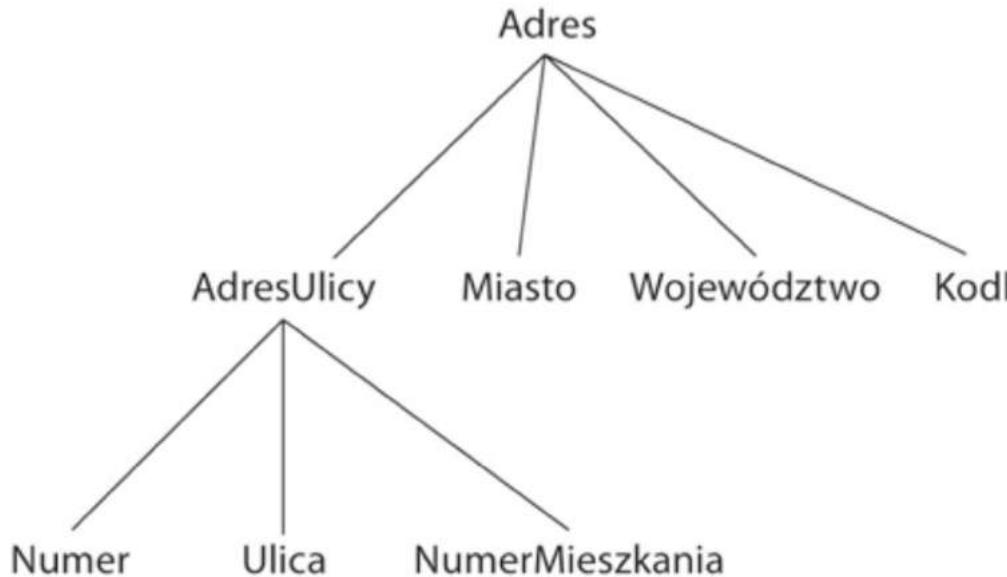
Encje i ich atrybuty

- » *Encja – byt ze świata rzeczywistego*
 - fizycznie istniejący obiekt: osoba, samochód...
 - obiekt, którego istnienie ma charakter koncepcyjny: firma, przedmiot...
- » Każda encja ma atrybuty – szczegółowe właściwości opisujące ją
 - pracownik: nazwisko, wiek, adres, pensja...



Typy atrybutów

- » Proste/złożone, jedno-/wielowartościowe, stałe/pochodne
- » Atrybuty złożone a proste (atomowe)



- » Atrybuty złożone przydatne gdy użytkownik czasem odwołuje się do danych złożonych tak jak do pojedynczego elementu danych



AGH

Typy atrybutów c.d.

- » Jednowartościowe a wielowartościowe
 - pojedyncza wartość dla konkretnej encji: wiek
 - zbiór wartości dla tej samej encji: kolory auta, stopnie naukowe
 - mogą mieć przypisane dolne/górne ograniczenia
- » Stałe a pochodne
 - wiek (pochodny) wywodzony z daty urodzenia (atrybut stały)



AGH

Wartości puste (null)

» Znaczenie:

- nie dotyczy: nr mieszkania, stopnie naukowe
- brak danych
 - wartość danego atrybutu istnieje, ale została pomijęta: wzrost
 - nie wiadomo czy wartość danego atrybutu w ogóle istnieje: telefon domowy

Skomplikowane atrybuty

- » Atrybuty złożone i wielowartościowe mogą być w dowolny sposób zagnieżdżone
 - można reprezentować dowolne zagnieżdżenie np. przez grupowanie składników atrybutu złożonego w (), oddzielenie składników przecinkiem a wielowartościowe w {}
 $\{NrTelefonu(\{Telefon(KodObszaru,Nr)\},Adres(AdresUlicy(Nr,Ulica, NrMieszkania),Miasto,Województwo,Kod))\}$



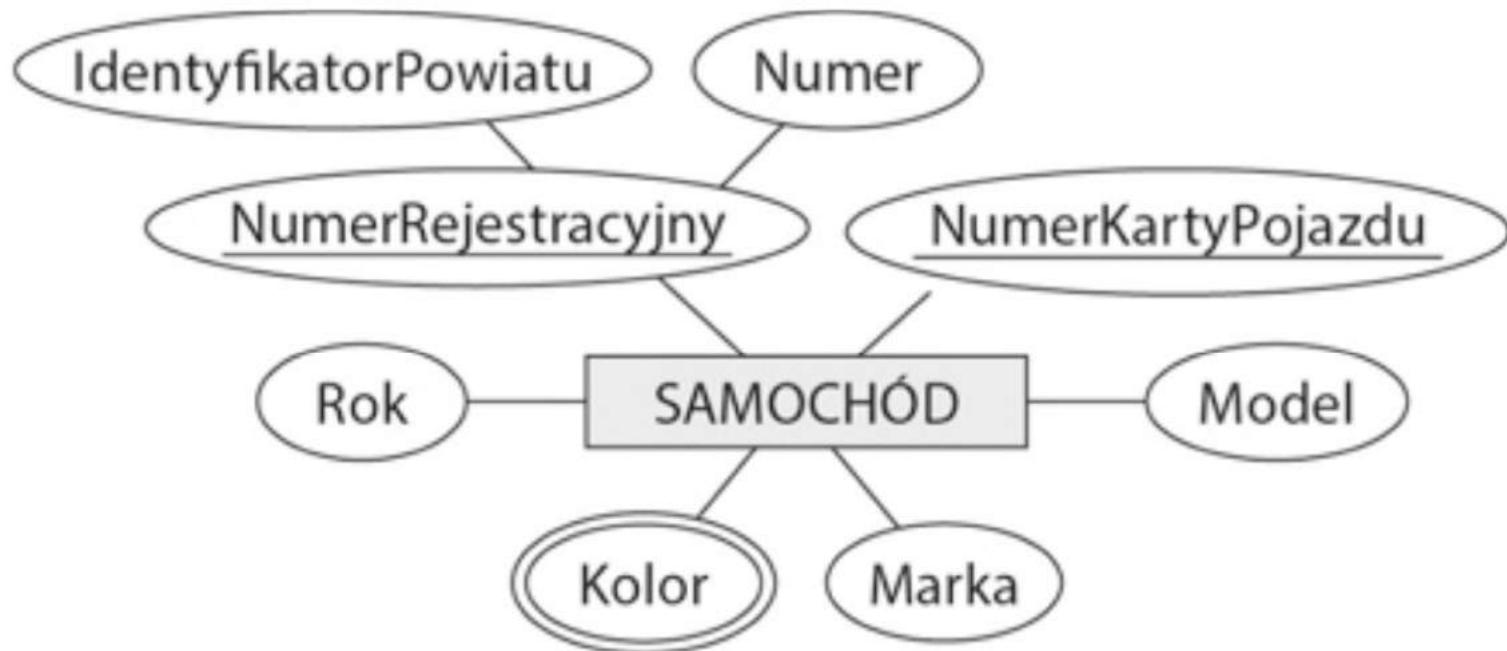
AGH

Typy encji i zbiory encji

- » Typ encji definiuje *zbiór (kolekcję)* encji, które mają takie same atrybuty
 - każdy typ encji w BD opisywany za pomocą jego nazwy i atrybutów
- » Zbiór encji – wyznaczony w dowolnym punkcie czasu zbiór wszystkich encji
- » Dwa typy encji: PRACOWNIK i FIRMA wraz z kilkoma encjami składowymi

Nazwa typu encji:	PRACOWNIK	FIRMA
Nazwisko, Wiek, Pensja	p_1 • (Jerzy Nowak, 55, 8000) p_2 • (Feliks Kosecki, 40, 3000) p_3 • (Justyna Borowska, 25, 2000) ⋮	f_1 • (ABCD S.A., Warszawa, Jerzy Nowak) f_2 • (Szybkie Komputery, Poznań, Wojciech Ignasiewicz) ⋮

Reprezentacja typów encji w diagramach ER



Typ encji opisuje *schemat* i *intensję kolekcji encji*, które mają taką samą strukturę.
Kolekcje encji konkretnego typu są grupowane w zbiory encji, które są nazywane *ekstensjami* typu encji

Atrybuty klucza typu encji

- » Ograniczenie unikatowości dla wskazanych atrybutów (tzw kluczy)
- » Atrybut klucza – jego wartości są różne w każdej konkretnej encji danego zbioru
- » Klucz może składać się z kilku atrybutów
 - atrybut złożony jako klucz w danym typie encji
 - musi być minimalny
- » Klucz – owal z podkreślona nazwą w środku
- » Własność unikatowości musi być spełniona dla *każdego zbioru encji* danego typu
 - wynika z rzeczywistych reguł występujących w mini-świecie

Zbiory wartości (dziedziny) atrybutów

- » Każdy prosty atrybut encji jest powiązany z odpowiednim zbiorem wartości (dziedziną wartości)
 - określa dopuszczalny zbiór wartości
 - nie są reprezentowane na diagramach ER
- » Z matematycznego punktu widzenia. atrybut A typu encji E , którego zbiorem wartości jest V , można zdefiniować za pomocą funkcji z E do zbioru potęgowego $P(V)$ zbioru V :

$$A: E \rightarrow P(V)$$

- » $A(e)$ – odwołanie do wartości atrybutu A należącego do encji e
- » Zbiór wartości V dla złożonego atrybutu A :

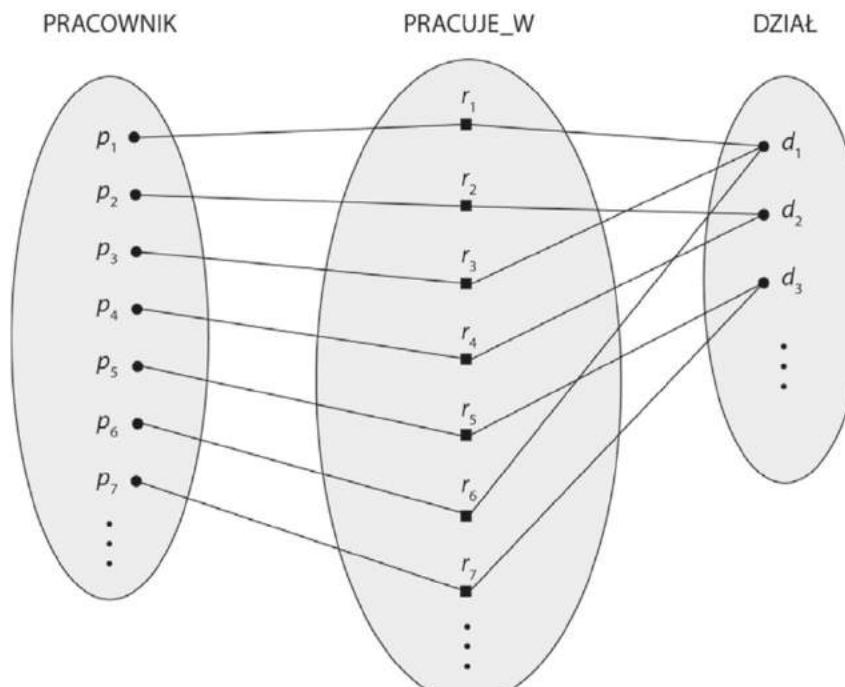
$$V = P(V_1) \times P(V_2) \times \dots \times P(V_n)$$

Projekt – początek [FIRMA.docx](#)



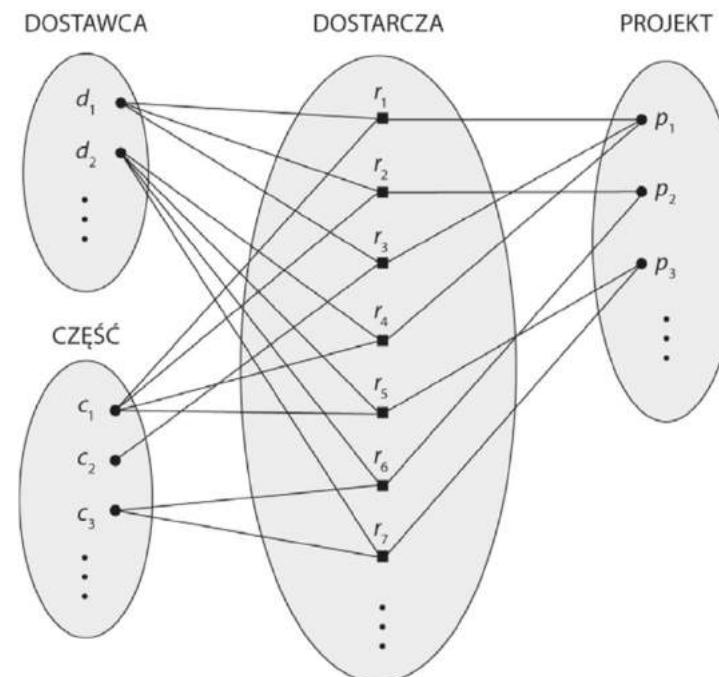
Typy, zbiory i egzemplarze związków

- » Typ związku R łączącego n typów encji (E_1, E_2, \dots, E_n) definiuje zbiór powiązań (zbiór związków) pomiędzy encjami zgodnymi z tymi typami encji
- » R – zbiór egzemplarzy związku r_i , gdzie każdy związek r_i łączy n pojedynczych encji (e_1, e_2, \dots, e_n)
 - każda encja e_j w związku r_i jest typu E_j , gdzie $1 <= j <= n$



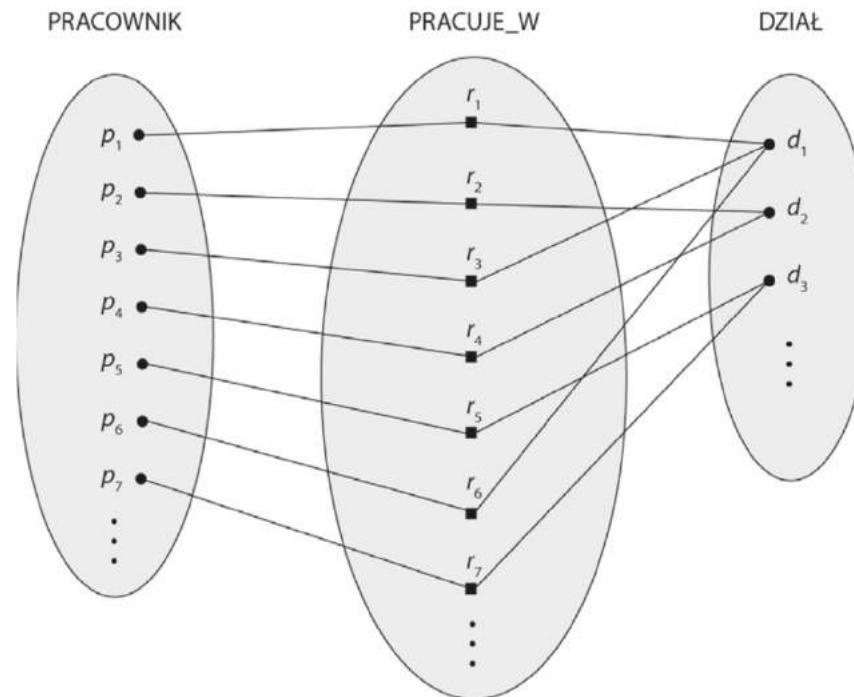
Stopień typu związku

- » Liczba typów encji składających się na dany związek
 - typ związku drugiego stopnia - typ binarny
 - typ związku trzeciego stopnia – trójskładnikowy (ternarny)



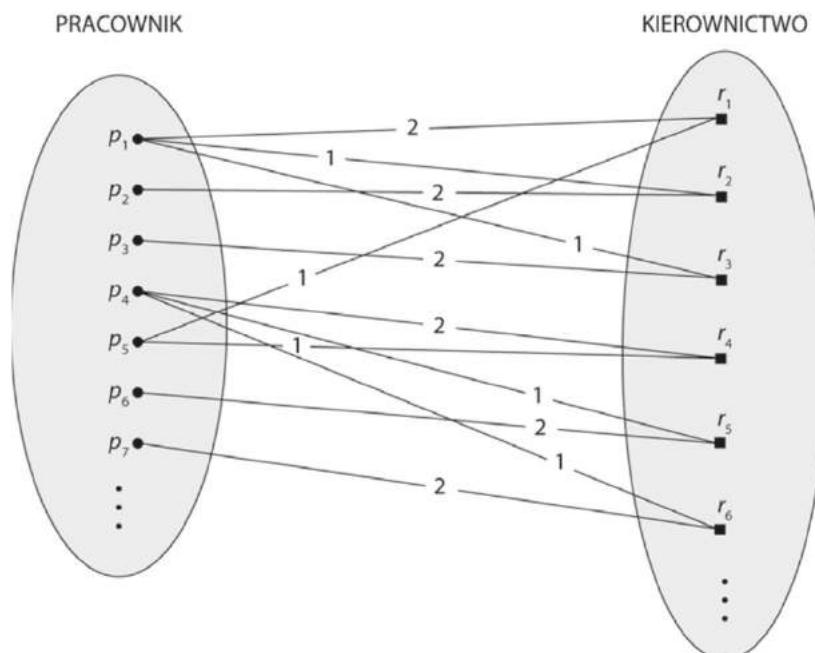
Związki w postaci atrybutów

» Czasem wygodnym rozwiążaniem jest traktować typy związków jakby były atrybutami



Nazwy ról i związki rekurencyjne

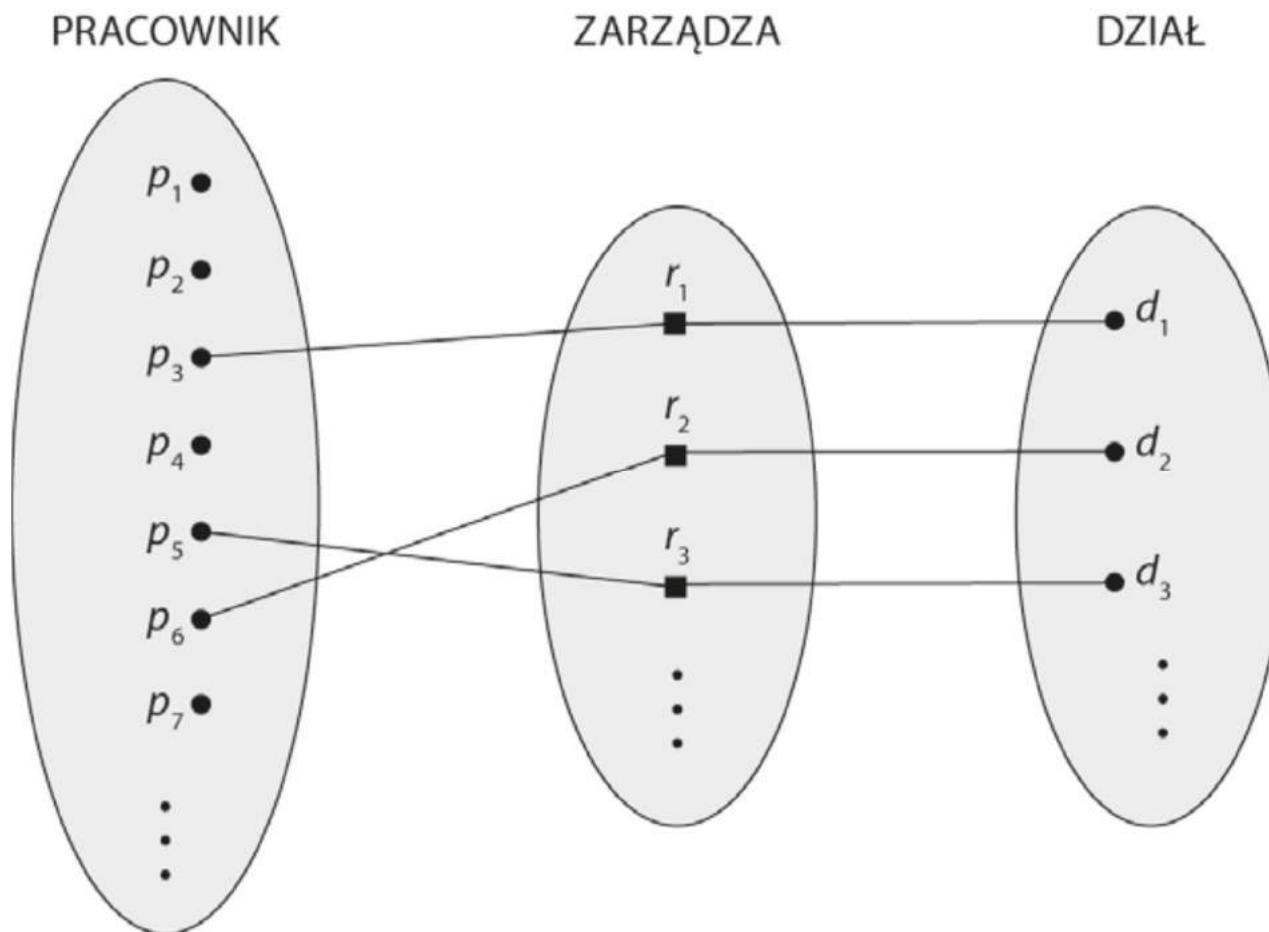
- » Każdy typ encji będący składnikiem jakiegoś typu związku pełni w tym związku konkretną rolę
 - *nazwy ról* określają znaczenie poszczególnych encji w każdym z egzemplarzy danego związku
- » *Ten sam typ encji* może występować w jednym typie związku więcej niż 1 raz – za każdym razem w *innej roli* – *związki rekurencyjne*



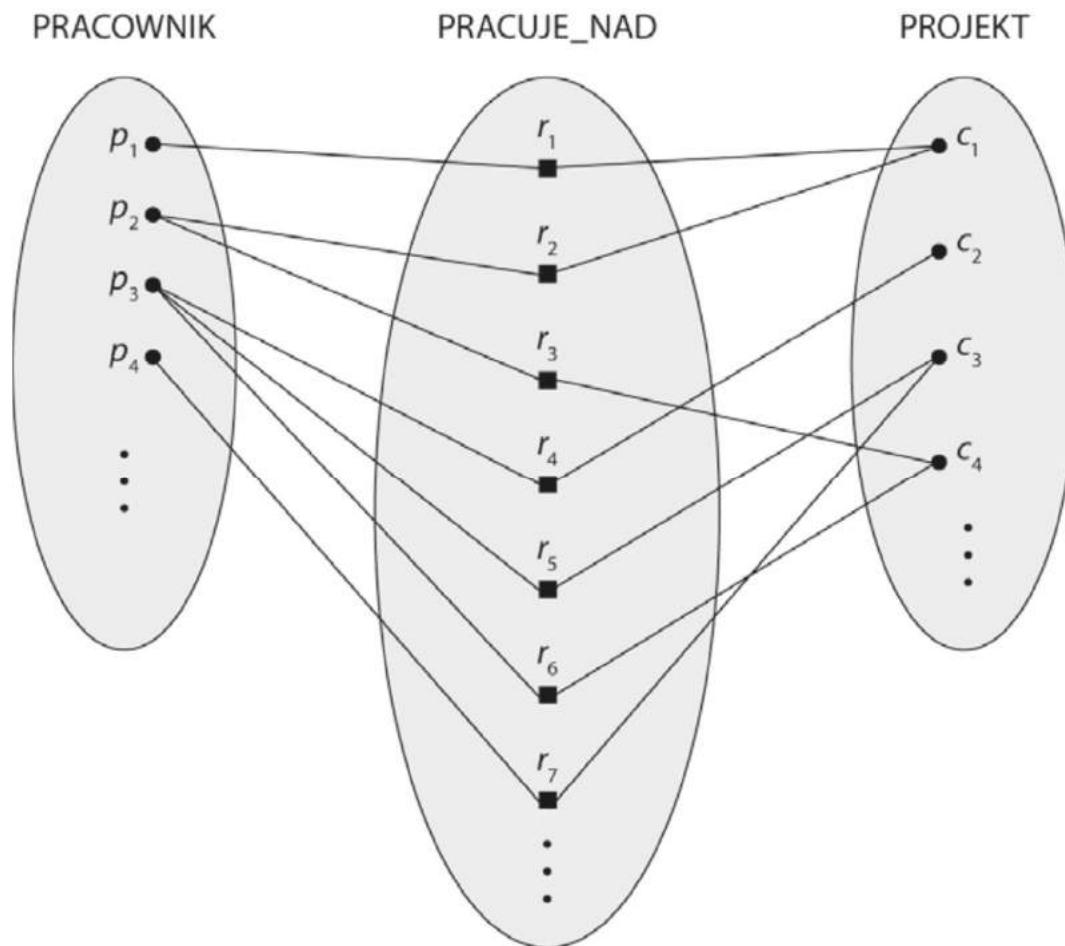
Ograniczenia dla typów związków

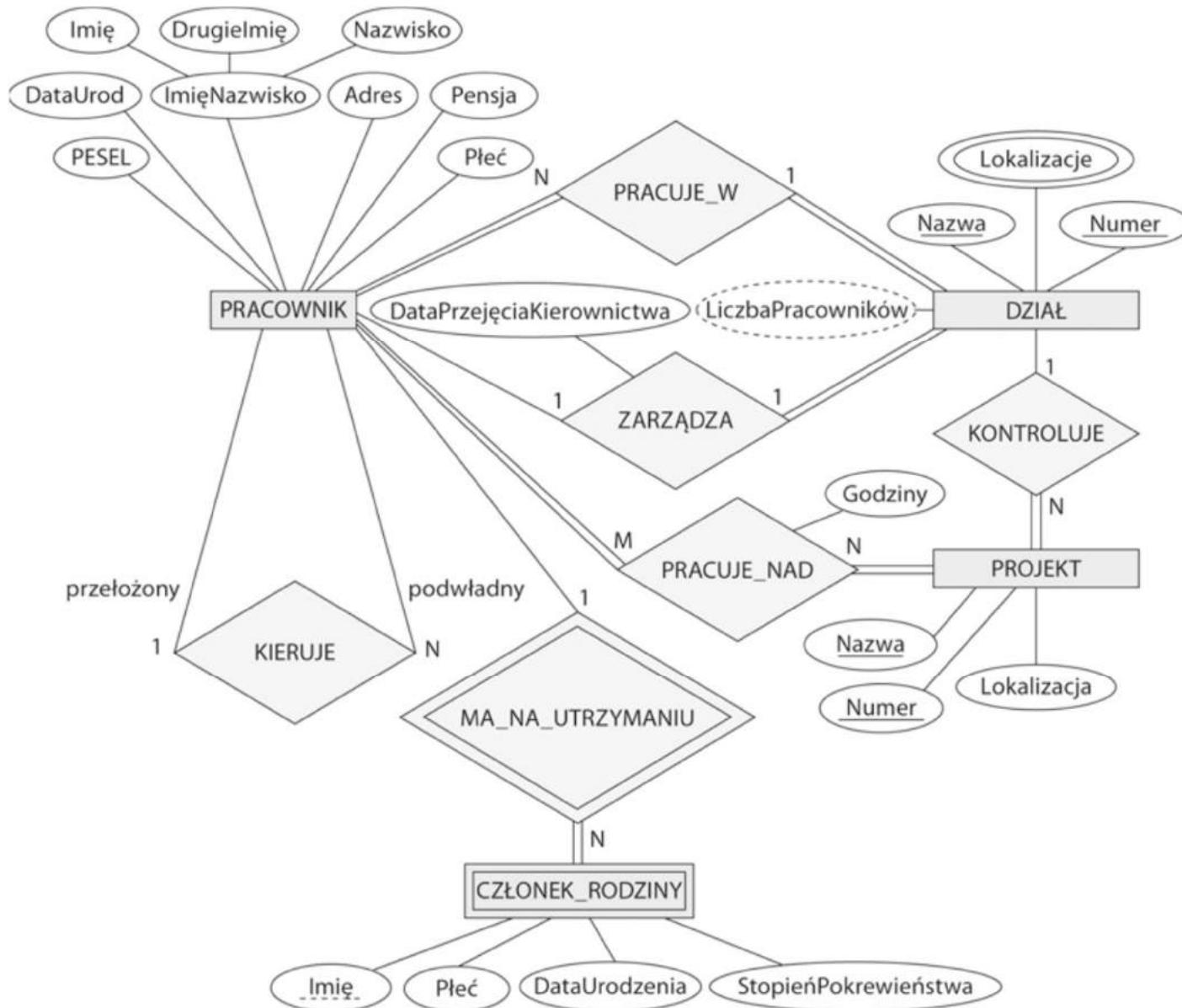
1. Ograniczenia współczynnika liczności
2. Ograniczenia udziału
 - » *Współczynnik liczności* dla związku binarnego - maksymalna liczbę egzemplarzy tego związku, w których może występować pojedyncza encja
 - PRACUJE_W
 - DZIAŁ:PRACOWNIK wynosi 1:N
 - każdy dział może być związany z (może zatrudniać) dowolną liczbą pracowników (N)
 - ale pojedynczy pracownik może być związany tylko z 1 działem

Ograniczenia współczynnika liczności związek typu 1:1



Ograniczenia współczynnika liczności związek typu M:N







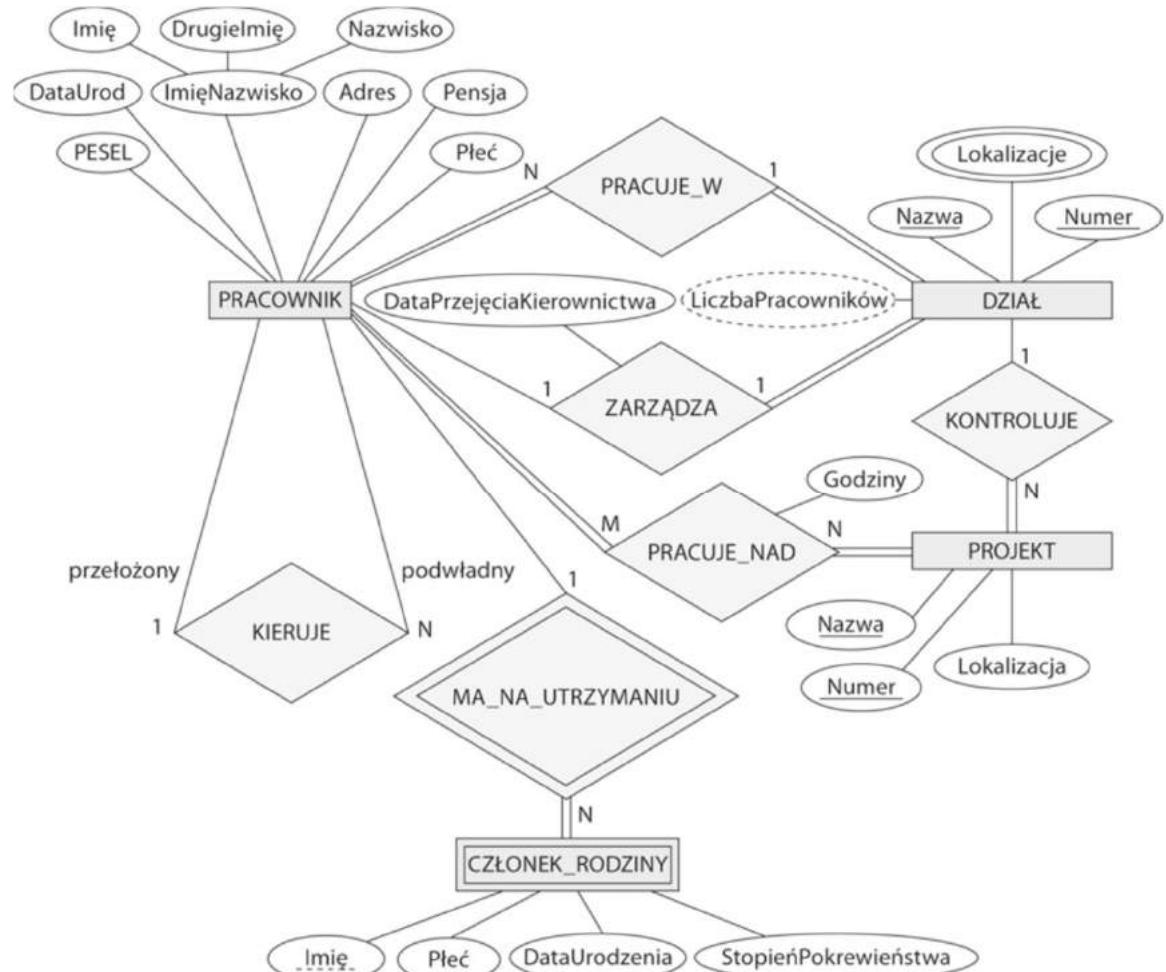
AGH

Ograniczenia udziału

- » Określają czy istnienie danej encji jest uzależnione od tego, czy jest ona związana z inną encją za pośrednictwem związku danego typu
 - wyznaczają minimalną liczbę egzemplarzy związku, w którym każda z encji musi występować – *ograniczenia minimalnej liczności*
- » Typy ograniczeń udziału:
 - *całkowite*
 - PRACUJE_W - każdy pracownik musi pracować w jakimś dziale
 - *pełny udział (udział całkowity)*
 - *zależność istnienia*
 - *częściowe*
 - ZARZĄDZA - nie każdy pracownik musi kierować działem
 - *udział częściowy*

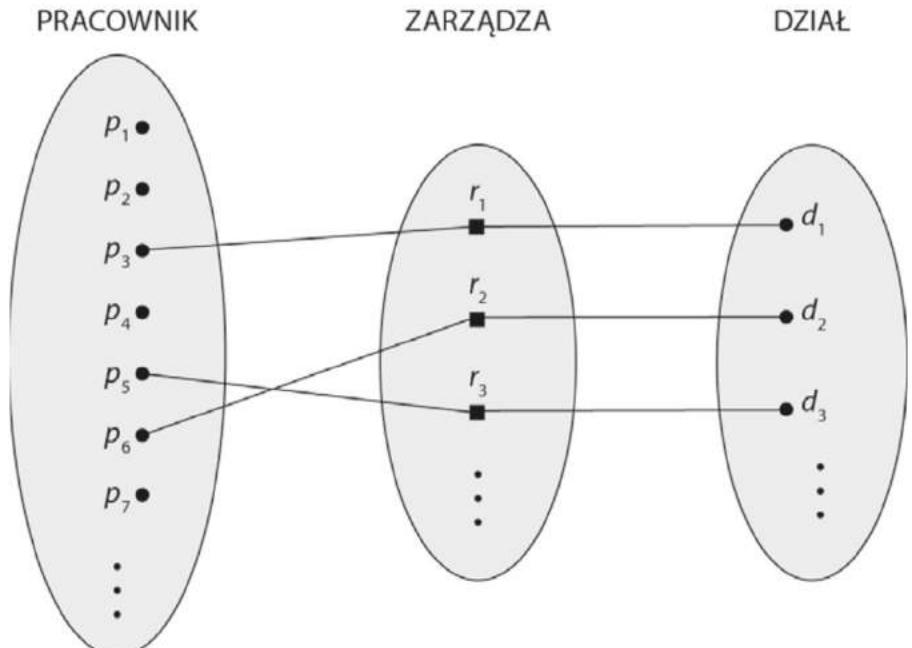
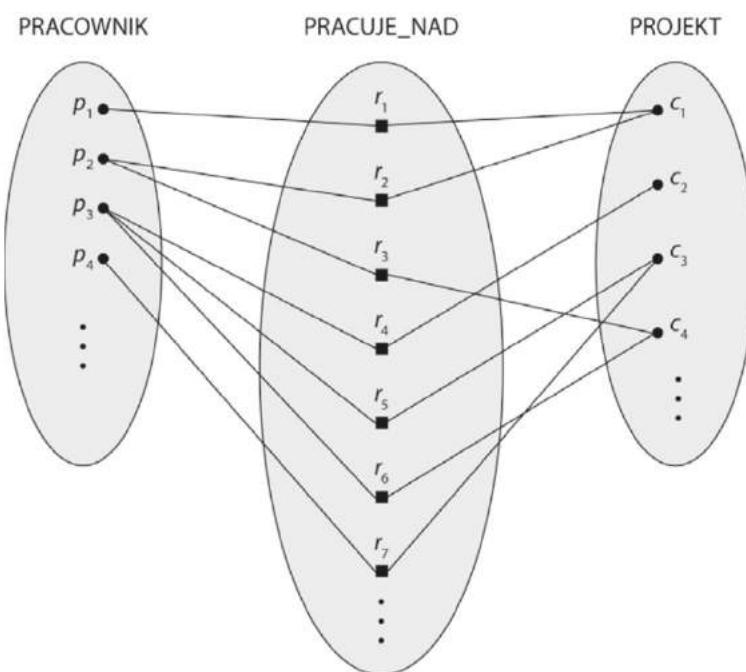
Ograniczenia strukturalne

- » Kombinacja współczynników liczności i ograniczeń udziału
- » W diagramach ER:
 - ograniczenia pełnego udziału (zależność istnienia) – podwójna linia
 - udział częściowy – pojedyncza linia



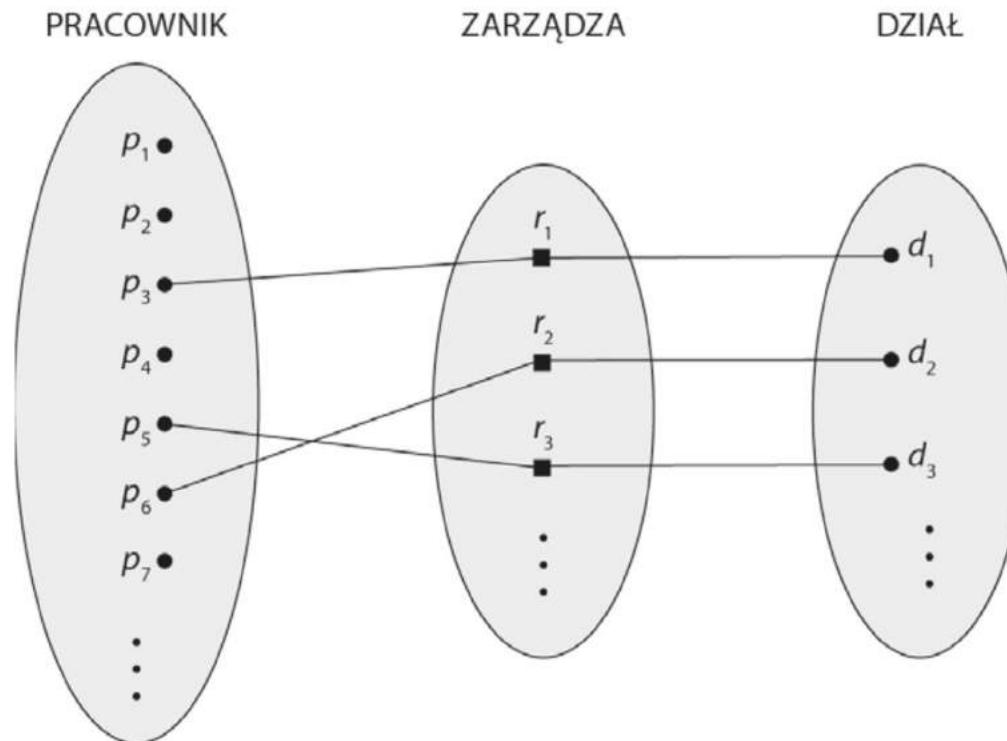
» Typy związków mogą mieć atrybuty

Atrybuty typów związków



Gdzie umieścić atrybuty typów związków?

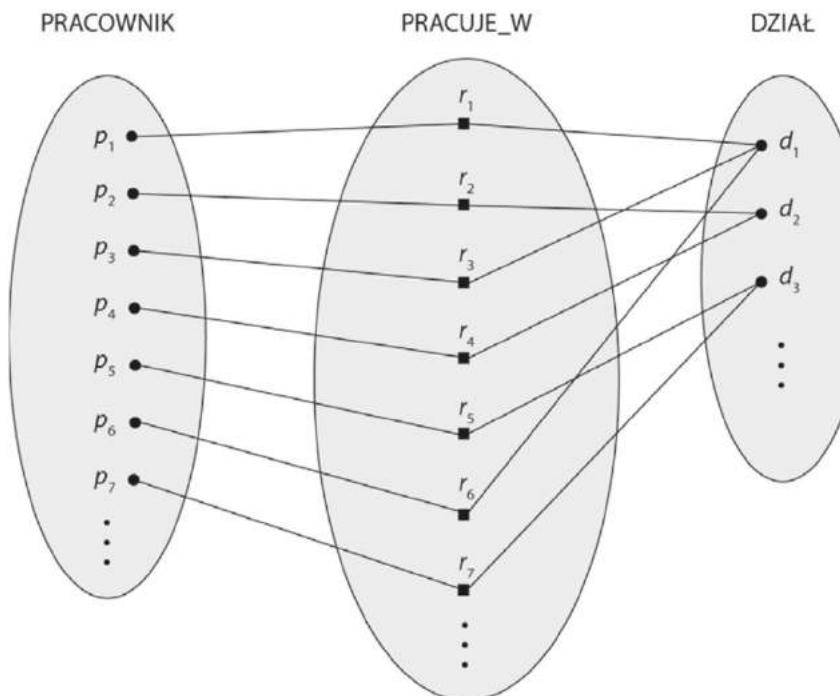
- » Umieszczanie uzależnione od współczynnika liczności:
 - 1:1 – obojętnie



Gdzie umieścić atrybuty typów związków?

» Umieszczanie uzależnione od współczynnika liczności:

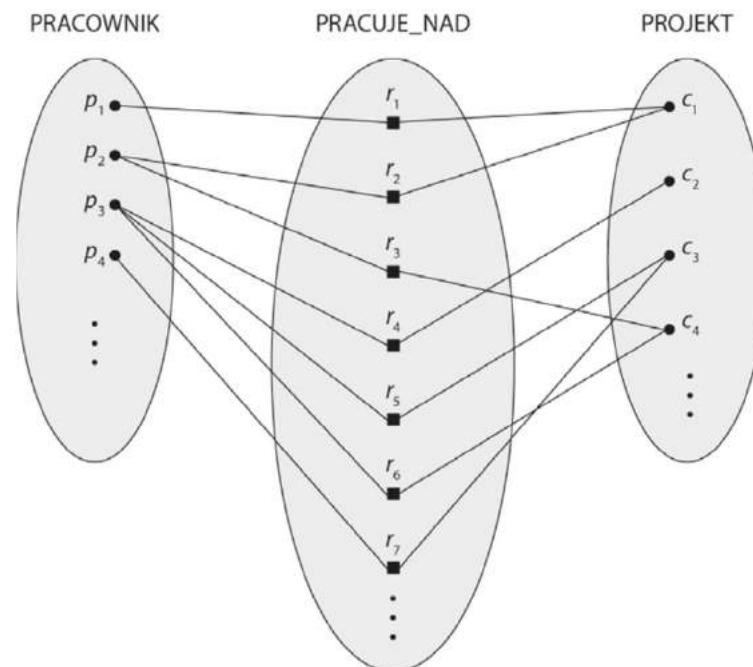
- 1:N – tylko do typów encji występujących po stronie N



Gdzie umieścić atrybuty typów związków?

» Umieszczanie uzależnione od współczynnika liczności:

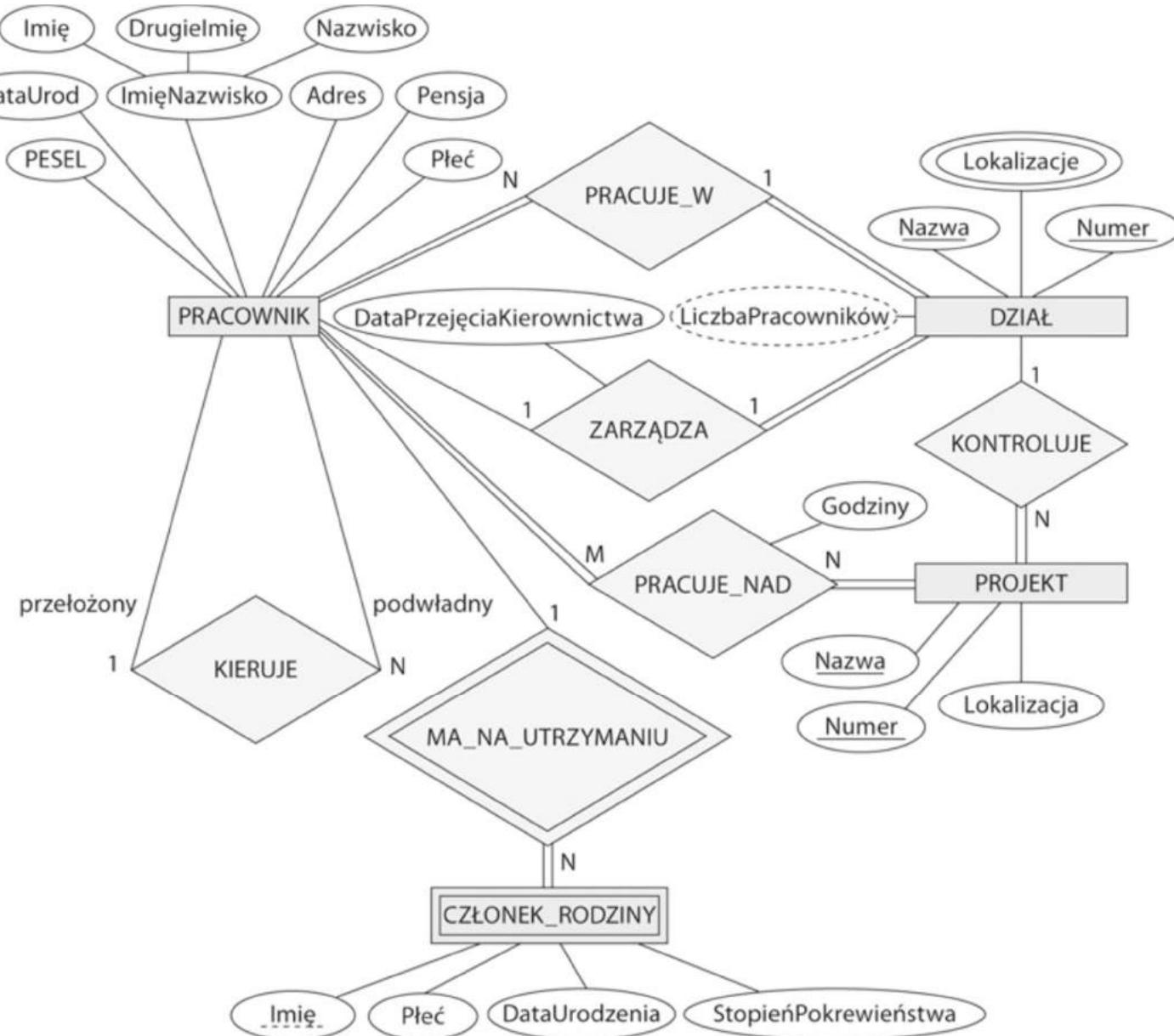
- M:N – niektóre atrybuty mogą być wyznaczane w oparciu o kombinację encji występujących w egzemplarzu związku, nie mogą być określone na bazie pojedynczej encji



Słabe typy encji

- » Typy encji, które nie zawierają atrybutów klucza
- » *Silne typy encji* – prawidłowe ze zdefiniowanymi atrybutami klucza
- » *Encje słabych typów* identyfikowane w oparciu o kombinacje powiązań z encjami innych typów oraz wartości jednego z ich atrybutów
- » *Identyfikujące (właścicielskie) typy encji* – inne typy związane ze słabymi encjami
 - typy związków wiążących słabe typy danych z ich właścicielami: *związki identyfikujące te typy*
- » Słabe typy danych zawsze mają nałożone *ograniczenie pełnego udziału* (zależności istnienia) względem identyfikujących je związków

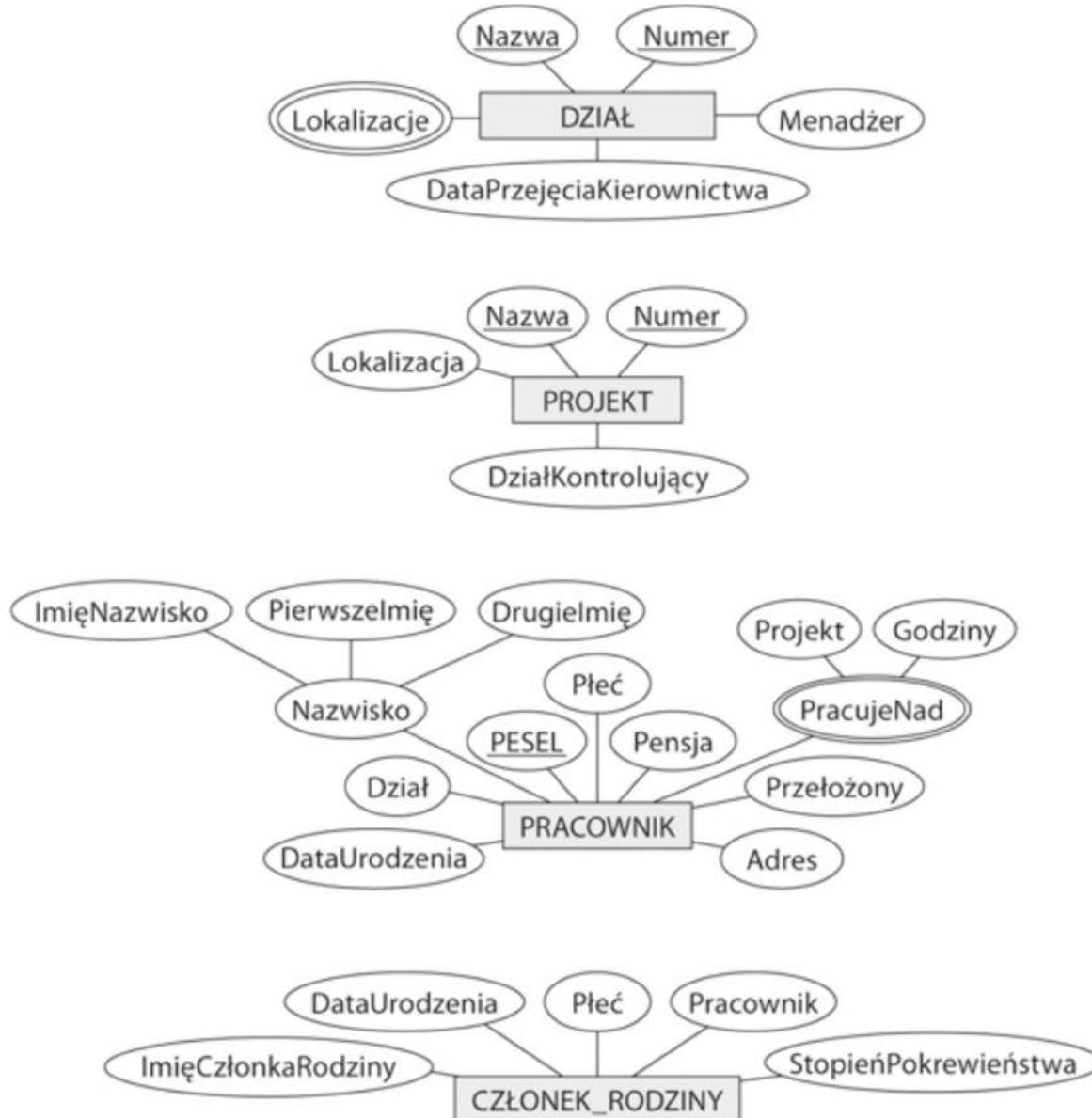
Słabe typy encji - przykład



Słabe typy encji - klucze

- » *Klucze częściowe (dyskryminatory) – zbiór atrybutów unikatowo identyfikujących słabe encje związane z tą samą encją właścielską*
- » Klucz częściowy może być złożony ze wszystkich *atrributów słabego typu encji*
- » Na diagramach związków encji (ER):
 - słabe typy encji i ich związki identyfikujące – podwójne linie otaczające reprezentujące je odpowiednie prostokąty i romby
 - klucz częściowy podkreślany za pomocą przerywanej lub kropkowanej linii

Projekt - przypomnienie



Projekt – zdefiniowane typy związków

- » **KIERUJE** 1:1 – łączy typy encji *Pracownik* i *Dział*
 - udział typu encji *Pracownik* w typie związków: częściowy
 - ale jednak: każdy dział przez cały czas musi mieć przydzielonego kierownika -> pełny dział udział typu związku
 - atrybut *DataPoczątkowa*
- » **PRACUJE_W** 1:N – łączy typy encji *Dział* i *Pracownik*
 - udziały pełne
- » **NADZORUJE** 1:N – łączy typy encji *Dział* i *Projekt*
 - udział typu encji *Projekt* - pełny
 - udział typu encji *Dział* - częściowy

Projekt – zdefiniowane typy związków c.d.

- » **KIERUJE** 1:N – łączy typy encji *Pracownik* (w roli przełożonego) z tym samym typem encji (w roli podwładnego)
 - udział obu występujących w tym związku encji: częściowy
- » **PRACUJE_NAD** M:N – łączy typy encji *Pracownik* i *Projekty*
 - udziały obu typów encji – pełne
 - atrybut *Godziny*
- » **JEST_NA_UTRZYZMANIU** 1:N – łączy typy encji *Pracownik* i *Członek_rodziny*
 - dodatkowo identyfikuje słany typ encji *Członek_rodziny*
 - udział typu encji *Pracownik* jest częściowy
 - udział typu encji *Członek_rodziny* - pełny

Projekt – modyfikacje

- » Do usunięcia atrybuty, które zostały przekształcone w odpowiednie związki



Prawidłowe nazewnictwo konstrukcji schematu

- » Przekształcanie opisu wymagań:
 - rzeczowniki – powinny tworzyć nazwy typów encji
 - czasowniki – powinny pełnić role nazw typów związków
 - nazwy atrybutów – na podstawie dodatkowych rzeczowników opisujących rzeczowniki określające typy encji
- » Oznaczanie związków binarnych:
 - od lewej do prawej i z góry na dół

Decyzje projektowe związane z tworzeniem schematu koncepcyjnego ER

- » Projektowanie schematu – iteracyjny proces udoskonalania
 - pojęcie może najpierw zostać zamodelowane w postaci atrybutu i potem przekształcone w związek, gdy okaże się, że dany atrybut jest odwołaniem do innego typu encji
 - atrybut występujący w wielu typach encji może zostać przeniesiony do niezależnego typu encji
 - odwrotnie: jeżeli istnieje typ encji *WYDZIAŁ* z pojedynczym atrybutem *NazwaWydziału*, i występuje w związku tylko z jednym z pozostałych typów encji (np. *STUDENT*) – można typ encji *Wydział* włączyć jako atrybut do typu encji *STUDENT*

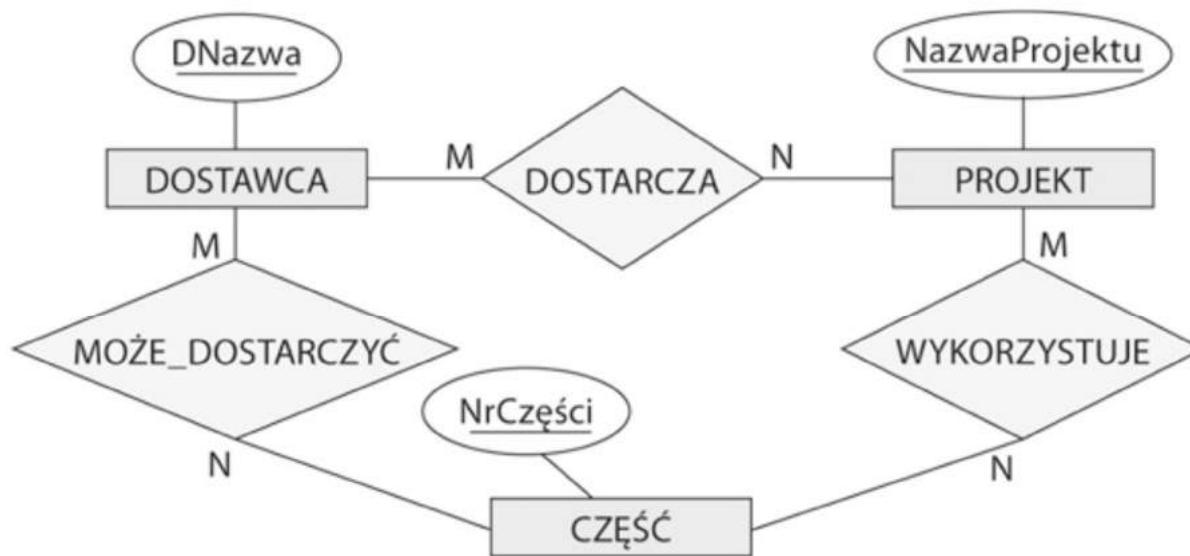
Typy związków stopnia wyższego niż drugi

- » *Stopień typu związków*
 - liczba typów encji występujących w tym typie
 - binarny
 - trójskładnikowy



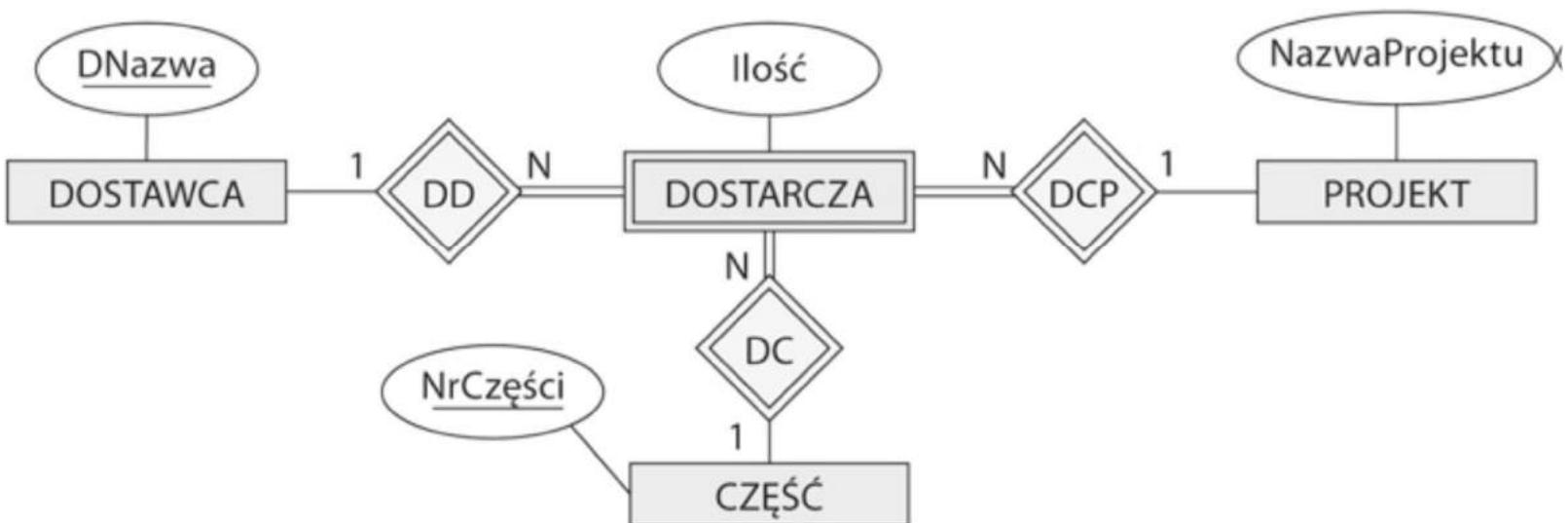
Typy związków stopnia wyższego niż drugi c.d.

- » Trzy związki binarne, które nie są równoważne związkowi DOSTARCZA



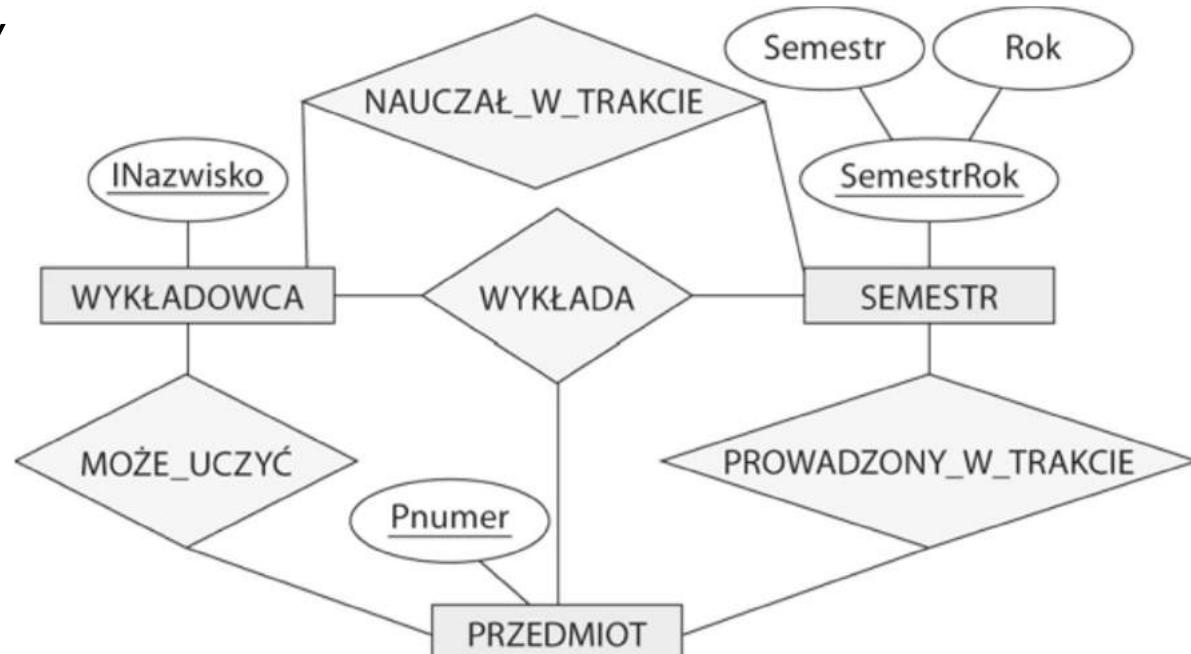
Typy związków stopnia wyższego niż drugi c.d

- » Związek DOSTARCZA reprezentowany w postaci słabego typu encji



Typy związków stopnia wyższego niż drugi c.d

- » MOŻE_UCZYĆ - wiąże przedmiot z wykładowcami, którzy *mogą prowadzić zajęcia* z danego przedmiotu
- » NAUCZANY_W_TRAKCIE – wiąże semestr z wykładowcami, którzy w czasie danego semestru prowadzą zajęcia z *jakiegoś przedmiotu*
- » PROWADZONY_W_TRAKCIE – wiąże semestr z przedmiotami które są nauczane w danym semestrze *przez któregoś z wykładowców*

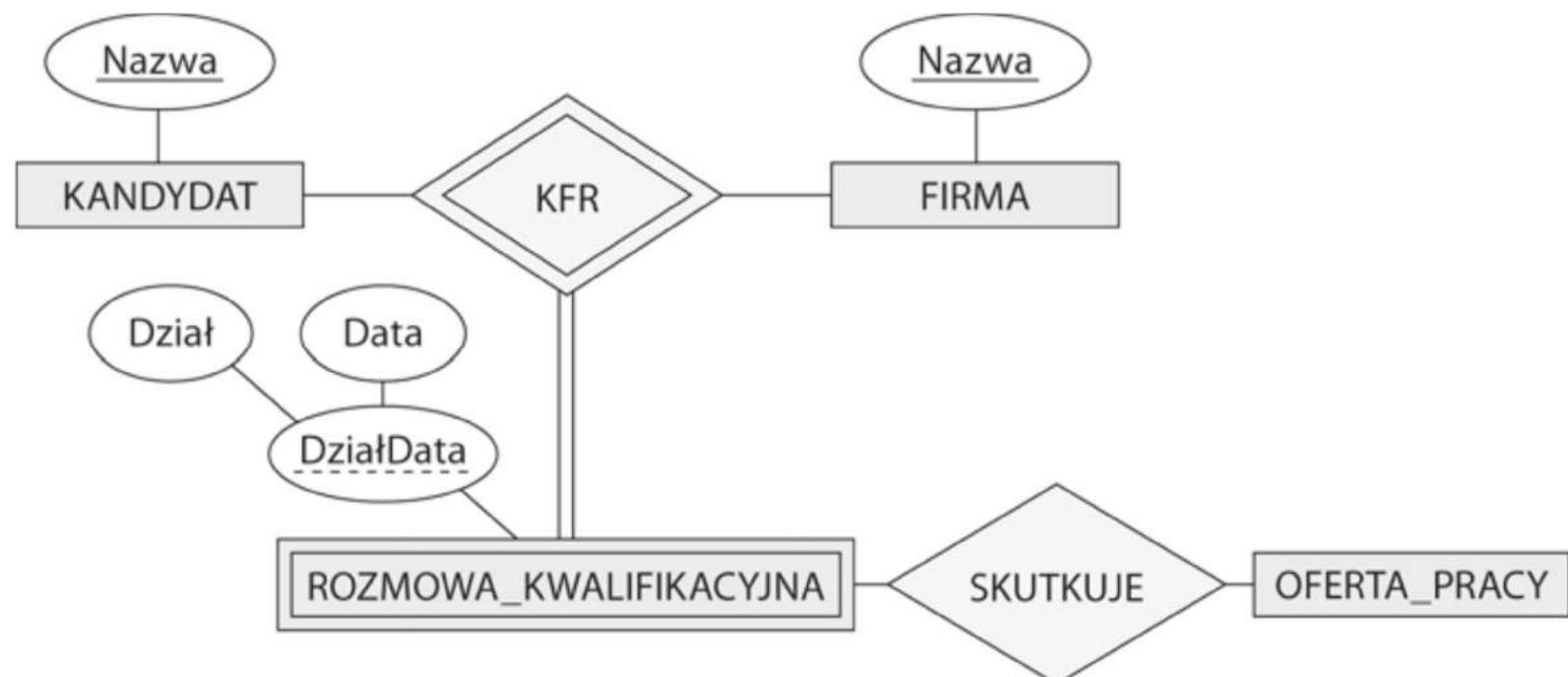


Rozwiązańe: dodatkowe ograniczenia

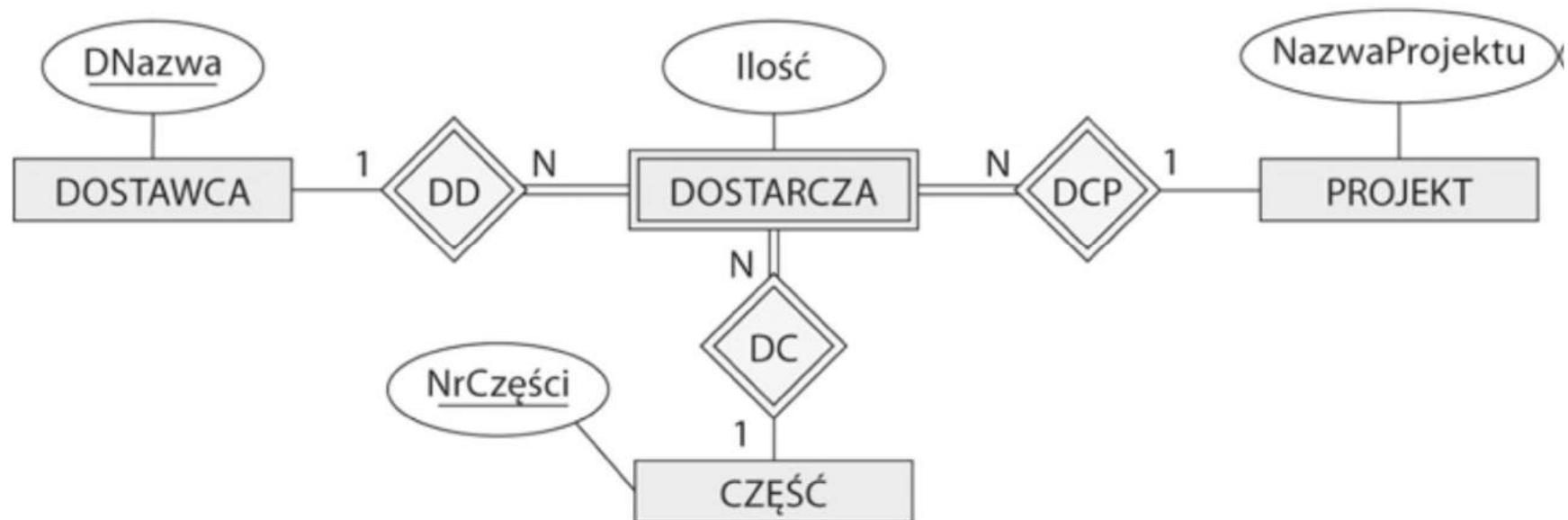
- » Generalnie: 3 związki binarne nie mogą zastępować pojedynczego związku trójskładnikowego
- » Ale: można wprowadzić *dodatkowe ograniczenia*:
 - licznosć typu związku MOŻE_UCZYĆ 1:1
 - można zrezygnować z typu związku trójskładnikowego WYKŁADA
 - istnienie jego encji można wywnioskować z trzech pozostałych typów związków binarnych
 - MOŻE_UCZYĆ, NAUCZANY_W_TRAKCIE, PROWADZONY_W_TRAKCIE

Rozwiązanie: słaby typ encji

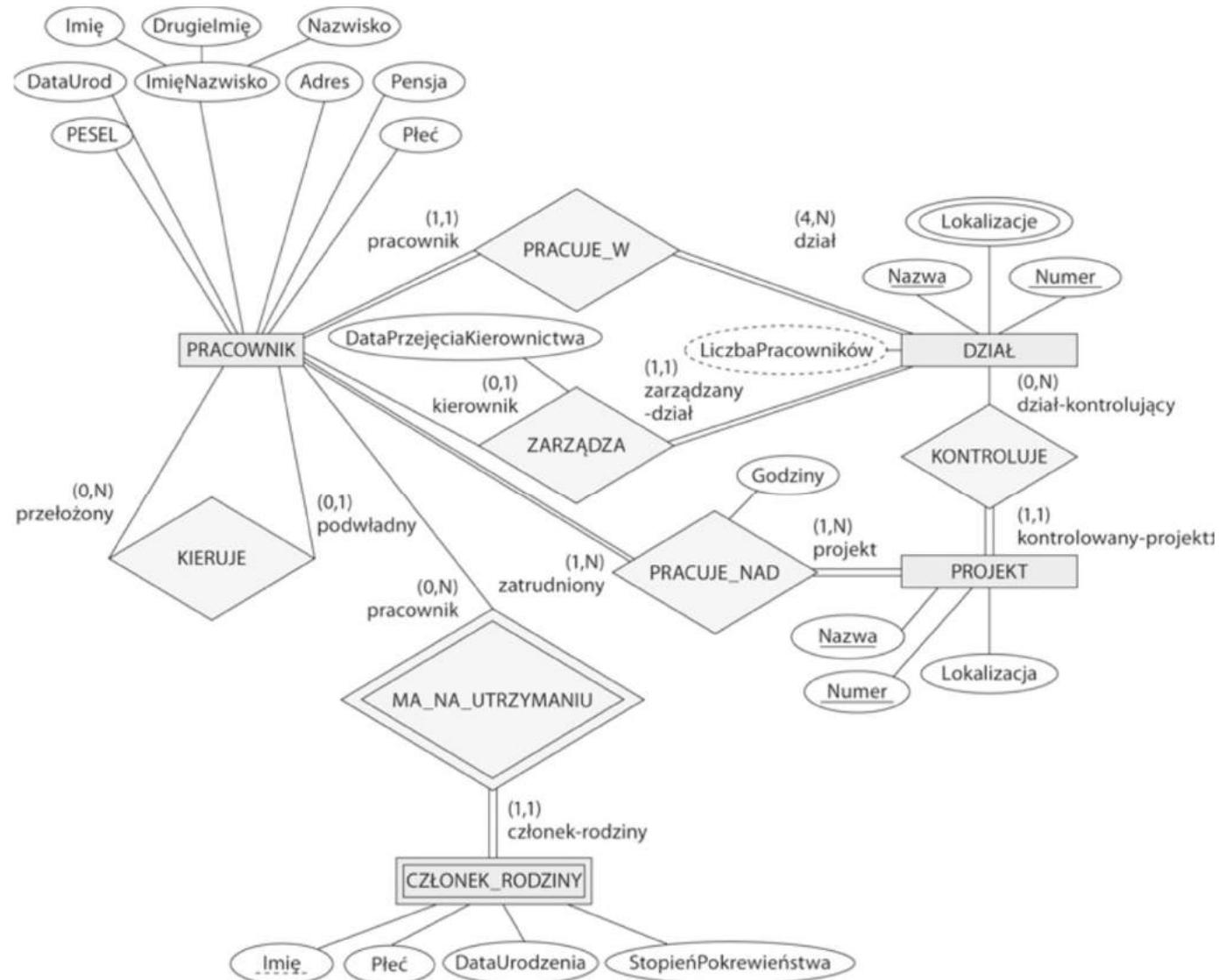
- » Kandydat może uczestniczyć w kilku rozmowach w tej samej firmie (w różnych działach/dniach)
 - ale oferta składana tylko w powiązaniu z jedną z rozmów



Przedstawianie ograniczeń strukturalnych - różne notacje



Przedstawianie ograniczeń strukturalnych - różne notacje c.d.



Przykład: baza danych uniwersytetu

Baza służy do przechowywania list studentów zapisanych na kursy i ocen końcowych

Po przeanalizowaniu reguł mini-świata i potrzeb użytkowników ustalono wymagania:

1. Uniwersytet jest podzielony na instytuty (*INSTYTUT*)

- każdy ma unikatową nazwę (*INazwa*), sekretariat (*ISekretariat*), nr tel (*ITelefon*) i osobę z kadry naukowej przypisaną do roli dziekana
- każdy instytut zarządza zbiorem wydziałów (*WYDZIAŁ*)
 - każdy ma unikatową nazwę (*WNazwa*), unikatowy numer (*WNumer*), sekretariat (*WSekretariat*), tel (*WTelefon*) i dyrektora –zapamiętywana data początkowa (*WDataObjęcia*) od kiedy pełni funkcję dyr. instytutu

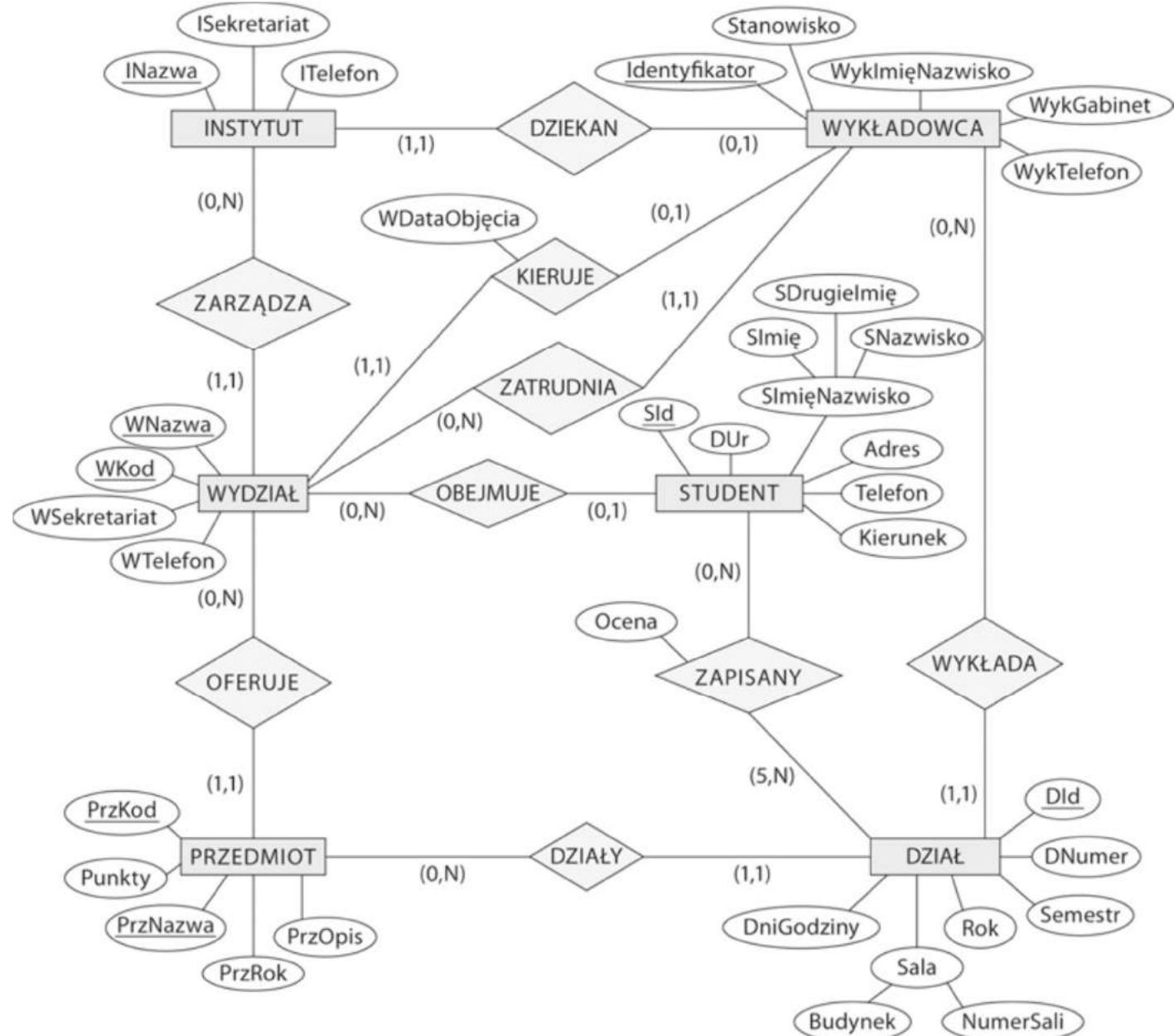
Przykład: baza danych uniwersytetu c.d.

2. Wydział oferuje zestaw przedmiotów (*PRZEDMIOT*)
 - każdy ma unikatową nazwę (*PrzNazwa*), unikatowy numer (*PrzNumer*), rok (*PrzRok*) i opis (*PrzOpis*)
3. W bazie zapisani są wykładowcy (*WYKŁADOWCA*)
 - każdy ma unikatowy identyfikator (*Identyfikator*), imię i nazwisko (*WykImięNazwisko*), gabinet (*WykGabinet*), telefon (*WykTelefon*) i stanowisko (*Stanowisko*)
 - każdy wykładowca pracuje dla jednego z instytutów
4. Baza przechowuje dane studentów (*STUDENT*)
 - nazwisko (*SImięNazwisko*: pierwsze imię *SImię*, drugie imię *SDrugieImię*, nazwisko *SNazwisko*), identyfikator (unikatowy) (*Sid*), adres (*Adres*), telefon (*Telefon*), kod kierunkowy (*Kierunek*) i data ur. (*Dur*)
 - student jest przypisany do jednego instytutu
 - konieczność przechowywania oceny studenta z każdego działu, jaki student zakończył

Przykład: baza danych uniwersytetu c.d.

5. Przedmioty są oferowane w formie działów (*DZIAŁ*)
 - każdy dział jest powiązany z 1 kursem i 1 wykładowcą i ma unikatowy identyfikator (*DId*) na poziomie wszystkich działów, a nie tylko w ramach aktualnego semestru
 - dział ma numer (*Dnumer*), semestr (*Semestr*), rok (*Rok*), salę (*Sala*) (połączenie kodu budynku (*Budynek*) i numeru Sali (*NumerSali*)) oraz dni i godziny (*DniGodziny*)
 - baza przechowuje wszystkie działy oferowane w kilku ostatnich lat+obecną ofertę
 - baza przechowuje studentów każdego działu i ich oceny (jeżeli są dostępne) – relacja wiele do wielu między studentami a działami
 - do działów musi być przypisanych przynajmniej 5 studentów

Diagram ER bazy danych uniwersytetu



Podsumowanie

- » Wysokopoziomowe, koncepcyjne modelowanie danych w oparciu o model związków encji (ER)
- » rodzaje atrybutów:
 - proste (niepodzielne), złożone, wielowartościowe
 - stałe, pochodne
- » Pojęcia związane z poziomem schematu modelu ER
 - typy encji wraz z odpowiadającymi im zbiorami encji
 - atrybuty klucza typów encji
 - zbiory wartości (dziedziny) atrybutów
 - typy związków wraz z odpowiadającymi im zbiorami związków
 - role udziału typów encji w typach związków



AGH

Podsumowanie c..

- » Określanie ograniczeń strukturalnych dla typów związków
 - współczynnik liczności (1:1, 1:N, M:N dla związków binarnych)
 - ograniczenia udziału (pełne i częściowe)
 - alternatywnie: (min,maks)
- » Słabe typy encji i związane z nimi właścielskie typy encji, identyfikujących typów związków, atrybuty klucza częściowego
- » Typy związków trójskładnikowych i wyższych stopni
 - warunki, w jakich różnią się od związków binarnych



AGH

Podsumowanie c.d.

- » Zaprezentowane elementy modelowania związków encji (typy encji, typy związków, atrybuty, klucze, ograniczenia strukturalne) umożliwiają modelowanie wielu aplikacji baz danych
 - ale: wiele nowszych, bardziej skomplikowanych aplikacji wymaga dodatkowych elementów do precyzyjnego modelowania

Rozszerzony model związków encji

Potrzeba tworzenia bardziej dokładnych schematów BD

- » Większa precyzja reprezentowania właściwości danych i ograniczeń modelowanego mini-świata
 - istotna w przypadku nowszych zastosowań technologii BD
 - projektowanie inżynierskie i produkcja przemysłowa (CAD/CAM)
 - telekomunikacja
 - skomplikowane systemy oprogramowania
 - systemy informacji geograficznej (GIS)
 - skomplikowane wymagania

Potrzeba tworzenia bardziej dokładnych schematów BD c.d.

- » Zdefiniowano dodatkowe pojęcia *semantycznego modelowania danych*
 - związki klasy-podklasy
 - dziedziczenie typów
 - specjalizacja/generalizacja
 - różne typy ograniczeń

Podklasy, nadklasy, dziedziczenie

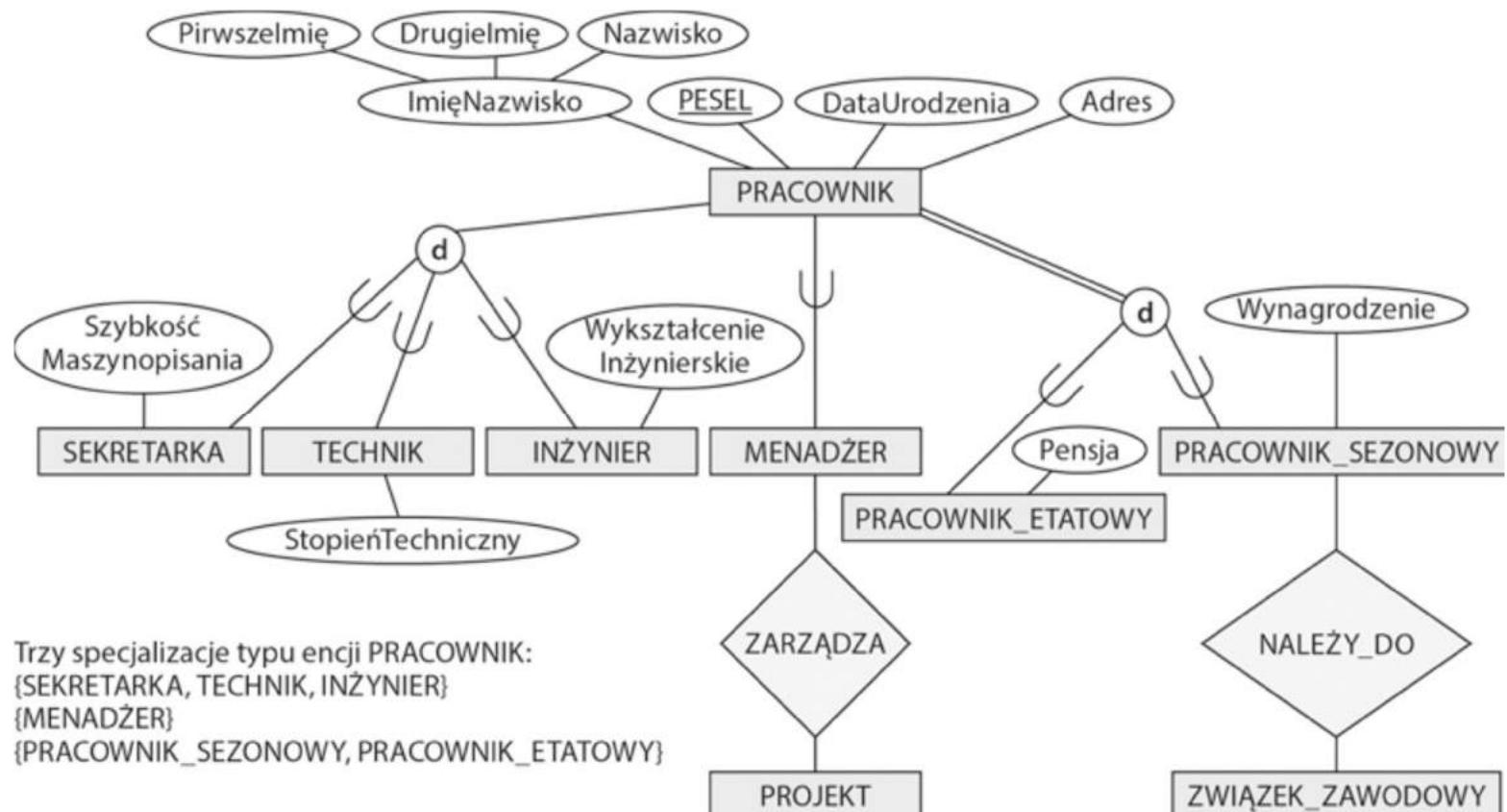
- » Brak standardowej terminologii
- » Typ encji wykorzystywany do reprezentowania
 - zarówno samego *typu encji*
 - jak i *zbioru encji określonego typu*, który istnieje w BD
 - typ encji *PRACOWNIK* opisuje
 - atrybuty i związki każdej encji reprezentującej pojedynczego pracownika
 - ale również definiuje bieżący zbiór encji typu PRACOWNIK w BD

Podklasy, nadklasy, dziedziczenie c.d.

- » W wielu przypadkach pojedynczy typ encji zawiera wiele mniejszych podgrup swoich encji, które mają określony sens i muszą być reprezentowane jawnie:
 - PRACOWNIK podzielony na SEKRETARKA, INŻYNIER, KIEROWNIK, TECHNIK, PRACOWNIK_ETATOWY, PRACOWNIK_SEZONOWY...
- » Zbiór encji w każdej z podgrup – podzbiór wszystkich encji należących do zbioru encji PRACOWNIK
 - => każda encja należąca do jednej z podgrup jednocześnie reprezentuje pracownika

Podklasy, nadklasy, dziedziczenie c.d.

- » Podgrupa = podklaśa typu encji PRACOWNIK
 - typ encji PRACOWNIK – nadklaśa dla każdej z podklaś



Podklasy, nadklasy, dziedziczenie c.d.

- » Związek między nadklassą a każdą z jej podklas:
 - związek nadklasy-podklasy (związek klasy-podklasy, *IS-A*)
- » Encja należąca do podklasy reprezentuje *ten sam byt świata rzeczywistego*, co pewna encja należąca do nadklasy
 - element składowy podklasy jest tym samym, co odpowiednia encja nadklasy, ale występuje w innej, *dokładniej sprecyzowanej roli*
- » Implementacja związku nadklasy-podklasy w BD – encje podklasy mogą być zaimplementowane w postaci osobnych rekordów powiązanych z encjami nadklasy za pomocą atrybuta klucza

Podklasy, nadklasy, dziedziczenie c.d.

- » Każda encja podklasy reprezentuje ten sam byt ze świata rzeczywistego co odpowiednia encja z nadklasy
 - encja podklasy powinna przechowywać nie tylko wartości własnych atrybutów
 - ale także wartości atrybutów pochodzących z nadklasy
- » Encja należąca do podklasy *dziedziczy* wszystkie atrybuty (+wartości) odpowiedniej encji należącej do nadklasy
 - encja podklasy dziedziczy też wszystkie związki, w których występuje nadklaśa

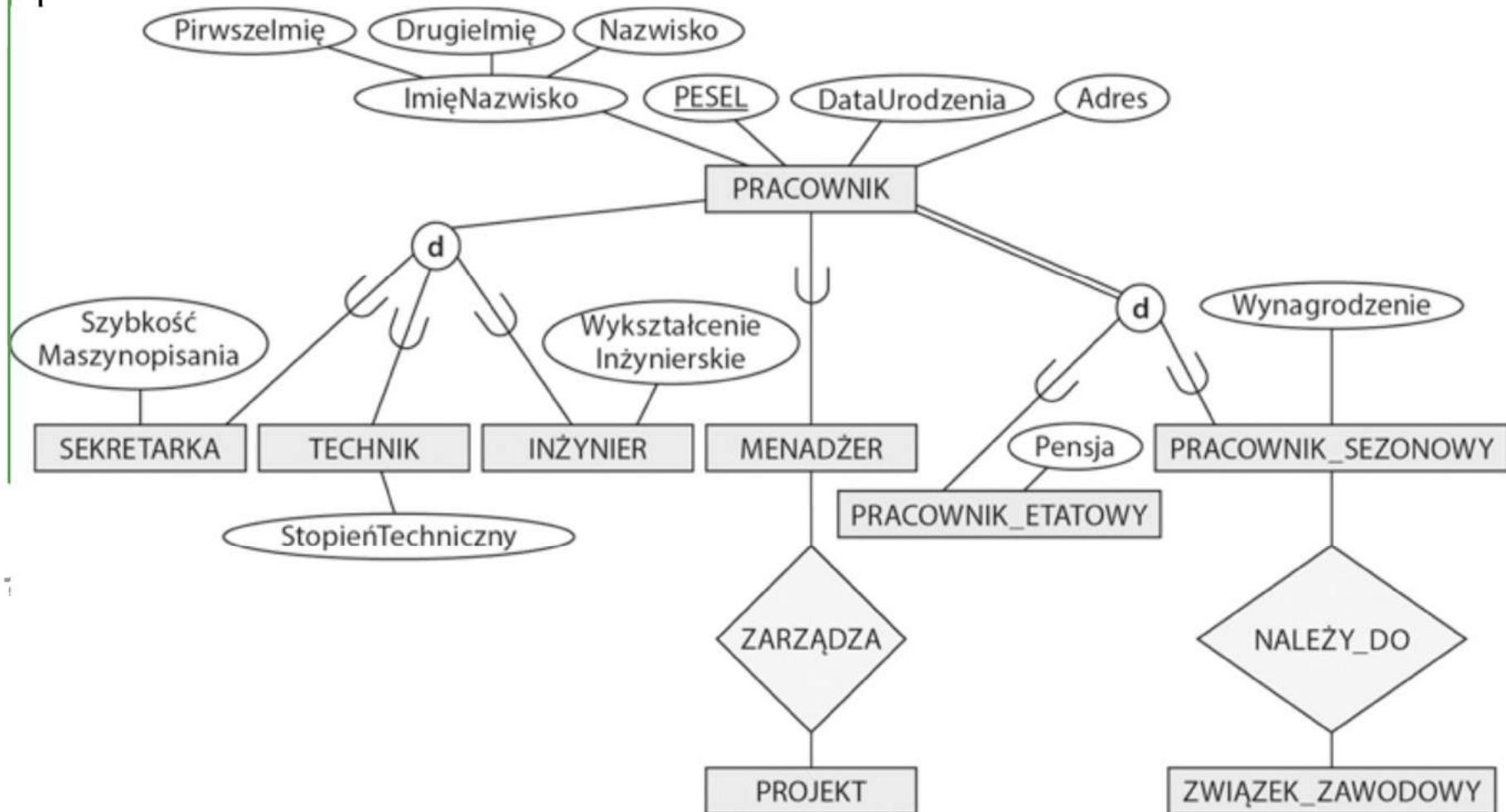


AGH

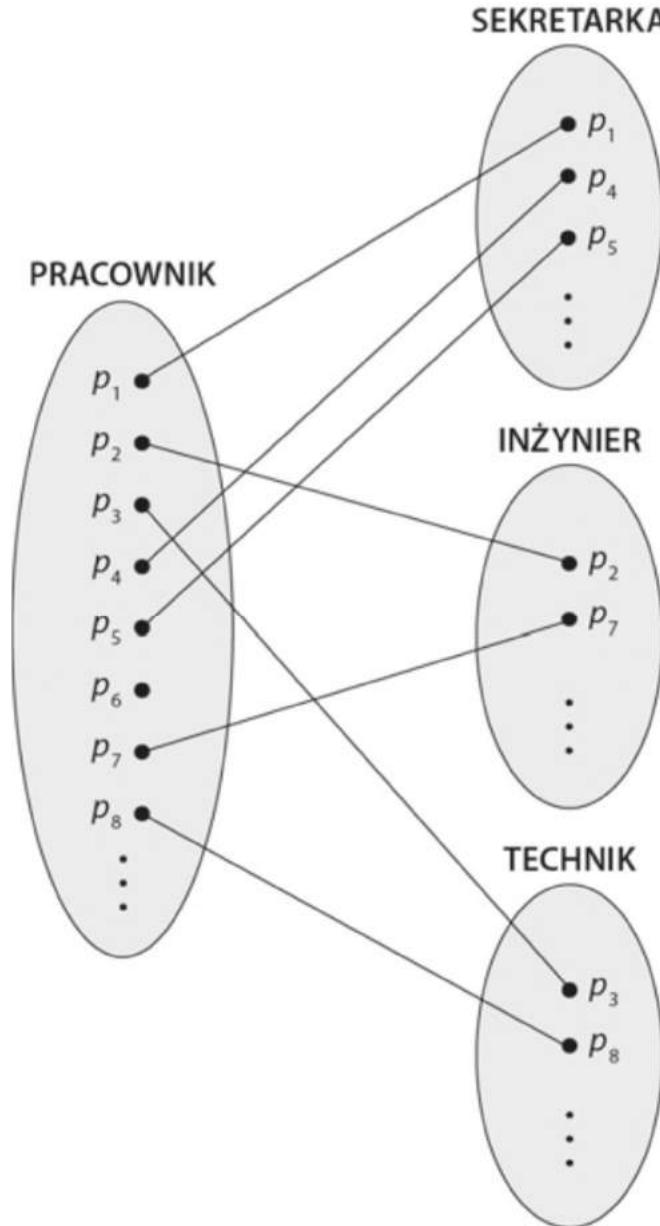
Specjalizacja

- » Proces definiowania *zbioru podklas* jednego typu encji (*nadklasa specjalizacji*)
 - zbiór podklas definiowany w oparciu o pewne wyróżniające charakterystyki encji wyodrębnione w nadklasie
 - {SEKRETARKA, INŻYNIER, TECHNIK} specjalizacja nadklasy PRACOWNIK w oparciu o *rodzaj pracy*
 - w oparciu o różne charakterystyki wyróżniające można wyznaczyć wiele specjalizacji dla tego samego typu encji:
 - {PRACOWNIK_ETATOWY, PRACOWNIK_SEZONOWY} – wg *rodzaju umowy*

Specjalizacja – graficzna reprezentacja



Egzemplarze specjalizacji

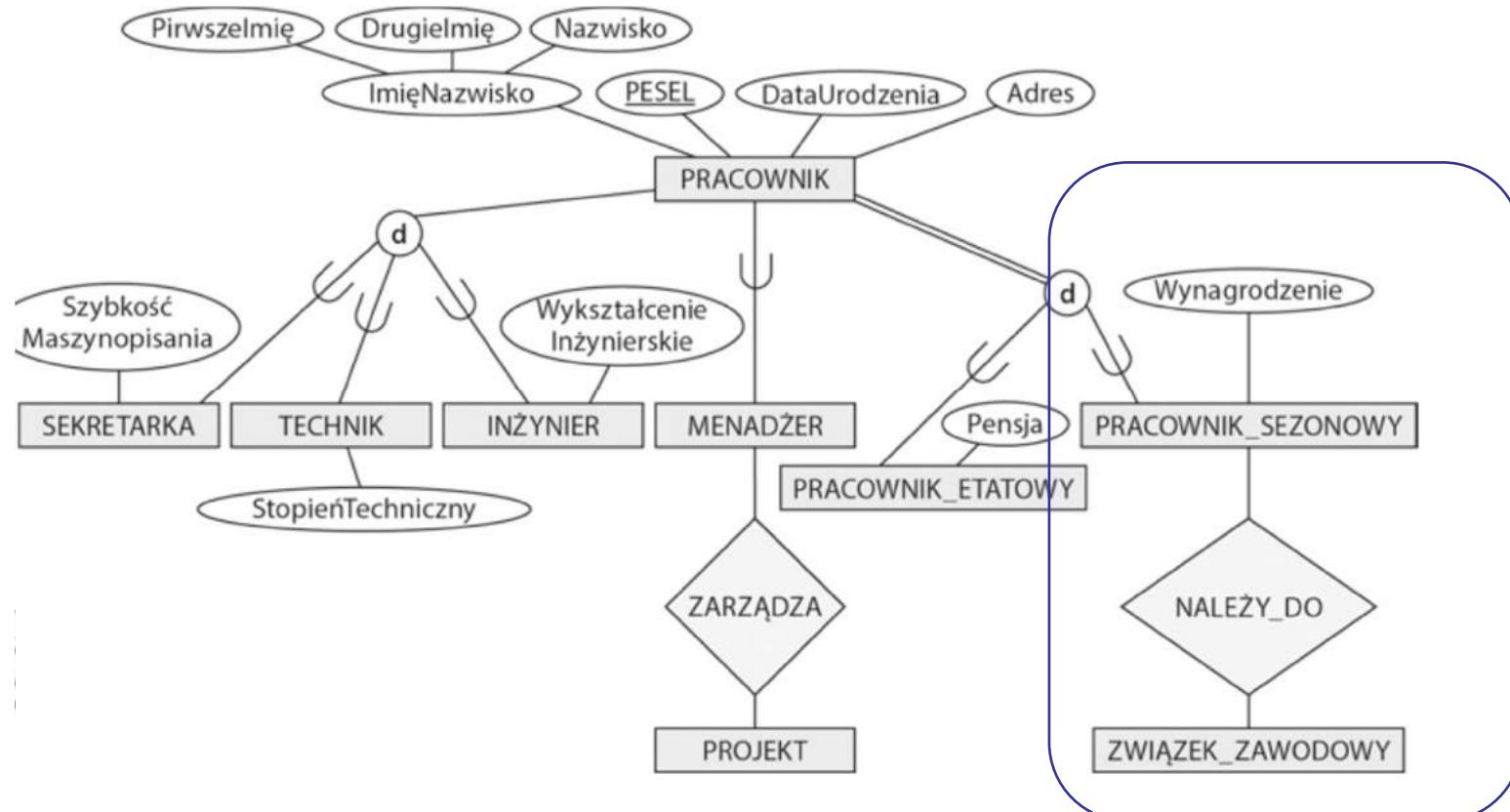


Po co związki klasy-podklasy i związki specjalizacji?

1. Pewne atrybuty mogą mieć zastosowanie tylko dla niektórych encji danej nadklasy – podklasa definiowana w celu właściwego pogrupowania tych encji, w przypadku których stosowanie określonego atrybutu/podzbioru atrybutów jest uzasadnione
 - podklasa SEKRETARKA - lokalny atrybut *SzybkośćMaszynopisania*, podklasa INŻYNIER – *WykształcenieInżynierskie* – ale ten sam zbiór atrybutów odziedziczonych z typu encji PRACOWNIK

Po co związki klasy-podklasy i związki specjalizacji? c.d.

2. W niektórych typach związków mogą występować wyłącznie encje należące do określonej podklasy

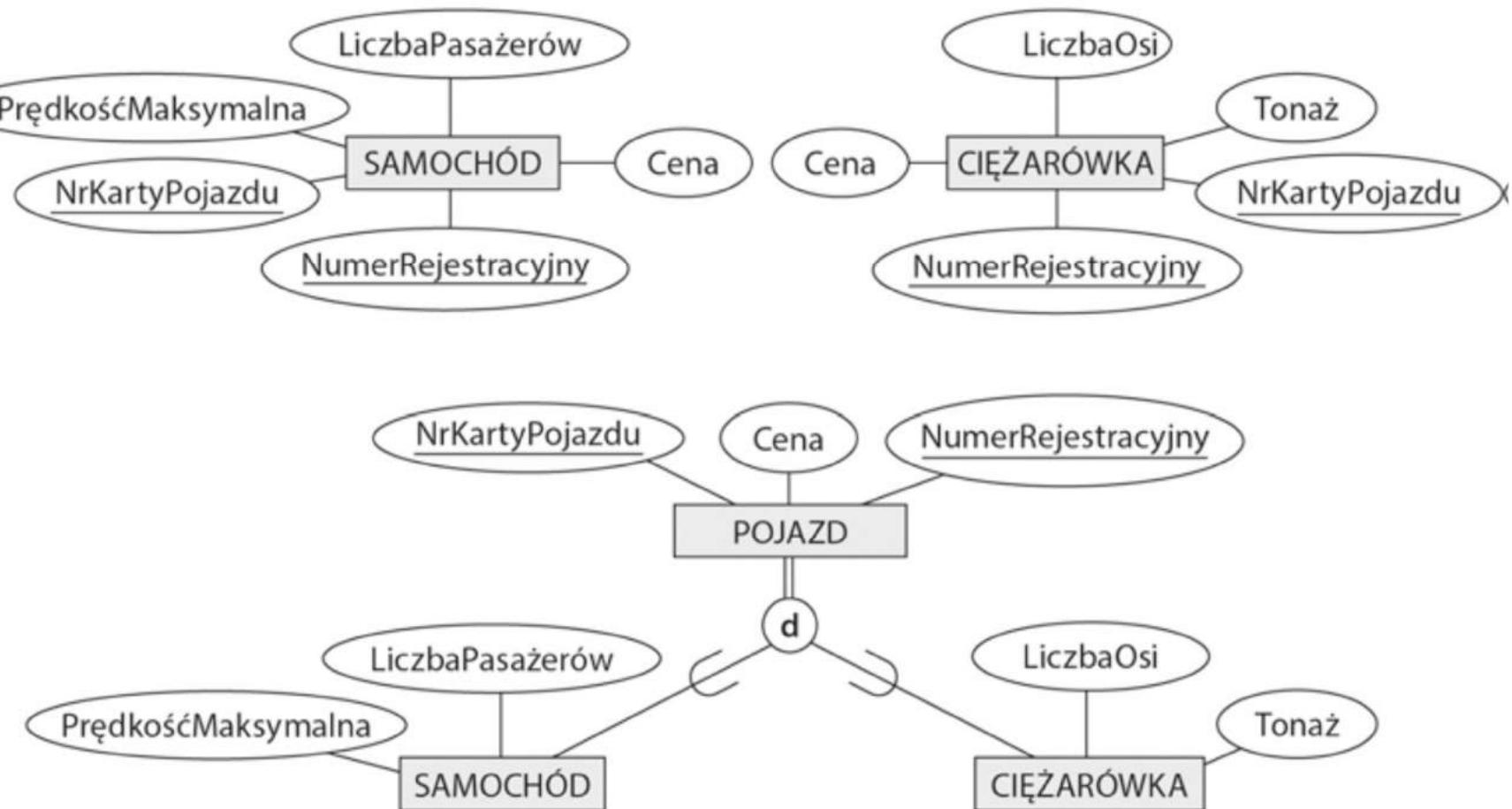


Generalizacja

» Odwrotny proces

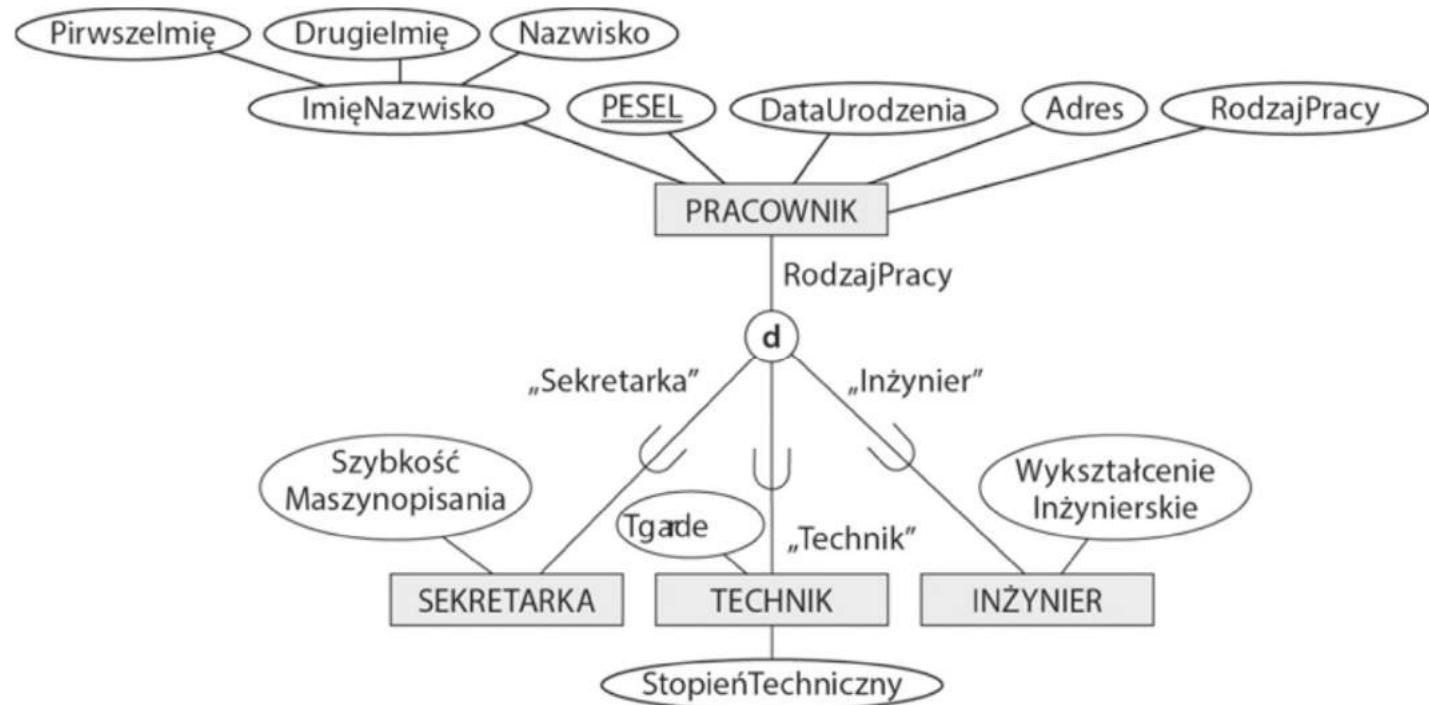
- likwidowanie różnic między różnymi typami encji
- identyfikacja ich wspólnych właściwości
- *uogólnianie (generalizacja) do postaci pojedynczej nadklasy*
 - oryginalne typy encji stanowią wyspecjalizowane podklasy

Generalizacja - przykład



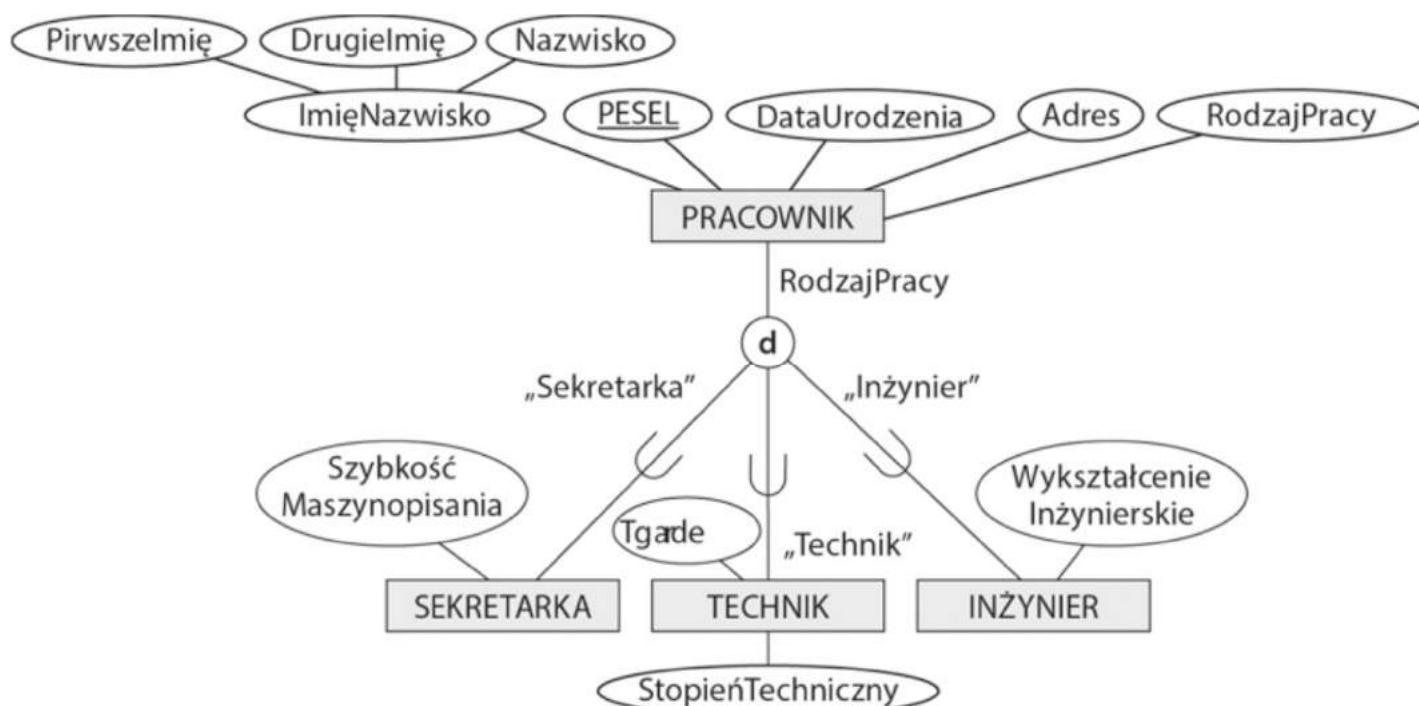
Ograniczenia dotyczące specjalizacji/generalizacji

» W przypadku niektórych specjalizacji można dokładnie określić encje, które będą należały do poszczególnych podklas przez wykorzystanie warunku dla wybranego atrybutu/grupy atrybutów nadklasy – *podklasy zdefiniowane przez predykat (warunek) - predykat definiujący podklaę*



Ograniczenia dotyczące specjalizacji/generalizacji c.d.

- » Specjalizacja *zdefiniowana przez atrybut* – wszystkie podklasy w danej specjalizacji mają zdefiniowany warunek członkostwa w oparciu o *ten sam atrybut – atrybut definiujący* dany proces specjalizacji

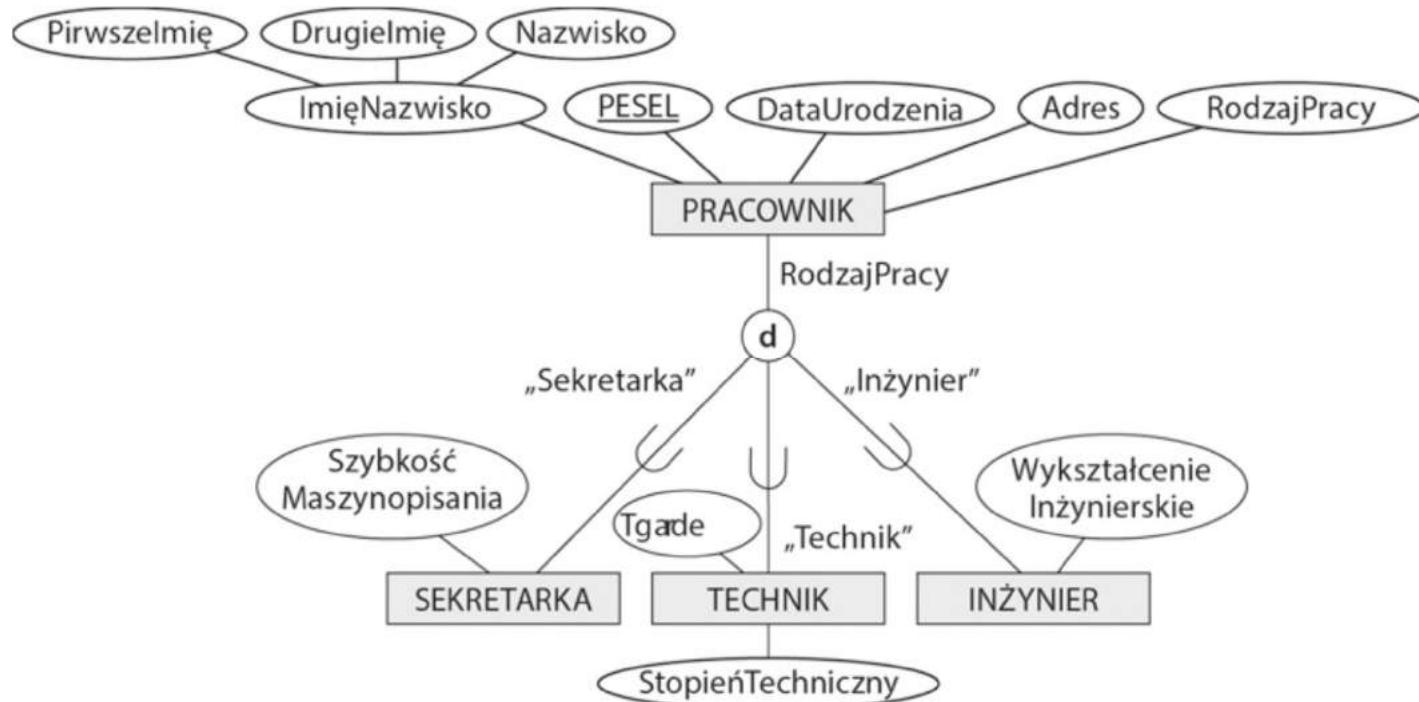


Ograniczenia dotyczące specjalizacji/generalizacji c.d.

- » Gdy nie ma zdefiniowanego warunku umożliwiającego proste przypisywanie encji do poszczególnych podklas – podklasy są *definiowane przez użytkownika*

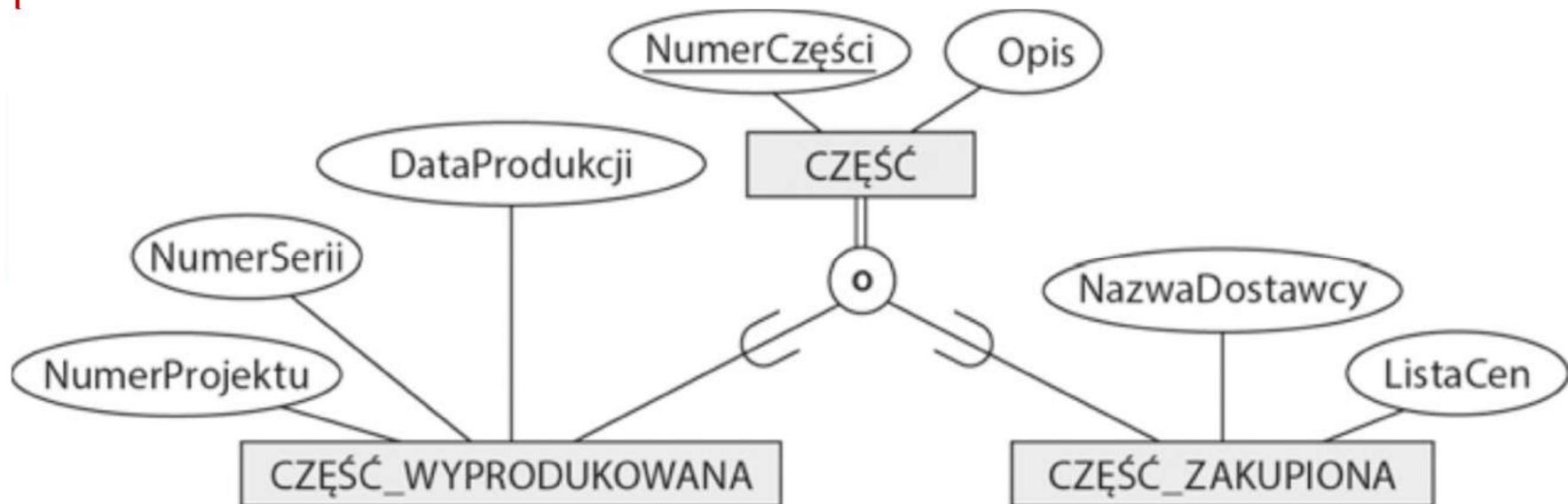
Specjalizacja – dodatkowe ograniczenia

» *Ograniczenie (wymagania) rozłączności* – wszystkie podklasy danej specjalizacji muszą być rozłączne



Specjalizacja – dodatkowe ograniczenia c.d.

- » Brak ograniczenia rozłączności – zbiory encji podklas mogłyby się częściowo lub w całości pokrywać



Specjalizacja – dodatkowe ograniczenia c.d.

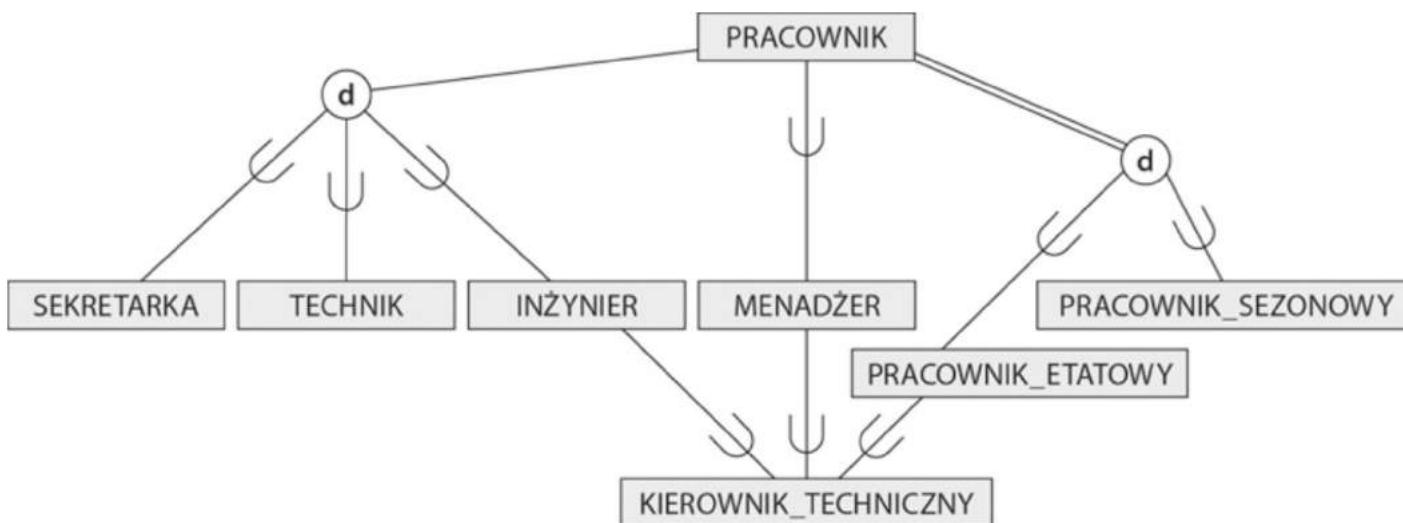
- » *Ograniczenia (wymagania) kompletności*
 - ograniczenie *kompletnej specjalizacji* – każda encja w nadklasie musi należeć przynajmniej do 1 podklasy tej specyfikacji
 - *częściowa specjalizacja* – są encje, które nie należą do żadnej z podklas
- » Ograniczenia rozłączności i kompletności są od siebie *niezależne*:
 - rozłączne, kompletne
 - rozłączne, częściowe
 - pokrywające się, kompletne
 - pokrywające się, częściowe

Reguły wstawiania/usuwania encji występujących w procesach specjalizacji/generalizacji

- » Usuwanie encji z nadklasy wiąże się z koniecznością jej automatycznego usunięcia ze wszystkich podklas, w których występuje
- » Wstawienie encji do nadklasy wiąże się z obowiązkiem jej wstawienia we wszystkich tych podklasach *zdefiniowanych przez predykat (przez atrybut)*, w przypadku których dana encja spełnia predykat definiujący
- » Wstawienie encji do nadklasy *kompletnej specjalizacji* wiąże się z koniecznością wstawienia tej encji w przynajmniej jednej z podklas tej specjalizacji
- »

Hierarchie i kraty specjalizacji/generalizacji

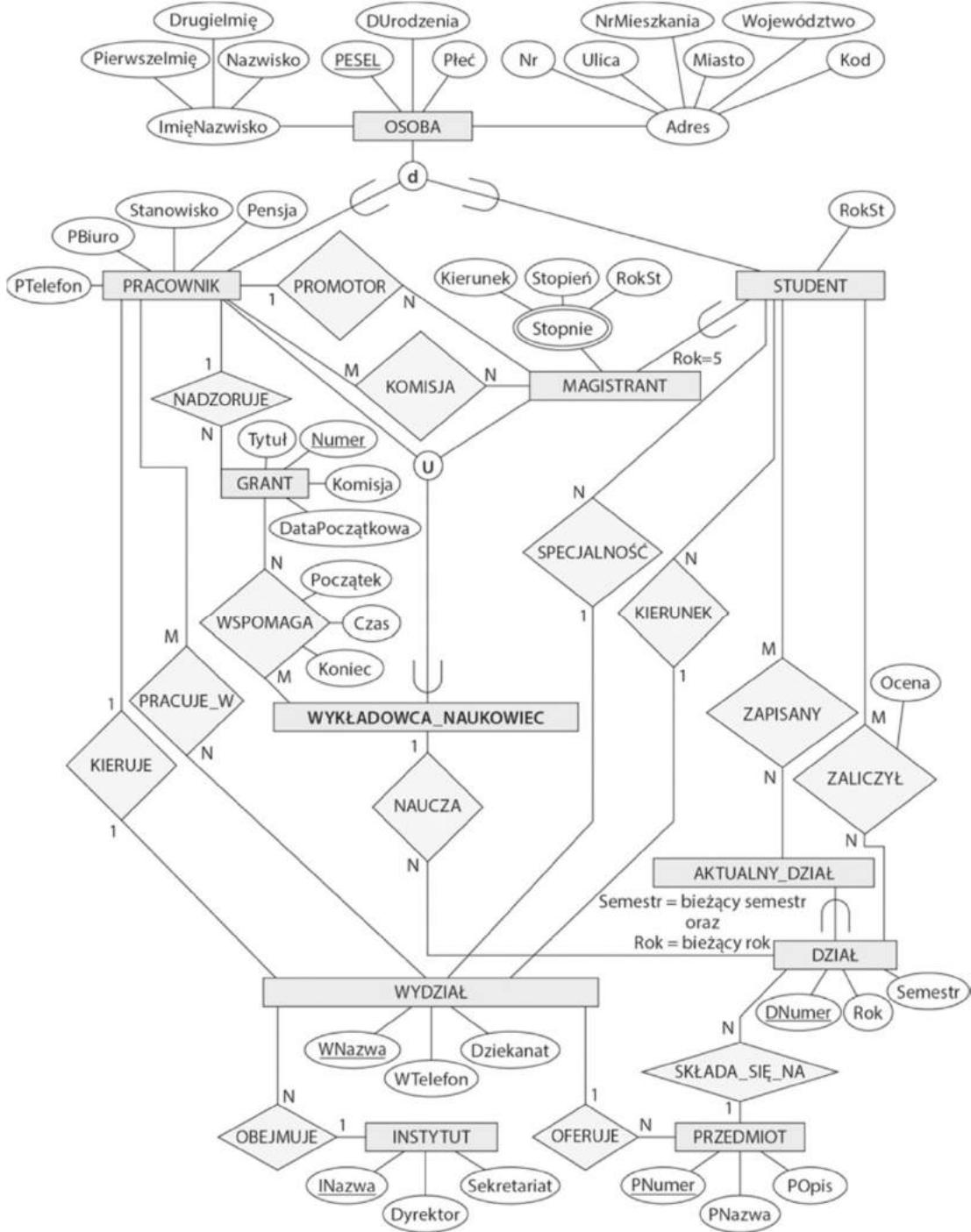
- » Podklasa może mieć zdefiniowane dalsze podklasy
 - *struktura drzewiasta* – każda podklasa może występować w tej roli tylko w jednym związku klasy-podkla
 - *krata specjalizacji* – każda podklasa może występować w roli podkla w więcej niż jednym związku klasy-podkla



Przykład schematu EER dla uniwersytetu

» Informacje o:

- studentach, wybranych przez nich kierunkach studiów, rejestrze ocen i zapisach
- przedmiotach oferowanych
- projektach badawczych prowadzonych przez kadrę naukową i studentów piątego roku



Wytyczne w procesie projektowania komponentów modelu EER

- » Uwzględniać tylko te podklasy, które są niezbędne
- » Jeżeli podklaśa ma niewiele specyficznych atrybutów i nie ma specyficznych związków – można je połączyć z nadklassą
- » Stosowanie rozłącznych/pokrywających się i pełnych/częściowych ograniczeń specjalizacji/generalizacji wynika z reguł mini-świata
 - domyślnie: ograniczenia pokrywające się i częściowe
 - bo nie nakładają żadnych ograniczeń na przynależność do podklas

Relacyjny model danych

Ograniczenia relacyjnych baz danych

Początki

- » *Edgar Frank „Ted” Codd, IBM 1970 – Nagroda Turinga*
- » Pierwsze komercyjne implementacje - wczesne lata 80-te
 - Oracle
 - SQL/DS. dla systemu operacyjnego MVS IBM
- » Najbardziej popularne RDBMS:
 - DB2 (IBM)
 - Oracle (Oracle)
 - Sybase (SAP)
 - SQL Server (Microsoft)
 - MySQL, PostgreSQL (otwarty dostęp do kodu źródłowego)

Model relacyjny - pojęcia

- » Baza danych = zbiór relacji
- » Relacja = tabela wartości
 - wiersz – zbiór powiązanych ze sobą wartości danych
 - wiersz – fakt odpowiadający bytowi (encji) lub związkowi ze świata rzeczywistego

Nazwisko	NumerIndeksu	Rok	Kierunek
Nowak	17	1	INF
Kowalski	8	2	INF

Dziedzina

» Dziedzina *D* - zbiór wartości atomowych

– określana przez wyznaczenie typu danych, z którego muszą pochodzić wszystkie wartości np.:

- *Numery_Telefonów_w_Polsce*: zb. 9-cyfrowych (2 cyfry – nr kierunkowy, 7 cyfr – nr lokalny) prawidłowych numerów telefonów w Polsce
- *Numery_PESEL*: zb. poprawnych, 11-cyfrowych numerów Pesel
- *Nazwiska* – zb. ciągów znakowych reprezentujących nazwiska
- *Średnie_ocen*: możliwe wartości reprezentujące obliczone średnie ocen; każda musi być liczbą rzeczywistą (zmiennoprzecinkową) z przedziału 2-5
- *Wiek_pracowników*: możliwe wartości reprezentujące wiek poszczególnych pracowników; liczba całkowita 15-80
- *Nazwy_wydziałów_akademickich*: zb. nazw wydziałów danej uczelni
- *Kod_wydziałów_akademickich*: zb. kodów wydziałów akademickich, np. INF

Dziedzina c.d.

- » Logiczne definicje dziedzin
- » Dla każdej dziedziny należy określić dodatkowo *typ* lub *format danych*
 - dla *Numery_Telefonów_w_Polsce*: ciąg znaków w postaci (ddd)ddd-dd-dd – każda *d* – cyfra dziesiętna, pierwsze 2 – nr kierunkowy
 - dla *Wiek_pracowników*: zbiór liczb całkowitych od 15 do 80
 - dla *Nazwy_wydziałów_akademickich*: zbiór wszystkich ciągów znaków reprezentujące prawidłowe nazwy wydziałów
- » Dziedzina opisywana za pomocą nazwy, typu danych i formatu
 - dodatkowe informacje ułatwiające interpretowanie wartości z danej dziedziny – np. jednostka

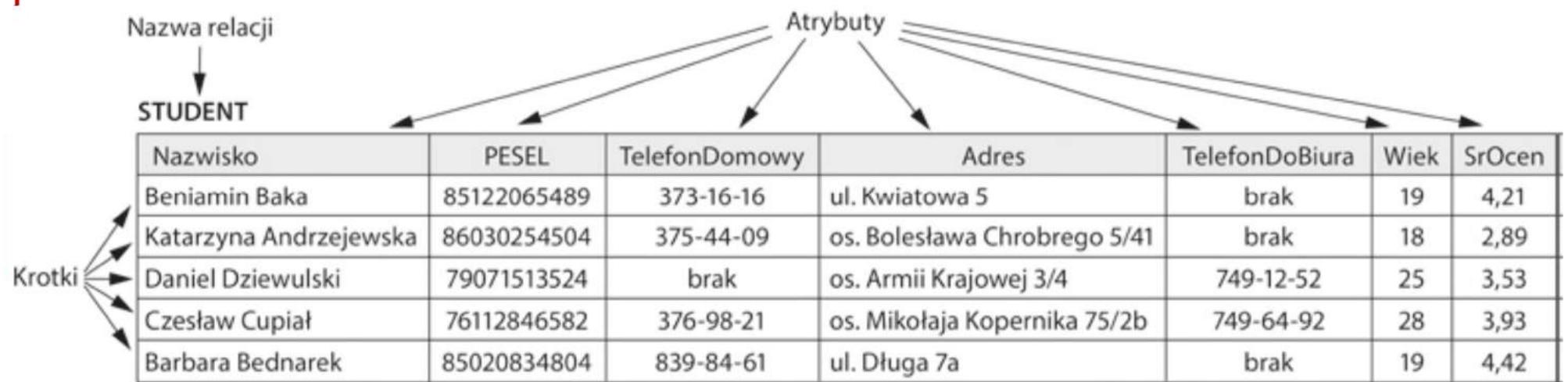
Schemat relacji

- » Schemat relacji R , $R(A_1, A_2, \dots, A_n)$
 - każdy atrybut A_i – nazwa roli odgrywanej przez określoną dziedzinę D w schemacie relacji R
 - D – dziedzina atrybutu A_i i zapisywane: $\text{dom}(A_i)$ (domain)
 - wykorzystywany do opisywania relacji; R – nazwa relacji
 - *stopień relacji* – liczba atrybutów jej schematu, czyli n
- » Relacja siódmego stopnia:
 - STUDENT(*Nazwisko*, *Pesel*, *TelefonDomowy*, *Adres*, *TelefonDoBiura*, *Wiek*, *ŚrOcen*)
 - STUDENT(*Nazwisko*: string, *Pesel* : string, *TelefonDomowy* : string, *Adres* : string, *TelefonDoBiura* : string, *Wiek* : integer, *ŚrOcen* : real)

Stan relacji

- » Stan relacji $r - r(R)$ – schematu relacji $R(A_1, A_2, \dots, A_n)$ – zbiór n -krotek:
 - $r = (t_1, t_2, \dots, t_n)$
- » Każda n -krotka t jest uporządkowaną listą n wartości:
 - $t = < v_1, v_2, \dots, v_n >$, gdzie każda wartość v_i ($1 \leq i \leq n$) jest elementem dziedziny $\text{dom}(A_i)$ lub specjalną wartością pustą
 - dla i -tej wartości w krotce t (która odpowiada atrybutowi A_i) odwołujemy się: $t[A_i]$ lub $t.A_i$ (lub $t[i]$)
 - intensja relacji dla schematu R
 - ekstensja relacji dla stanu $r(R)$

Przykład relacji





AGH

Formalna definicja relacji

- » Relacja (lub stan relacji) $r(R)$ – relacja matematyczna stopnia n zdefiniowana na dziedzinach $\text{dom}(A_1), \text{dom}(A_2), \dots, \text{dom}(A_n)$, która jest podzbiorem iloczynu kartezjańskiego dziedzin definiujących R
 - $r(R) \subseteq (\text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n))$
- » Łączna liczba możliwych krotek, które mogą istnieć w stanie relacji $r(R)$:
 - $|\text{dom}(A_1)| \times |\text{dom}(A_2)| \times \dots \times |\text{dom}(A_n)|$
- » Stan relacji w określonym momencie (tzw. *bieżący stan relacji*) odzwierciedla tylko 1 poprawny zbiór krotek (ze wszystkich możliwych kombinacji), który reprezentuje konkretny stan świata rzeczywistego
- » Można definiować wiele atrybutów, których wartości należą do tej samej dziedziny, ale reprezentują inne role (interpretacje) tej samej dziedziny

Właściwości relacji

- » Kolejność krotek nie jest częścią definicji relacji
- » Idenyczna relacja z inną kolejnością krotek

Nazwisko	PESEL	TelefonDomowy	Adres	TelefonDoBiura	Wiek	SrOcen
Daniel Dziewulski	79071513524	brak	os. Armii Krajowej 3/4	749-12-52	25	3,53
Barbara Bednarek	85020834804	839-84-61	ul. Długa 7a	brak	19	4,42
Czesław Cupiał	76112846582	376-98-21	os. Mikołaja Kopernika 75/2b	749-64-92	28	3,93
Katarzyna Andrzejewska	86030254504	375-44-09	os. Bolesława Chrobrego 5/41	brak	18	2,89
Beniamin Baka	85122065489	373-16-16	ul. Kwiatowa 5	brak	19	4,21

Właściwości relacji c.d.

- » *n*-krotka – uporządkowana lista *n* wartości
 - porządek wartości w krotce (kolejność atrybutów w schemacie relacji) jest istotna
- » Każda wartość w krotce – atomowa (niepodzielna na składniki)
 - nie mogą być stosowane atrybuty złożone i wielowartościowe
 - płaski model relacyjny – 1NF
 - wielowartościowe -> osobne relacje
 - złożone -> atomowe atrybuty składowe
- » Wartości puste (*null*)
 - wartości nieznane, nie mające zastosowania dla danej krotki, wartość istnieje, ale nie jest dostępna
 - najlepiej ich unikać

Ograniczenia modelu relacyjnego

- 1) Ograniczenia wynikające ze stosowanego modelu danych
 - *charakterystyczne ograniczenia oparte na modelu danych*, ograniczenia pośrednie
- 2) Ograniczenia, które mogą być bezpośrednio wyrażane w schematach danego modelu danych - definicje w DDL
 - *ograniczenia oparte na schemacie*, ograniczenia bezpośrednie
- 3) Ograniczenia, które *nie mogą* być wyrażane bezpośrednio w schematach stosowanego modelu danych – wyrażane i wymuszane na poziomie programów aplikacji
 - *ograniczenia oparte na aplikacjach, ograniczenia semantyczne, reguły biznesowe*

Zależności danych: zależności funkcyjne i zależności wielowartościowe

Ograniczenia oparte na schemacie

- » Ograniczenia dziedziny
- » Ograniczenia klucza
- » Ograniczenia wartości pustych
- » Więzy integralności encji
- » Więzy integralności odwołań (powiązań)

Ograniczenia dziedziny

- » Określają, że wewnątrz każdej z krotek zawartość każdego atrybutu A musi być atomową wartością należącą do dziedziny $\text{dom}(A)$
- » Do powiązanych z dziedzinami typów danych należą najczęściej:
 - liczby całkowite (krótkie liczby całkowite, liczby całkowite, długie liczby całkowite)
 - liczby rzeczywiste (zmiennoprzecinkowe, zmiennoprzecinkowe podwójnej precyzji)
 - znaki, wartości logiczne, ciągi znaków stałej długości, ciągi znaków zmiennej długości, datę, czas, znacznik czasu, specjalne typy danych
- » Inne możliwe dziedziny mogą być opisywane przez zakresy wartości wymienionych typów danych lub wyliczeniowe typy danych (wszystkie dopuszczalne wartości są jawnie wymienione)

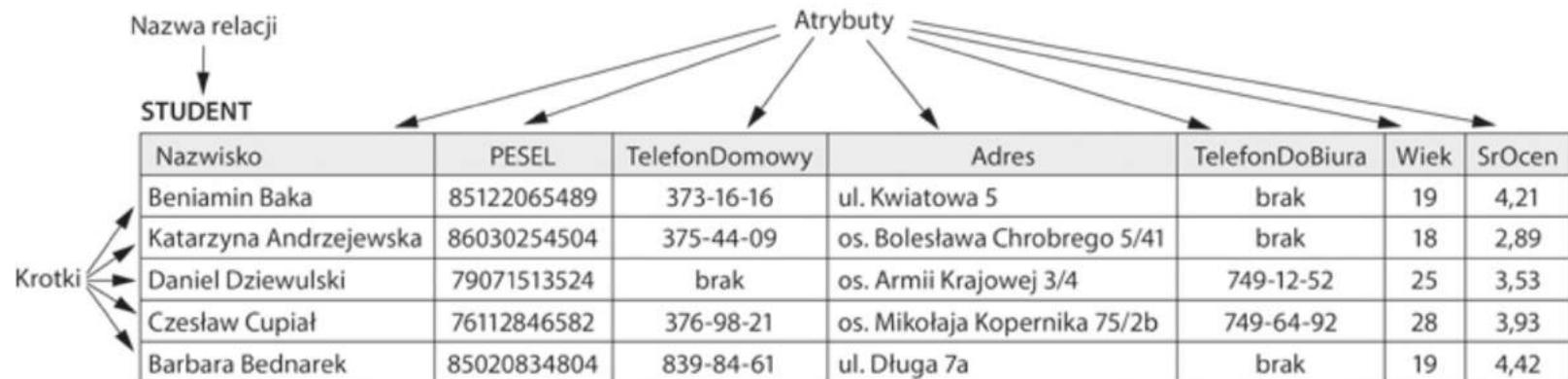
Ograniczenia klucza

- » Relacja – zbiór krotek => nie mogą istnieć dwie krotki będące taką samą kombinacją wartości we wszystkich atrybutach – SK – podzbiór atrybutów
 - $t_1[SK] \neq t_2[SK]$
 - każdy taki zbiór atrybutów SK – *nadklucz (superkey)* schematu relacji R
 - określa *ograniczenie unikatowości*: w żadnym stanie r relacji R nie mogą istnieć dwie różne krotki mające taką samą wartość nadklucza SK
 - każda relacja ma co najmniej 1 domyślny nadklucz – zbiór wszystkich swoich atrybutów
 - nadklucz może zawierać atrybuty nadmiarowe
- » Klucz K schematu relacji R jest takim nadkluczem, że:
 - usunięcie dowolnego atrybutu A z klucza K powoduje, że otrzymany atrybut K' nie jest już nadkluczem

Ograniczenia klucza c.d.

» Klucz podlega ograniczeniom:

- dwie różne krotki w żadnym stanie relacji nie mogą zawierać identycznych wartości we (wszystkich) atrybutach należących do klucza – cecha *unikatowości* dotyczy również nadklucza
- klucz jest minimalnym nadkluczem, czyli takim, z którego nie można usunąć żadnych atrybutów, aby zachować ograniczenie unikatowości – cecha *minimalności* jest wymagana dla klucza, ale opcjonalna dla nadklucza



Ograniczenia klucza c.d.

- » Schemat relacji może mieć więcej niż 1 klucz – każdy z nich *klucz kandydujący*
 - jeden z nich wyznaczany do roli *klucza głównego*
 - pozostałe klucze kandydujące – są *kluczami unikatowymi*

SAMOCHÓD

NumerRejestracyjny	NumerNadwozia	Marka	Model	Rok
DW 5345E	A69352	Ford	Mustang	02
FZ 3242B	B43696	Oldsmobile	Cutlass	05
PO 2943H	X83554	Oldsmobile	Delta	01
GDA C545	C43742	Mercedes	190-D	99
GDA E629	Y82935	Toyota	Camry	04
DW 6521N	U028365	Jaguar	XJS	04

Ograniczenia wartości pustych

- » Inne ograniczenie nakładane na atrybuty:
 - czy stosowanie wartości pustych jest dopuszczalne czy zabronione

SAMOCHÓD

NumerRejestracyjny	NumerNadwozia	Marka	Model	Rok
DW 5345E	A69352	Ford	Mustang	02
FZ 3242B	B43696	Oldsmobile	Cutlass	05
PO 2943H	X83554	Oldsmobile	Delta	01
GDA C545	C43742	Mercedes	190-D	99
GDA E629	Y82935	Toyota	Camry	04
DW 6521N	U028365	Jaguar	XJS	04

Relacyjne bazy danych i schematy relacyjnych baz danych

- » Schemat relacyjnej bazy danych S jest zbiorem schematów relacji $S = \{R_1, R_2, \dots, R_m\}$ oraz zbiorem więzów integralności WI
- » Stan relacyjnej bazy danych BD schematu S jest zbiorem stanów relacji $BD=\{r_1, r_2, \dots, r_m\}$, gdzie każde r_i jest takim stanem relacji R_i , że spełnia więzy integralności zdefiniowane w zbiorze WI



AGH PRACOWNIK

IMIĘ	INICJAŁDRIMIENIA	NAZWISKO	<u>PESEL</u>	DATAUR	ADRES	PŁEĆ	PENSJA	PESELPRZEŁOŻ	NRDZ
------	------------------	----------	--------------	--------	-------	------	--------	--------------	------

DZIAŁ

NAZWADZ	<u>NUMERDZ</u>	PESELKIEROWNIKA	DATAPRZEJKIEROWNICTWA
---------	----------------	-----------------	-----------------------

LOKALIZACJE_DZIAŁÓW

<u>NUMERDZ</u>	<u>LOKALIZACJADZ</u>
----------------	----------------------

PROJEKT

NAZWAPROJ	<u>NUMERPROJ</u>	LOKALIZACJAPROJ	NR_DZ
-----------	------------------	-----------------	-------

PRACUJE_NAD

<u>PESELPRAC</u>	<u>NR_PROJ</u>	GODZINY
------------------	----------------	---------

CZŁONEK_RODZINY

<u>PESELPRAC</u>	<u>IMIĘ CZŁONKA RODZINY</u>	PŁEĆ	DATAUR	STOPIEŃ_POKREWIEŃSTWA
------------------	-----------------------------	------	--------	-----------------------

Jeden z możliwych stanów bazy danych - przykład

PRACOWNIK

IMIE	INICJAŁDRIMIENIA	NAZWISKO	PESEL	DATAUR	ADRES	PLEC	PENSJA	PESELPRZEŁOŻ	NRDZ
Jan	B	Szewczyk	65010912345	1965-01-09	ul. Kręta 4/3, Szczecin	M	3000	55120834598	5
Franciszek	T	Wieszczycki	55120834598	1955-12-08	os. T. Kościuszki 4a/11, Szczecin	M	4000	37111045873	5
Alicja	J	Zalewska	68011932514	1968-01-19	os. Piastowskie 2/23, Świnoujście	K	2500	41062013258	4
Janina	S	Wojtczak	41062013258	1941-06-20	ul. Kwiatońska 78, Mosina	K	4300	37111045873	4
Robert	K	Napierski	62091502054	1962-09-15	ul. Podgórska 7, Police	M	3800	55120834598	5
Joanna	A	Englert	72073110039	1972-07-31	ul. Wielka 40, Szczecin	K	2500	55120834598	5
Albert	W	Janiszewski	69032923149	1969-03-29	os. Jana Pawła II 13/4, Szczecin	M	2500	41062013258	4
Józef	E	Bąk	37111045873	1937-11-10	os. Centrum 45, Szczecin	M	5500	brak	1

DZIAŁ

NAZWADZ	NUMERDZ	PESELKIEROWNIKA	DATAPRZEJKIEROWNICTWA
Badania	5	55120834598	1988-05-22
Administracja	4	41062013258	1995-01-01
Centrala	1	37111045873	1981-06-19

LOKALIZACJE_DZIAŁÓW

NUMERDZ	LOKALIZACJADZ
1	Szczecin
4	Poznań
5	Bydgoszcz
5	Świebodzin
5	Szczecin

PRACUJE_NAD

PESELPRAC	NR_PROJ	GODZINY
65010912345	1	32.5
65010912345	2	7.5
62091502054	3	40.0
72073110039	1	20.0
72073110039	2	20.0
55120834598	2	10.0
55120834598	3	10.0
55120834598	10	10.0
55120834598	20	10.0
68011932514	30	30.0
68011932514	10	10.0
69032923149	10	35.0
69032923149	30	5.0
41062013258	30	20.0
41062013258	20	15.0
37111045873	20	null

PROJEKT

NAZWAPROJ	NUMERPROJ	LOKALIZACJAPROJ	NR_DZ
ProduktX	1	Bydgoszcz	5
ProduktY	2	Świebodzin	5
ProduktZ	3	Szczecin	5
Komputeryzacja	10	Poznań	4
Reorganizacja	20	Szczecin	1
ZwiększenieZysków	30	Poznań	4

CZŁONEK_RODZINY

PESELPRAC	IMIE_CZŁONKA_RODZINY	PLEC	DATAUR	STOPIEN_POKREWIEŃSTWA
55120834598	Alicja	K	1986-04-05	CÓRKA
55120834598	Teodor	M	1983-10-25	SYN
55120834598	Janina	K	1958-05-03	ZONA
41062013258	Alojzy	M	1942-02-28	MĄZ
65010912345	Michał	M	1988-01-04	SYN
65010912345	Alicja	K	1988-12-30	CÓRKA
65010912345	Elżbieta	K	1967-05-05	ZONA

Stany bazy danych

- » Stan bazy danych, który nie spełnia więzów integralności – *stan nieprawidłowy*
- » Stan, który spełnia wszystkie więzy integralności zdefiniowane w WI, nazywany jest *stanem prawidłowym*

Nazwy atrybutów

- » Atrybuty reprezentujące w różnych relacjach to samo pojęcie ze świata rzeczywistego mogą ale nie muszą mieć takich samych nazw
- ale: atrybuty, które reprezentują różne elementy występujące w świecie rzeczywistym mogą mieć takie same nazwy w różnych relacjach

PRACOWNIK

IMIĘ	INICJAŁDRIMIENIA	NAZWISKO	PESEL	DATAUR	ADRES	PŁEĆ	PENSJA	PESELPRZEŁOŻ	NRDZ
------	------------------	----------	-------	--------	-------	------	--------	--------------	------

DZIAŁ

NAZWADZ	NUMERDZ	PESELKIEROWNIKA	DATAPRZEJKIEROWNICTWA
---------	---------	-----------------	-----------------------

LOKALIZACJE_DZIAŁÓW

NUMERDZ	LOKALIZACJADZ
---------	---------------

PROJEKT

NAZWAPROJ	NUMERPROJ	LOKALIZACJAPROJ	NR_DZ
-----------	-----------	-----------------	-------

PRACUJE_NAD

PESELPRAC	NR_PROJ	GODZINY
-----------	---------	---------

CZŁONEK_RODZINY

PESEL DRAC	IMIE_CZŁONKA_RODZINY	PŁEĆ	DATAUR	STODIENI_POKREWIEŃSTWA
------------	----------------------	------	--------	------------------------

Integralność encji, integralność odwołań i klucze obce

- » Więzy integralności encji określają: żadna wartość atrybutu pełniącego rolę klucza głównego nie może być pusta
- » Więzy integralności odwołań są definiowane pomiędzy parami relacji
 - wykorzystuje się je do utrzymywania spójności powiązanych ze sobą krotek, które należą do tych dwóch relacji

PRACOWNIK

IMIĘ	INICJAŁDRIMIENIA	NAZWISKO	PESEL	DATAUR	ADRES	PLEĆ	PENSJA	PESELPRZEŁOZ	NRDZ
Jan	B	Szewczyk	65010912345	1965-01-09	ul. Kręta 4/3, Szczecin	M	3000	55120834598	5
Franciszek	T	Wieszczycki	55120834598	1955-12-08	os. T. Kościuszki 4a/11, Szczecin	M	4000	37111045873	5
Alicja	J	Zalewska	68011932514	1968-01-19	os. Piastowskie 2/23, Swarzędz	K	2500	41062013258	4
Janina	S	Wojtczak	41062013258	1941-06-20	ul. Kwiatowa 78, Mosina	K	4300	37111045873	4
Robert	K	Napierski	62091502054	1962-09-15	ul. Podgórska 7, Police	M	3800	55120834598	5
Joanna	A	Englert	72073110039	1972-07-31	ul. Wielka 40, Szczecin	K	2500	55120834598	5
Albert	W	Janiszewski	69032923149	1969-03-29	os. Jana Pawła II 13/4, Szczecin	M	2500	41062013258	4
Józef	E	Bąk	37111045873	1937-11-10	os. Centrum 45, Szczecin	M	5500	brak	1

DZIAŁ

NAZWADZ	NUMERDZ	PESELKIEROWNIKA	DATAPRZEJKIEROWNICTWA
Badania	5	55120834598	1988-05-22
Administracja	4	41062013258	1995-01-01
Centrala	1	37111045873	1981-06-19



AGH

Klucz obcy

- » Zbiór atrybutów *FK* w schemacie relacji R_1 , jest *kluczem obcym (foreign key)*, który *odwołuje się* do relacji R_2 tylko wtedy, gdy spełnione są następujące warunki:
 - wartości atrybutów w zbiorze *FK* należą do tej samej dziedziny co atrybuty pełniące rolę klucza głównego schematu relacji R_2 : atrybuty w zbiorze *FK* odwołują się do relacji R_2 lub wskazują na tę relację
 - wartość klucza obcego (*FK*) w krotce t_1 bieżącego stanu $r_1(R_1)$ albo musi być równa wartości klucza głównego (*PK*) pewnej krotki t_2 w bieżącym stanie $r_2(R_2)$ albo musi być wartością pustą.
 - $t_1[FK]=t_2[PK]$
- » R_1 – relacja odwołująca, R_2 relacja wskazywana
- » więzy integralności odwołania relacji R_1 do R_2 są spełnione

Więzy integralności odwołania - przykład

- » Więzy integralności odwołań zwykle określane w oparciu o *powiązania istniejące między encjami reprezentowanymi przez różne schematy relacji*

PRACOWNIK

IMIĘ	INICJAŁDRIMIENIA	NAZWISKO	PESEL	DATAUR	ADRES	PŁEĆ	PENSJA	PESELPRZEŁOŻ	NRDZ
Jan	B	Szewczyk	65010912345	1965-01-09	ul. Kręta 4/3, Szczecin	M	3000	55120834598	5
Franciszek	T	Wieszczycki	55120834598	1955-12-08	os. T. Kościuszki 4a/11, Szczecin	M	4000	37111045873	5
Alicja	J	Zalewska	68011932514	1968-01-19	os. Piastowskie 2/23, Swarzędz	K	2500	41062013258	4
Janina	S	Wojtczak	41062013258	1941-06-20	ul. Kwiatowa 78, Mosina	K	4300	37111045873	4
Robert	K	Napierski	62091502054	1962-09-15	ul. Podgórska 7, Police	M	3800	55120834598	5
Joanna	A	Englert	72073110039	1972-07-31	ul. Wielka 40, Szczecin	K	2500	55120834598	5
Albert	W	Janiszewski	69032923149	1969-03-29	os. Jana Pawła II 13/4, Szczecin	M	2500	41062013258	4
Józef	E	Bąk	37111045873	1937-11-10	os. Centrum 45, Szczecin	M	5500	brak	1

DZIAŁ

NAZWADZ	NUMERDZ	PESELKIEROWNIKA	DATAPRZEJKIEROWNICTWA
Badania	5	55120834598	1988-05-22
Administracja	4	41062013258	1995-01-01
Centrala	1	37111045873	1981-06-19

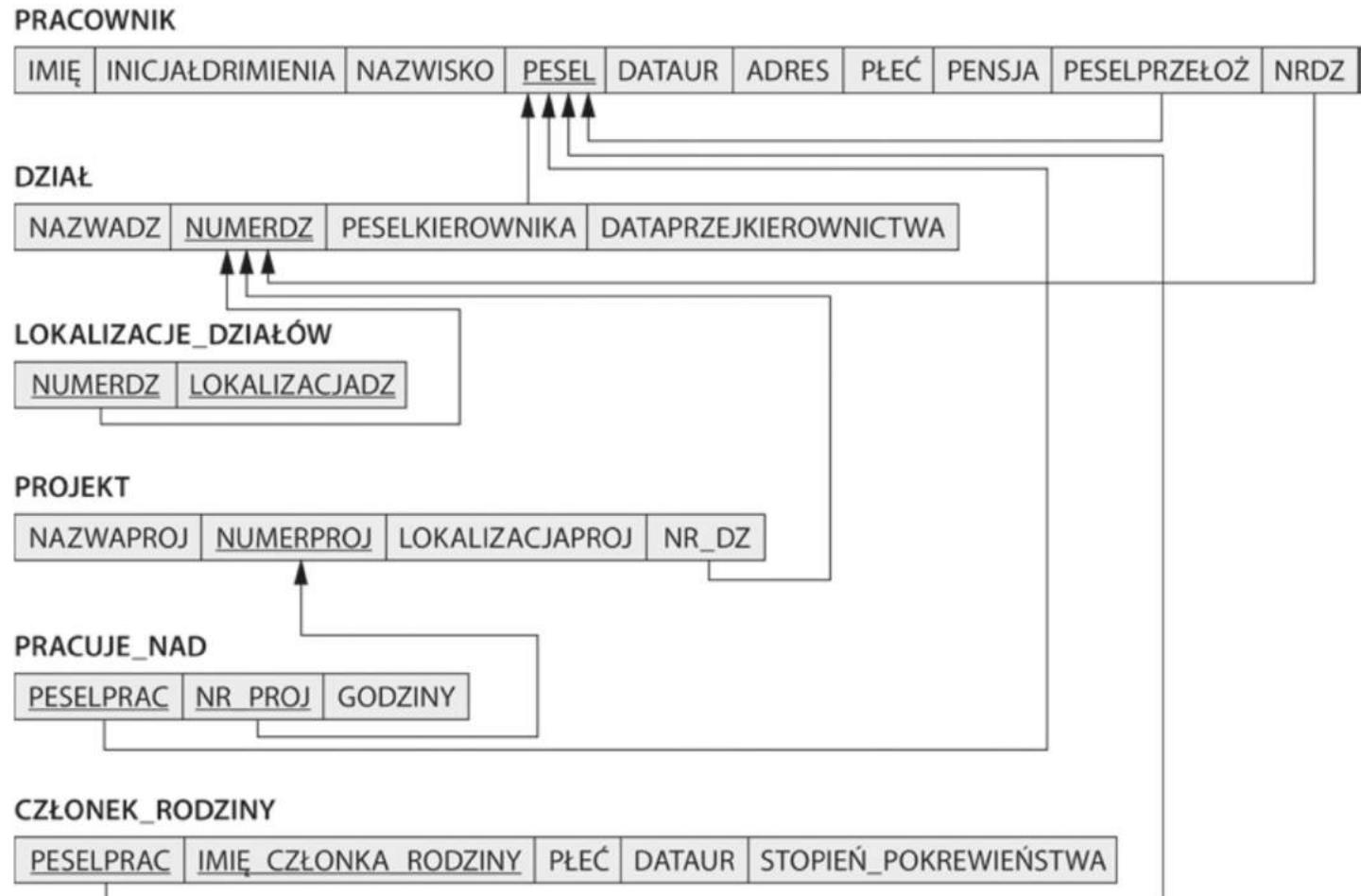
Modelowanie zależności hierarchicznej

» Klucz obcy może się odwoływać do własnej relacji

PRACOWNIK

IMIĘ	INICJAŁDRIMIENIA	NAZWISKO	PESEL	DATAUR	ADRES	PŁEĆ	PENSJA	PESELPRZEŁOŻ	NRDZ
Jan	B	Szewczyk	65010912345	1965-01-09	ul. Kręta 4/3, Szczecin	M	3000	55120834598	5
Franciszek	T	Wieszczycki	55120834598	1955-12-08	os. T. Kościuszki 4a/11, Szczecin	M	4000	37111045873	5
Alicja	J	Zalewska	68011932514	1968-01-19	os. Piastowskie 2/23, Swarzędz	K	2500	41062013258	4
Janina	S	Wojtczak	41062013258	1941-06-20	ul. Kwiatowa 78, Mosina	K	4300	37111045873	4
Robert	K	Napierski	62091502054	1962-09-15	ul. Podgórska 7, Police	M	3800	55120834598	5
Joanna	A	Englert	72073110039	1972-07-31	ul. Wielka 40, Szczecin	K	2500	55120834598	5
Albert	W	Janiszewski	69032923149	1969-03-29	os. Jana Pawła II 13/4, Szczecin	M	2500	41062013258	4
Józef	E	Bąk	37111045873	1937-11-10	os. Centrum 45, Szczecin	M	5500	brak	1

Przedstawianie więzów integralności na diagramach



Definiowanie więzów integralności

- » Wszystkie więzy integralności powinny być definiowane za poziomie schematu relacyjnej bazy danych (w jej definicji) jeżeli zgodność z nimi ma być wymuszana we wszystkich stanach tej bazy danych
 - język DDL powinien oferować środki niezbędne do określania różnego rodzaju więzów, które będą automatycznie wymuszane przez SZBD

Pozostałe typy ograniczeń

- » Ograniczenia integralności semantycznej
 - „pensja pracownika nie powinna przekraczać pensji jego bezpośredniego przełożonego”
 - „maksymalna liczba godzin poświęconych w ciągu tygodnia przez pracownika na realizację wszystkich projektów nie może przekraczać 56”
- » *Ograniczenia stanów a ograniczenia przejść*
 - „pensja pracownika może tylko rosnąć (nie może maleć)”

Operacje aktualizacji

- » Podstawowe operacje aktualizujące wykonywane na relacjach
 - *wstawianie* – dodawanie do istniejącej relacji nowej krotki/krotek
 - *usuwanie* – usuwanie krotek
 - *aktualizacja (modyfikacja)* zmiana wartości niektórych atrybutów w istniejących krotkach
- » Konieczność weryfikacji czy więzy integralności nie zostaną naruszone w wyniku tych operacji
 - ograniczenia dziedziny, ograniczenia klucza, więzy integralności encji, więzy integralności odwołań

Operacje wstawiania

- » Może naruszyć każdy z czterech rodzajów integralności i więzów:
 - *ograniczenia dziedziny* naruszone, gdy wartość atrybutu nie należy do odpowiedniego zbioru wyznaczonego do roli dziedziny wartości tego atrybutu lub jest niewłaściwego typu
 - *ograniczenia klucza* – gdy wartość klucza nowej krotki t już istnieje w innej krotce relacji $r(R)$
 - naruszenie więzów *integralności encji* może polegać na przekazaniu wartości pustej dla atrybutu klucza nowej krotki t
 - więzy *integralności odwołań* można naruszyć wstawiając taką wartość którykolwiek z klucz obcych nowej krotki t, która nie istnieje we wskazanej relacji

Operacje wstawiania - przykłady

- » <'Cecylia','Ilona','Kowalska',null,'1960-05-05','ul. Wiatraczna 3. Leśmierz',K,2800,null,4>
- » <'Alicja','Joanna','Żak','68011932514','1968-01-19','ul. Wiatraczna 3. Leśmierz',K,2800,'410620132586',4>
- » <'Cecylia','Ilona','Kowalska','60040512839','1960-04-05','os. Bajkowe 34/4h', K,280
- » <'Cecylia','Ilona','Kowalska','60040512839','1960-04-05','ul. Wiatraczna 3. Leśmierz',K,2800,null,4> 0,'410620132586,7>

PRACOWNIK

IMIE	INICJAŁDRIMIENIA	NAZWISKO	PESEL	DATAUR	ADRES	PLEC	PENSJA	PESELPRZEOŁ	NRDZ
Jan	B	Szewczyk	65010912345	1965-01-09	ul. Kręta 4/3, Szczecin	M	3000	55120834598	5
Franciszek	T	Wieszczycki	55120834598	1955-12-08	os. T. Kościuszki 4a/11, Szczecin	M	4000	37111045873	5
Alicja	J	Zalewska	68011932514	1968-01-19	os. Piastowskie 2/23, Swarzędz	K	2500	41062013258	4
Janina	S	Wojtczak	41062013258	1941-06-20	ul. Kwiatowa 78, Mosina	K	4300	37111045873	4
Robert	K	Napierski	62091502054	1962-09-15	ul. Podgóra 7, Police	M	3800	55120834598	5
Joanna	A	Englert	72073110039	1972-07-31	ul. Wielka 40, Szczecin	K	2500	55120834598	5
Albert	W	Janiszewski	69032923149	1969-03-29	os. Jana Pawła II 13/4, Szczecin	M	2500	41062013258	4
Józef	E	Bąk	37111045873	1937-11-10	os. Centrum 45, Szczecin	M	5500	brak	1

DZIAŁ

NAZWADZ	NUMERDZ	PESELKIEROWNIKA	DATAPRZEJKIEROWNICTWA
Badania	5	55120834598	1988-05-22
Administracja	4	41062013258	1995-01-01
Centrala	1	37111045873	1981-06-19

LOKALIZACJE_DZIAŁÓW

NUMERDZ	LOKALIZACJADZ
1	Szczecin
4	Poznań
5	Bydgoszcz
5	Świebodzin
5	Szczecin

PRACUJE_NAD

PESELPRAC	NR_PROJ	GODZINY
65010912345	1	32.5
65010912345	2	7.5
62091502054	3	40.0
72073110039	1	20.0
72073110039	2	20.0
55120834598	2	10.0
55120834598	3	10.0
55120834598	10	10.0
55120834598	20	10.0
68011932514	30	30.0
68011932514	10	10.0
69032923149	10	35.0
69032923149	30	5.0
41062013258	30	20.0
41062013258	20	15.0
37111045873	20	null

PROJEKT

NAZWAPROJ	NUMERPROJ	LOKALIZACJAPROJ	NR_DZ
ProduktX	1	Bydgoszcz	5
ProduktY	2	Świebodzin	5
ProduktZ	3	Szczecin	5
Komputeryzacja	10	Poznań	4
Reorganizacja	20	Szczecin	1
ZwiększenieZysków	30	Poznań	4

CZŁONEK_RODZINY

PESELPRAC	IMIE_CZŁONKA_RODZINY	PLEC	DATAUR	STOPIEŃ_POKREWIEDNISTWA
55120834598	Alicja	K	1986-04-05	CÓRKA
55120834598	Teodor	M	1983-10-25	SYN
55120834598	Janina	K	1958-05-03	ZONA
41062013258	Alojzy	M	1942-02-28	MAZ
65010912345	Michał	M	1988-01-04	SYN
65010912345	Alicja	K	1988-12-30	CÓRKA
65010912345	Elżbieta	K	1967-05-05	ZONA

Operacja wstawiania - akcja po naruszeniu ograniczeń

- » Domyślne rozwiązanie: odrzucenie
 - taka obsługa operacji przez SZBD, która pozwoli wyjaśnić użytkownikowi powody odrzucenia próby wstawienia krotki
- » Inne rozwiązanie:
 - próba *poprawienia elementu, który spowodował odrzucenie operacji*
 - np. podanie wartości atrybutu PESEL+akceptacja nowej krotki
 - np. zmiana wartości NRDZ na prawidłową/możliwość dodania nowej krotki do DZIAŁ z NRDZ=7

Operacja usuwania

- » Może naruszać wyłącznie więzy integralności odwołań w sytuacji, gdy usuwana krotka jest wskazywana przez klucze obce występujące w pozostałych krotkach
- » Usuń z PRACUJE_NAD krotkę z PESELPRAC='68011932514' oraz NR_PROJ=10
- » Usuń z PRACOWNIK krotkę z PESEL='68011932514'
- » Usuń z PRACOWNIK krotkę z PESEL='55120834598'

PRACOWNIK

IMIĘ	INICJAŁDRIMIENIA	NAZWISKO	PESEL	DATAUR	ADRES	PŁEĆ	PENSJA	PESELPRZELOŻ	NRDZ
Jan	B	Szewczyk	65010912345	1965-01-09	ul. Kręta 4/3, Szczecin	M	3000	55120834598	5
Franciszek	T	Wieszczycki	55120834598	1955-12-08	os. T. Kościuszki 4a/11, Szczecin	M	4000	37111045873	5
Alicja	J	Zalewska	68011932514	1968-01-19	os. Piastowskie 2/23, Swarzędz	K	2500	41062013258	4
Janina	S	Wojtczak	41062013258	1941-06-20	ul. Kwiatowa 78, Mosina	K	4300	37111045873	4
Robert	K	Napierski	62091502054	1962-09-15	ul. Podgórzna 7, Police	M	3800	55120834598	5
Joanna	A	Englert	72073110039	1972-07-31	ul. Wielka 40, Szczecin	K	2500	55120834598	5
Albert	W	Janiszewski	69032923149	1969-03-29	os. Jana Pawła II 13/4, Szczecin	M	2500	41062013258	4
Józef	E	Bąk	37111045873	1937-11-10	os. Centrum 45, Szczecin	M	5500	brak	1

DZIAŁ

NAZWADZ	NUMERDZ	PESELKIEROWNIKA	DATAPRZEJKIEROWNICTWA
Badania	5	55120834598	1988-05-22
Administracja	4	41062013258	1995-01-01
Centrala	1	37111045873	1981-06-19

LOKALIZACJE_DZIAŁÓW

NUMERDZ	LOKALIZACJADZ
1	Szczecin
4	Poznań
5	Bydgoszcz
5	Świebodzin
5	Szczecin

PRACUJE_NAD

PESELPRAC	NR_PROJ	GODZINY
65010912345	1	32.5
65010912345	2	7.5
62091502054	3	40.0
72073110039	1	20.0
72073110039	2	20.0
55120834598	2	10.0
55120834598	3	10.0
55120834598	10	10.0
55120834598	20	10.0
68011932514	30	30.0
68011932514	10	10.0
69032923149	10	35.0
69032923149	30	5.0
41062013258	30	20.0
41062013258	20	15.0
37111045873	20	null

PROJEKT

NAZWAPROJ	NUMERPROJ	LOKALIZACJAPROJ	NR_DZ
ProduktX	1	Bydgoszcz	5
ProduktY	2	Świebodzin	5
ProduktZ	3	Szczecin	5
Komputeryzacja	10	Poznań	4
Reorganizacja	20	Szczecin	1
ZwiększenieZysków	30	Poznań	4

CZŁONEK_RODZINY

PESELPRAC	IMIĘ_CZŁONKA_RODZINY	PŁEĆ	DATAUR	STOPIEŃ_POKREWIEŃSTWA
55120834598	Alicja	K	1986-04-05	CÓRKA
55120834598	Teodor	M	1983-10-25	SYN
55120834598	Janina	K	1958-05-03	ZONA
41062013258	Alojzy	M	1942-02-28	MĄŻ
65010912345	Michał	M	1988-01-04	SYN
65010912345	Alicja	K	1988-12-30	CÓRKA
65010912345	Elżbieta	K	1967-05-05	ZONA

Operacja usuwania - akcja po naruszeniu ograniczeń

1. Odmowa wykonania operacji usunięcia (*ograniczenie*)
2. Próba przeniesienia (propagowania) operacji usunięcia przez usunięcie wszystkich krotek, które odwołują się do tej krotki/krotek przeznaczonych do usunięcia w oryginalnej operacji (*kaskadowe przeniesienie*)
3. Zmodyfikowanie wartości atrybutu odwołującego, który jest źródłem naruszenia wiązów integralności (*ustawienie wartości pustej lub domyślnej*)
 - » możliwość łączenia rozwiązań

Operacja aktualizacji

- » Aktualizuj atrybut PENSJA krotki relacji PRACOWNIK z wartością PESEL='68011932514' do wartości 2800
- » Aktualizuj atrybut NRDZ krotki relacji PRACOWNIK z wartością PESEL='68011932514' do wartości 1
- » Aktualizuj atrybut NRDZ krotki relacji PRACOWNIK z wartością PESEL='68011932514' do wartości 7
- » Aktualizuj atrybut PENSJA krotki relacji PRACOWNIK z wartością PESEL='68011932514' do wartości '410620132586'

PRACOWNIK

IMIĘ	INICJAŁDRIMIENIA	NAZWISKO	PESEL	DATAUR	ADRES	PŁEĆ	PENSJA	PESELPRZEŁOŻ	NRDZ
Jan	B	Szewczyk	65010912345	1965-01-09	ul. Kręta 4/3, Szczecin	M	3000	55120834598	5
Franciszek	T	Wieszczycki	55120834598	1955-12-08	os. T. Kościuszki 4a/11, Szczecin	M	4000	37111045873	5
Alicja	J	Zalewska	68011932514	1968-01-19	os. Piastowskie 2/23, Śwarzędz	K	2500	41062013258	4
Janina	S	Wojtczak	41062013258	1941-06-20	ul. Kwiatowa 78, Mosina	K	4300	37111045873	4
Robert	K	Napierski	62091502054	1962-09-15	ul. Podgórska 7, Police	M	3800	55120834598	5
Joanna	A	Englert	72073110039	1972-07-31	ul. Wielka 40, Szczecin	K	2500	55120834598	5
Albert	W	Janiszewski	69032923149	1969-03-29	os. Jana Pawła II 13/4, Szczecin	M	2500	41062013258	4
Józef	E	Bąk	37111045873	1937-11-10	os. Centrum 45, Szczecin	M	5500	brak	1

DZIAŁ

NAZWADZ	NUMERDZ	PESELKIEROWNIKA	DATAPRZEJKIEROWNICTWA
Badania	5	55120834598	1988-05-22
Administracja	4	41062013258	1995-01-01
Centrala	1	37111045873	1981-06-19

LOKALIZACJE_DZIAŁÓW

NUMERDZ	LOKALIZACJADZ
1	Szczecin
4	Poznań
5	Bydgoszcz
5	Świebodzin
5	Szczecin

PRACUJE_NAD

PESELPRAC	NR_PROJ	GODZINY
65010912345	1	32.5
65010912345	2	7.5
62091502054	3	40.0
72073110039	1	20.0
72073110039	2	20.0
55120834598	2	10.0
55120834598	3	10.0
55120834598	10	10.0
55120834598	20	10.0
68011932514	30	30.0
68011932514	10	10.0
69032923149	10	35.0
69032923149	30	5.0
41062013258	30	20.0
41062013258	20	15.0
37111045873	20	null

PROJEKT

NAZWAPROJ	NUMERPROJ	LOKALIZACJAPROJ	NR_DZ
ProduktX	1	Bydgoszcz	5
ProduktY	2	Świebodzin	5
ProduktZ	3	Szczecin	5
Komputeryzacja	10	Poznań	4
Reorganizacja	20	Szczecin	1
ZwiększenieZysków	30	Poznań	4

CZŁONEK_RODZINY

PESELPRAC	IMIĘ_CZŁONKA_RODZINY	PŁEĆ	DATAUR	STOPIEŃ_POKREWIEŃSTWA
55120834598	Alicja	K	1986-04-05	CÓRKA
55120834598	Teodor	M	1983-10-25	SYN
55120834598	Janina	K	1958-05-03	ZONA
41062013258	Alojzy	M	1942-02-28	MĄŻ
65010912345	Michał	M	1988-01-04	SYN
65010912345	Alicja	K	1988-12-30	CÓRKA
65010912345	Elżbieta	K	1967-05-05	ZONA

Operacja aktualizacji - akcja po naruszeniu ograniczeń

- » Aktualizacja wartości atrybutu, który nie jest ani częścią klucza głównego ani częścią klucza obcego
 - SZBD musi tylko sprawdzić, czy nowa wartość ma właściwy typ danych i należy do określonej wcześniej dziedziny wartości
- » Modyfikowanie wartości klucza głównego = usunięcie krotki i wstawienie nowej
- » Modyfikowanie klucza obcego
 - SZBD musi się upewnić, że nowa wartość odwołuje się do istniejącej krotki we wskazanej relacji lub, że zmiana wprowadza wartość pustą
- » Naruszenie więzów integralności odwołań
 - podobne do obsługi naruszeń wynikających z nieprawidłowych operacji usuwania



Podstawy języka SQL

SQL - cechy

- » Jedno z głównych źródeł popularności relacyjnych baz danych
- » Jest standardem – łatwość wdrożenia innego RDBMS
 - ale: wiele różnic między różnymi SZBD
 - użycie standardowych konstrukcji
- » Możliwość pisania poleceń w aplikacji BD, które będą mogły uzyskiwać dostęp do informacji zawartych w dwóch lub więcej RDBMS bez konieczności zmiany języka bazy danych
- » Zapewnia wysokopoziomowy interfejs języka deklaratywnego
 - użytkownik określa co ma być wynikiem zapytania
 - optymalizacja oraz sposób wykonania – po stronie SZBD

Standardy SQL

- » *Structured Query Language* – strukturalny język zapytań
 - początkowo *SEQUEL* (Structured English QUERy Language)
 - IBM Research – jako część interfejsu dla Systemu R
- » ANSI + ISO
 - standardowa wersja języka SQL (ANSI 1986): SQL-86 (SQL1)
 - SQL-92 (SQL2)
 - (SQL3) SQL:1999
 - SQL:2003, SQL:2006 – dodano m.in. mechanizmy języka XML
 - SQL:2008 – dodano mechanizmy obiektowe
 - SQL:2011 – najnowsza
- » Nowe funkcje w standardzie => kilka lat na wprowadzenie do komercyjnych SZBD

Cechy SQL

- » Wszechstronny język baz danych – jednocześnie pełni rolę języka DDL i DML
 - definiowanie danych
 - tworzenie zapytań
 - aktualizacja danych
- » Standard SQL dodatkowo:
 - definiowanie perspektyw
 - określenie zabezpieczeń i metod uwierzytelniania
 - definiowanie węzłów integralności
 - sterowanie wykonaniem transakcji
 - reguły osadzania wyrażeń SQL w uniwersalnych językach programowania (Java, C/C++)

Standard SQL - części

» Od SQL:1999

- *rdzenna specyfikacja*
 - mają być implementowane przez wszystkich producentów RDBMS
- opcjonalne *pakiety (rozszerzenia)*
 - mogą być implementowane w postaci opcjonalnych modułów sprzedawanych niezależnie
 - np. data mining, dane przestrzenne, dane zależne od czasu, hurtownie danych, OLAP, dane multimedialne

Schemat i katalog

- » SQL wykorzystuje terminy *tabeli, wiersza i kolumny* (zamiast *relacji, krotki, atrybutu*)
- » Schemat SQL (SQL.2)
 - nazwa
 - identyfikator uwierzytelniania
 - deskryptory poszczególnych elementów schematu
 - tabele, typy, ograniczenia (więzy), perspektywy, dziedziny...
- » CREATE SCHEMA *Firma* AUTHORIZATION 'JNOWAK'
- » Katalog – nazwany zbiór schematów
 - INFORMATION_SCHEMA

CREATE TABLE

- » [FIRMA skrypt.docx](#)
- » W środowisku wykonywania Create Table niejawnie określa się schemat SQL, w którym nowe relacje są deklarowane:
 - CREATE TABLE *FIRMA.PRACOWNIK*
 - CREATE TABLE *PRACOWNIK*
- » Relacje *bazowe* a relacje *wirtualne*
- » Atrybuty tabeli bazowej uporządkowane w kolejności definiowania
 - wiersze (krotki) nie są uporządkowane wewnątrz relacji
- » Klucze obce
 - z użyciem odwołań cyklicznych
 - dotyczą nieutworzonej tabeli
 - można dołożyć ALTER TABLE

Typy danych w standardzie SQL

- » Numeryczne
 - INTEGER lub INT, SMALLINT
 - FLOAT lub REAL, DOUBLE PRECISION)
- » Ciągi znaków
 - CHAR(n) lub CHARACTER(n)
 - VARCHAR(n), CHAR VARYING(n) lub CHARACTER VARYING(n)
 - **CHARACTER LARGE OBJECT (CLOB)**
- » Ciągi bitowe
 - BIT(n)
 - BIT VARYING(n)
 - **BINARY LARGE OBJECT (BLOB)**

Typy danych w standardzie SQL c.d.

- » Wartości logiczne
 - reprezentuje TRUE i FALSE
 - NULL -> logika trójwartościowa -> UNKNOWN
- » Data i czas
 - DATA - YEAR, MONTH, DAY
 - TIME – HOUR, MINUTE, SECOND
 - TIME(i)
 - TIME WITH TIME ZONE

Typy danych w standardzie SQL - dodatkowe

- » **TIMESTAMP**
- » **INTERVAL**
- » **CREATE DOMAIN *PESEL_TYP* AS CHAR(11)**
 - może mieć przypisaną opcjonalną specyfikację domyślną
 - **DEFAULT**
- » **CREATE TYPE**
 - do tworzenia typów UDT (*user defined table*)

Definiowane ograniczenia i wartości domyślnych dla atrybutów

- » Ograniczenie NOT NULL
 - niejawnie nakładane na atrybuty będące częścią klucza głównego
 - można zdefiniować dla dowolnych atrybutów
- » Ograniczenie DEAFULT [DEAFULT.docx](#)
 - wartością domyślną dla wszystkich atrybutów nie objętych ograniczeniem NOT NULL jest wartość NULL
- » Ograniczenie CHECK
 - NUMERDZ INT NOT NULL CHECK (NUMERDZ > 0 and NUMERDZ <21)
 - CREATE DOMAIN D_NUM AS INTEGER
CHECK (NUM_DZ > 0 and NUM_DZ <21)

Definiowanie ograniczeń klucza i więzów integralności odwołań

- » *Relacja Dział:* NUMERDZ INT PRIMARY KEY
- » NAZWADZ VARCHAR(15) UNIQUE
- » FOREIGN KEY
 - więzy integralności odwołań mogą by naruszane podczas wstawiania/usuwania krotek i modyfikacji wartości klucza obcego/głównego
 - domyślne działanie: odmowa (opcja RESTRICT)
 - alternatywne: dołączenie klauzuli akcji wywoływanych za pośrednictwem odwołań: SET NULL, CASCADE, SET DEFAULT [DEAFULT.docx](#)

Nadawanie nazw definiowanym ograniczeniom

- » Nazwy wszystkich ograniczeń (wiązów) wewnątrz pojedynczego schematu muszą być unikatowe
- » [DEAFULT.docx](#)

Klauzula CHECK do określania ograniczeń dla krotek

» Ograniczenia krotek

- stosowane osobno dla pojedynczej krotki
- podlegają weryfikacji przy każdej próbie wstawienia/zmodyfikowania krotki
- CHECK (DATA_UTWORZENIA_DZIAŁU < DATAPRZEJKIEROWNICTWA)

Operacje porównywania z wartością NULL i logika trójwartościowa

- » Interpretacje:
 - Wartość nieznana – istniejąca, ale niepodana, nie wiadomo czy istnieje
 - data urodzenia, nr stacjonarny
 - Wartość niedostępna lub zatajona – istniejąca ale celowo niepodana
 - osoba ma telefon domowy, ale odmówiła podania
 - Brak zastosowania danego atrybutu – niedotycząca/niezdefiniowana dla tej krotki
 - WykształcenieWyższe dla osób, które nie ukończyły studiów wyższych
- » Każda wartość NULL jest traktowana jako wartość różna od wszystkich pozostałych wartości NULL
- » Wartość NULL w operacjach porównania – wynik NIEZNANY

Operacje porównywania z wartością NULL i logika trójwartościowa c.d.

» Logika trójwartościowa TRUE, FALSE, UNKNOWN

AND	TRUE	FALSE	UNKNOWN
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
UNKNOWN	UNKNOWN	FALSE	UNKNOWN

OR	TRUE	FALSE	UNKNOWN
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
UNKNOWN	TRUE	UNKNOWN	UNKNOWN

NOT	
TRUE	FALSE
FALSE	TRUE
UNKNOWN	UNKNOWN

Operacje porównywania z wartością NULL i logika trójwartościowa c.d.

» Ogólna zasada:

- w zapytaniach selekcji-projekcji-łączenia wybierane są tylko te krotki, dla których wyrażenie logiczne z klauzuli WHERE jest prawdziwe (TRUE)
- kombinacje krotek, dla których to samo wyrażenie jest fałszywe lub dla których wynik tego wyrażenia nie jest znany – nie są kwalifikowane do wyniku relacji

Inne konstrukcje języka SQL: *WITH* SQL:99

- » Zdefiniowanie tabeli używanej tylko na potrzeby konkretnego zapytania

```
WITH DUŻEDZIAŁY (NRDZ) AS
  (SELECT      NRDZ
   FROM        PRACOWNIK
   GROUP BY    NRDZ
   HAVING      COUNT (*) >5)
  SELECT      NRDZ, COUNT (*)
  FROM        PRACOWNIK
  WHERE       Pensja>40000 AND NRDZ IN DUŻEDZIAŁY
  GROUP BY    NRDZ;
```

Inne konstrukcje języka SQL: CASE SQL:99

» Jeżeli wartość powinna zależeć od określonych warunków

```
UPDATE PRACOWNIK  
SET PENSJA =  
CASE WHEN NRDZ =5 THEN PENSJA + 200  
      WHEN NRDZ =4 THEN PENSJA + 150  
      WHEN NRDZ =1 THEN PENSJA + 300  
      ELSE PENSJA + 0;
```

Wyzwalacze typu „zdarzenie, warunek, akcja”

- » Monitor informacji zawartych w BD
- » Reguła: czy pensja pracownika jest większa od wynagrodzenia jego bezpośredniego przełożonego - uruchomiana przez:
 - wstawienie nowego rekordu pracownika
 - modyfikacje pensji pracownika
 - zmianę przełożonego danej osoby

```
CREATE TRIGGER NARUSZENIE_PENSJA
BEFORE INSERT OR UPDATE OF PENSJA, PESEL_PRZEŁOŻ
ON PRACOWNIK
FOR EACH ROW
WHEN (NEW.PENSJA > (SELECT PENSJA FROM PRACOWNIK
WHERE PESEL=NEW.PESEL_PRZEŁOŻ))
POWIADOM_PRZEŁOŻ(NEW.PESEL_PRZEŁOŻ,
NEW.PESEL):
```

Wyzwalacze typu „zdarzenie, warunek, akcja” c.d.

- » *Zdarzenia* – operacje aktualizacji bezpośrednio przeprowadzane na bazie (np. wstawienie nowego rekordu pracownika, zmiana wynagrodzenia pracownika lub zmiana przełożonego)
 - BEFORE – wyzwalacza powinien zostać uruchomiony PRZED operacją wyzwalającą (AFTER)
- » *Warunek* (klauzula WHEN) – określa czy akcja reguły ma być wykonana – po zaistnieniu zdarzeniu sprawdzany jest opcjonalny warunek
 - jeżeli warunek niezdefiniowany: akcja wykonana bezwarunkowo
 - jeżeli zdefiniowany: akcja wykonana tylko wtedy gdy jest spełniony
- » *Akcja* – wyrażenie SQL, procedura składowana (POWIADOM_PRZEŁOŻ)

Wyzwalacze typu „zdarzenie, warunek, akcja” c.d.

» Zastosowania:

- utrzymywanie spójności bazy danych
- monitorowanie aktualizacji bazy
- automatyczne aktualizowanie danych pochodnych

```
<wyzwalacz> ::= CREATE TRIGGER <nazwa wyzwalacza>
(AFTER|BEFORE) <zdarzenia aktywujące> ON <nazwa tabeli>
[FOR EACH ROW]
[WHEN <warunek>]
<akcje wyzwalacza>;
<zdarzenia aktywujące> ::= <zdarzenie wyzwalacza> {OR <zdarzenie wyzwalacza>}
<zdarzenie wyzwalacza> ::= INSERT | DELETE | UPDATE [OF <nazwa kolumny> {, <nazwa
    kolumny>}]
<akcja wyzwalacza> ::= <blok PL/SQL>
```

Perspektywy (widoki)

- » Perspektywa – pojedyncza tabela wywiedziona z innych tabel (bazowe, perspektywy)
 - tabela wirtualna – ograniczenie aktualizacji
- » Mechanizm definiowania tabeli z informacjami, do których często musimy się odwoływać – nie musi fizycznie istnieć
 - zamiast wielokrotnego definiowania złączeń wielu tabel można stworzyć perspektywę zawierającą wynik takiego złączenia
 - tabele definiujące perspektywę



AGH

CREATE VIEW PRACUJE_NAD1

```
AS SELECT IMIĘ, NAZWISKO, NAZWAPROJ, GODZINY  
FROM PRACOWNIK INNER JOIN PRACUJE_NAD  
ON PRACOWNIK.PESEL=PRACUJE_NAD.PESELPRAC  
INNER JOIN PROJEKT  
ON PRACUJE_NAD.NR_PROJ=PROJEKT.NUMERPROJ
```

CREATE VIEW INFO_O_DZIAŁACH(NAZWA_DZIAŁU, LICZBA_PRAC, SUMA_PENSJI)

```
AS SELECT NAZWADZ, COUNT(*), SUM(PENSJA)  
FROM DZIAŁ INNER JOIN PRACOWNIK  
ON DZIAŁ.NUMERDZ = PRACOWNIK.NRDZ  
GROUP BY NAZWADZ
```

PRACUJE_NAD1

IMIĘ	NAZWISKO	NAZWAPROJ	GODZINY
------	----------	-----------	---------

INFO_O_DZIAŁACH

NAZWA_DZIAŁU	LICZBA_PRAC	SUMA_PENSJI
--------------	-------------	-------------

PRACOWNIK

IMIĘ	INICJAŁDRIMIENIA	NAZWISKO	PESEL	DATAUR	ADRES	PŁEĆ	PENSJA	PESELPRZEŁOŻ	NRDZ
------	------------------	----------	-------	--------	-------	------	--------	--------------	------

DZIAŁ

NAZWADZ	NUMERDZ	PESELKIEROWNIKA	DATAPRZEJKIEROWNICTWA
---------	---------	-----------------	-----------------------

LOKALIZACJE_DZIAŁÓW

NUMERDZ	LOKALIZACJADZ
---------	---------------

PROJEKT

NAZWAPROJ	NUMERPROJ	LOKALIZACJAPROJ	NR_DZ
-----------	-----------	-----------------	-------

PRACUJE_NAD

PESELPRAC	NR_PROJ	GODZINY
-----------	---------	---------

CZŁONEK_RODZINY

PESELPRAC	IMIĘ_CZŁONKA_RODZINY	PŁEĆ	DATAUR	STOPIEŃ_POKREWIEŃSTWA
-----------	----------------------	------	--------	-----------------------

Definiowanie perspektyw c.d.

- » Imiona i nazwiska wszystkich pracowników, którzy są bezpośrednio zaangażowani w realizację projektu 'ProjektX':

```
SELECT IMIĘ, NAZWISKO  
FROM PRACUJE_NAD1  
WHERE NAZWAPROJ='ProjektX'
```

- » Uproszczenie pewnych zapytań
- » Mechanizm zabezpieczający i ułatwiający autoryzację
- » Zapewnienie zgodności danych zawartych w perspektywie z danymi w tabelach bazowych – SZBD

```
DROP VIEW PRACUJE_NAD1
```

Implementacja perspektyw

» Dwa podejścia:

- modyfikacja zapytania

```
SELECT IMIĘ, NAZWISKO  
FROM PRACUJE_NAD1  
WHERE NAZWAPROJ='ProjektX'
```



```
SELECT IMIĘ, NAZWISKO  
FROM PRACOWNIK INNER JOIN PRACUJE_NAD  
ON PRACOWNIK.PESEL=PRACUJE_NAD.PESEL  
INNER JOIN PROJEKT ON PRACUJE_NAD.NR_PROJ=PROJEKT.NRPROJ  
WHERE NAZWAPROJ='ProjektX'
```

- materializacja perspektywy

- fizyczne tworzenie tymczasowej tabeli dla perspektywy
- konieczność zastosowania efektywnego mechanizmu automatycznego aktualizowania tabeli perspektywy
 - przyrostowa natychmiastowa aktualizacja
 - natychmiastowa/leniwa/okresowa aktualizacja

Mechanizm aktualizowania perspektyw

- » Perspektywa zbudowana w oparciu o dane zawarte w pojedynczej tabeli bazowej może być aktualizowana, jeżeli jej atrybuty obejmują PK relacji bazowej oraz zawiera wszystkie atrybuty zadeklarowane z ograniczeniem NOT NULL, które nie mają określonych wartości domyślnych
- » Perspektywy zdefiniowana na wielu tabelach stanowiące ich złączenie nie mogą być aktualizowane
- » Perspektywy zdefiniowane z wykorzystaniem grupowania lub funkcji agregujących nie mogą być aktualizowane
- » Klauzula WITH CHECK OPTION



AGH

Perspektywy jako mechanizm uwierzytelniania

- » Zastosowanie perspektyw do ukrywania określonych atrybutów lub krotek przed nieuprawnionymi użytkownikami
- » Użytkownik może zobaczyć tylko informacje o osobach pracujących dla działu 5

```
CREATE VIEW PRACWYD5 AS  
SELECT *  
FROM PRACOWNIK  
WHERE NRDZ = 5
```

- » Widoczne mogą być tylko imię, nazwisko i adres pracownika

```
CREATE VIEW PODSTAWOWE_DANE_PRAC AS  
SELECT NAZWISKO, IMIĘ, ADRES  
FROM PRACOWNIK
```



AGH DROP

Polecenia zmiany schematu

- DROP SCHEMA FIRMA CASCADE|RESTRICT
- DROP TABLE CZŁONEK_RODZINY CASCADE|RESTRICT
- » ALTER
 - dodawanie/usuwanie kolumn, dodawanie/usuwanie ograniczeń
 - ALTER TABLE FIRMA.PRACOWNIK ADD COLUMN STANOWISKO VARCHAR(12)
 - ALTER TABLE FIRMA.PRACOWNIK DROP COLUMN ADRES CASCADE
 - ALTER TABLE FIRMA.DZIAŁ ALTER COLUMN PESELKIEROWNIKA
DROP DEFAULT
 - ALTER TABLE FIRMA.DZIAŁ ALTER COLUMN PESELKIEROWNIKA
SET DEFAULT 55120834598
 - ALTER TABLE FIRMA.PRACOWNIK
DROP CONSTRAINT PRACPRZEŁOŻKO CASCADE



Algebra relacyjna



AGH

Algebra relacyjna

» Model danych:

- elementy odpowiadające za definiowanie struktury i ograniczeń baz danych
- zbiór operacji umożliwiających manipulowanie informacjami z bazy danych
 - *algebra relacyjna – wyrażenia algebry relacyjnej*
 - wynik każdej operacji: relacja, która może być dalej manipulowana za pomocą operacji tej samej algebry

» Algebra relacyjna:

- formalna podstawa dla operacji modelu relacyjnego
- wykorzystywana podczas implementowania i optymalizowania zapytań w systemach zarządzania RDBMS
- niektóre z elementów włączone do SQL

Operacje algebry relacyjnej

- » Operacje na zbiorach pochodzących z matematycznej teorii zbiorów
 - suma, część wspólna (iloczyn zbiorów), różnice zbiorów, iloczyn kartezjański
- » Operacje opracowane specjalnie dla relacyjnych baz danych
 - selekcja, projekcja, złączenia
- » Dodatkowe operacja
 - funkcje agregacji, uogólnionej projekcji

Relacyjne operacje unarne: selekcja

- » $\sigma_{\text{warunek selekcji}} (R)$
- » Pracownicy, którzy albo są zatrudnieni w dziale 4 i zarabiają miesięcznie ponad 2500, albo pracują w dziale 5 i zarabiają miesięcznie ponad 3000:
$$\sigma_{(NRDZ=4 \text{ I } PENSJA>2500) \text{ LUB } (NRDZ=5 \text{ I } PENSJA>3000)} (PRACOWNIK)$$
- » stosowany do pojedynczej relacji (unarny)
- » osobno dla każdej krotki
- » stopień relacji wynikowej taki sam jak relacji R
- » selektywność warunku – odsetek krotek wybranych przez warunek
- » przemienna

$$\sigma_{(NRDZ=4 \text{ I } PENSJA>2500)} (PRACOWNIK)$$

```
SELECT * FROM PRACOWNIK WHERE NRDZ=4 AND PENSJA > 2500
```

Relacyjne operacje unarne: projekcja

» $\pi_{<\text{lista atrybutów}>}(R)$

$\pi_{PŁEĆ, PENSJA}(\text{Pracownik})$

- » stopień relacji wynikowej=liczba atrybutów na liście
- » usuwane wszystkie powtórzenia krotek => zbiór różnych krotek
 - mechanizm eliminowania powtórzeń
 - z powtórzeniami – wielozbior – niedozwolony, ale dopuszczalny w SQL

$\pi_{PŁEĆ, PENSJA}(\text{Pracownik})$

SELECT DISTINCT PŁEĆ, PRACOWNIK
FROM PRACOWNIK

Sekwencje operacji i operacja zmiana nazwy

$$\pi_{IMIĘ, NAZWISKO, PENSJA}(\sigma_{NRDZ=5}(PRACOWNIK))$$

- » Operacje przypisania:
 - $PRAC_DZIAŁU_5 \leftarrow \sigma_{NRDZ=5}(PRACOWNIK)$
 - $WYNIK \leftarrow \pi_{IMIĘ, NAZWISKO, PENSJA}(PRAC_DZIAŁU_5)$
- » Zmiana nazwy
 - $TYMCZASOWA \leftarrow \sigma_{NRDZ=5}(PRACOWNIK)$
 - $R(IMIĘ, NAZWISKO, PENSJA) \leftarrow \pi_{IMIĘ, NAZWISKO, PENSJA}(TYMCZASOWA)$
- » $\rho_{S(B1, B2, \dots, Bn)}(R)$ lub $\rho_S(R)$ lub $\rho_{(B1, B2, \dots, Bn)}(R)$
- » SELECT P.IMIĘ as pierwsze_imię, P.NAZWISKO as nazwisko, P.PENSJA as pensja
FROM PRACOWNIK AS P
WHERE P.NRDZ=5

Operacje pochodzące z teorii zbiorów sumy, części wspólnej i różnicy

- » Między relacjami $R(A_1, A_2, \dots, A_n)$ i $S(B_1, B_2, \dots, B_n)$ istnieje zgodność *złączenia*, jeżeli obie są tego samego stopnia n oraz $\text{dom}(A_i) = \text{dom}(B_i)$ dla $1 \leq i \leq n$
- » Dla dwóch relacji R i S między którymi istnieje zgodność relacji, można zdefiniować operacje:
 - suma $R \cup S$
 - część wspólna $R \cap S$
 - różnica zbiorów $R - S$
- » SQL: UNION, INTERSECT, EXCEPT
- » dla wielozbiorów: UNION ALL, INTERSECT ALL, EXCEPT ALL

Operacja iloczynu (produkту) kartezjańskiego

- » Lista nazwisk wszystkich członków rodzin pracowników płci żeńskiej:
 - $PRAC_K \leftarrow \sigma_{PŁEĆ='K'}(PRACOWNIK)$
 - $NAZWISKA_PRAC \leftarrow \pi_{IMIĘ, NAZWISKO, PESEL}(PRAC_K)$
 - $CZŁONKOWIE_RODZINY_PRAC \leftarrow NAZWISKA_PRAC \times CZŁONEK_RODZINY$
 - $CZŁONKOWIE_RODZINY_DANEGO_PRAC \leftarrow \sigma_{PESEL=PESELPRAC}(CZŁONKOWIE_RODZINY_PRAC)$
 - $WYNIK \leftarrow \pi_{IMIĘ, NAZWISKO, IMIĘCZŁONKA_{RODZINY}}(CZŁONKOWIE_RODZINY_DANEGO_PRAC)$
- » sekwencja iloczynu kartezjańskiego i selekcji – częsta -> specjalna operacjałączenia

Binarne operacje na relacjach: złączenie

- » Operacja złączenia dla pary relacji $R(A_1, A_2, \dots, A_n)$ i $S(B_1, B_2, \dots, B_m)$:
 - $R \bowtie_{\text{warunek złączenia}} S$
 - wynik: $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$
 - po jednej krotce dla każdej kombinacji krotek relacji R i S pod warunkiem, że ta kombinacja spełnia warunek złączenia
- » Ogólna postać warunku złączenia:
 - $<\text{warunek}> | <\text{warunek}> | \dots | <\text{warunek}>$
 - gdzie każdy warunek ma postać $A_i \theta B_j$,
 - A_i – atrybut R , B_j – atrybut S , A_i i B_j należą do tej samej dziedziny
 - θ (theta) jest jednym z operatorów porównania $\{=, <, \leq, >, \geq, \neq\}$
- » Operacje złączenia z tak zdefiniowanym warunkiem złączenia – **złączenia theta**

Odmiany operacjiłączenia: równo-łączenie

- » *Operacje równo-łączenia* – jedynym operatorem porównania jest znak równości
 - relacja wynikowa zawsze zawiera jedną lub więcej par atrybutów z identycznymi wartościami we wszystkich krotkach
 - $KIEROWNIK_DZIAŁU \leftarrow DZIAŁ \bowtie_{PESELKIEROWNIKA=PESEL} (PRACOWNIK)$

MENADŻER_DZIAŁU

DNAZWA	DNUMER	PESEL MENADŻERA	...	PIMIĘ	DIMIĘ	NAZWISKO	PESEL	...
Badania	5	55120834598	...	Franciszek	Teodor	Wieszczycki	55120834598	...
Administracja	4	41062013258	...	Janina	Sylwia	Wojtczak	41062013258	...
Centrala	1	37111045873	...	Józef	Edward	Bąk	37111045873	...

Odmiany operacji złączenia: złączenie naturalne

- » *Złączenie naturalne (*)* – usuwa z relacji wynikowej nadmiarowy atrybut
 - obydwa atrybuty złączenia – taka sama nazwa
 - jest wykonywane przez porównywanie *wszystkich par atrybutów*, które w obu relacjach mają takie same nazwy i połączenie tych warunków za pomocą operatora **I**

Właściwości operacjiłączenia

- » Jeżeli relacja R zawiera n_r krotek oraz relacja S zawiera n_S krotek, relacja wynikowa operacjiłączenia $R \bowtie_{warunek\ złączenia} (S)$ będzie zawierała minimalnie 0 a maksymalnie $n_r^*n_S$ krotek
- » Oczekiwany rozmiar relacji wynikowej uzyskanej w procesiełączenia podzielony przez rozmiar maksymalny ($n_r^*n_S$) daje współczynnik zwany *selektywnościąłączenia*
- » Złączenie wewnętrzne – operacja *dopasuj i połącz* (iloczyn kartezjański+selekcja)
- » Zbiór operacji algebry relacyjnej $\{\sigma, \pi, \cup, \rho, -, \times\}$ jest *kompletny*
 - wszystkie pozostałe operacje oryginalne algebry relacyjnej można wyrazić w postaci sekwencji operacji należących to tego zbioru

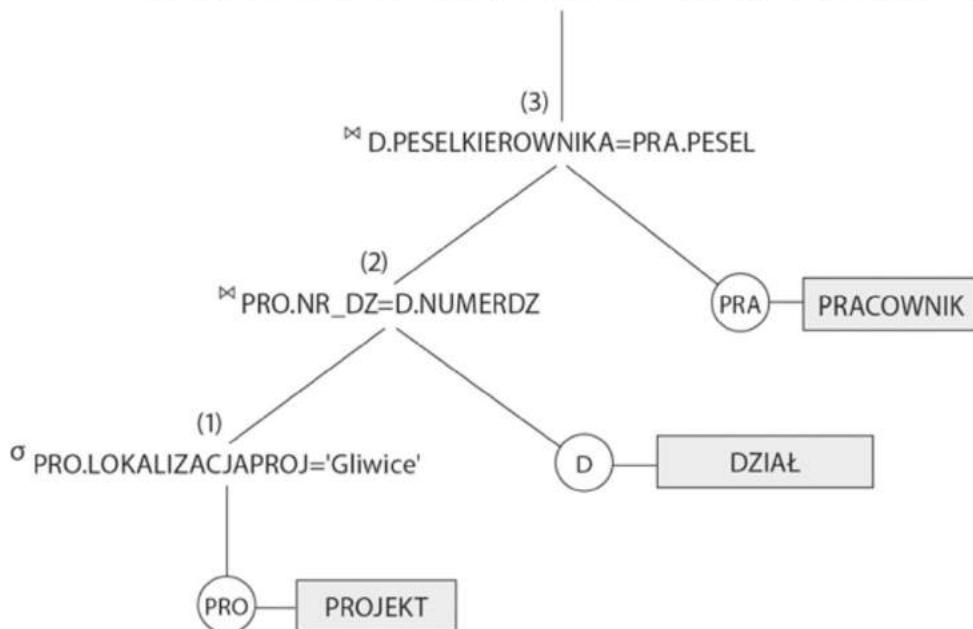
Notacja drzew zapytań

- » Notacja standardowo używana w RSZBD do wewnętrznego reprezentowania zapytań
 - notacja drzew zapytań (drzewa wykonania, przetwarzania zapytań)
 - struktura danych do zewnętrznego reprezentowania zapytań w RSZBD
- » Drzewo zapytań – drzewiasta struktura danych odpowiadająca wyrażeniom algebry relacyjnej
 - relacje wejściowe – *liście*
 - operacje algebry relacyjnej – *węzły wewnętrzne*
- » Przetwarzanie drzewa zapytań
 - wykonywanie operacji z wewnętrznych węzłów gdy dostępne są ich operandy reprezentowane przez węzły podzielne
 - węzeł wewnętrzny jest zastępowany relacją wynikową z wykonywanej operacji
 - przetwarzanie kończy się po wykonaniu operacji z korzenia i uzyskaniu wynikowej relacji dla zapytania

Notacja drzew zapytań c.d.

» Dla każdego projektu zlokalizowanego w Gliwicach podaj numer projektu, numer działu zarządzającego oraz nazwisko, adres i datę urodzenia kierownika tego działu

$\pi_{PRO.NUMERPROJ, PRO.NR_DZ, PRA.NAZWISKO, PRA.ADRES, PRA.DATAUR}$

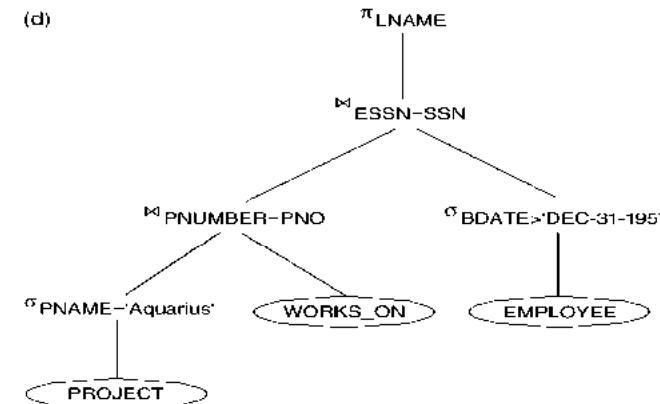
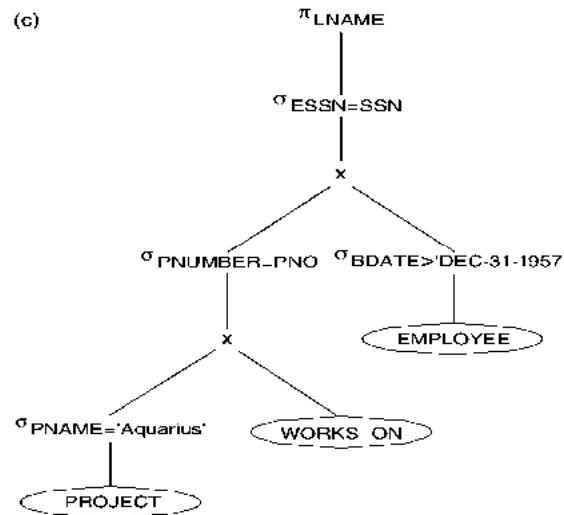


```

SELECT NUMERPROJ, NR_DZ, NAZWISKO, ADRES, DATAUR
FROM PROJEKT, DZIAŁ, PRACOWNIK
WHERE NR_DZ = NUMERDZ AND PESELKIEROWNIKA = PESEL AND
LOKALIZACJAPROJ = 'Gliwice';
    
```

$\pi_{NUMERPROJ, NR_DZ, NAZWISKO, ADRES, DATAUR}(((\sigma_{LOKALIZACJA= 'Gliwice'}(PROJEKT)) \bowtie_{NR_DZ=NUMERDZ(DZIAŁ)} \bowtie_{PESELKIEROWNIKA=PESEL(PRACOWNIK)})$

Optymalizacja planu



Normalizacja schematów relacyjnych baz danych

Poprawność schematów relacji

» Dwie płaszczyzny

- poziom logiczny (koncepcyjny)
 - interpretacja schematów relacji, znaczenie atrybutów
- poziom realizacji (składowania)
 - sposób przechowywania i aktualizowania krotek relacji

Metody postępowania przy projektowaniu BD

- » Wstępująca metodologia projektowania (*bottom-up*, projektowanie przez syntezę)
 - punkt wyjścia: podstawowe związki występujące między pojedynczymi atrybutami i używa ich w celu konstruowania schematów relacji
- » Zstępująca metodologia projektowania (*top-down*, projektowanie przez analizę)
 - jako punkt wyjścia przyjmuje pewną liczbę pogrupowanych atrybutów w formie relacji, które w naturalny sposób istnieją obok siebie
 - relacje są badane pojedynczo i zbiorczo, co prowadzi do dalszej dekompozycji projektu, do momentu, aż zostaną spełnione określone wymagania.

Cel projektowania

- » Powstanie zbioru relacji
 - zachowanie informacji
 - minimalizowanie nadmiarowości
- » Proces normalizacji
 - analizowanie relacji w celu zapewnienia spełnienia wymagań coraz bardziej rygorystycznych postaci normalnych i ich dekompozycja

Nieformalne wskazówki dotyczące projektowania schematów relacji

Miary jakości projektu schematu relacji:

1. Semantyka atrybutów w schemacie jest jasna
2. Eliminacja nadmiarowych informacji w krotkach
3. Eliminacja wartości *null* w krotkach
4. Uniemożliwienie generowania fałszywych krotek

Ad 1. Upewnienie się, że semantyka atrybutów w schemacie jest jasna

- » **Wskazówka 1:** Schemat relacji należy zaprojektować tak, aby można było w prosty sposób objąć jego znaczenie
- » Nie należy łączyć w pojedynczą relację atrybutów pochodzących z różnych typów encji i typów związków
- » Intuicyjnie: jeżeli schemat relacji odpowiada jednemu typowi encji lub jednemu typowi związku, wyjaśnienie jego znaczenia jest proste

PRACOWNIK	Klucz obcy
IMIĘNAZWPRAC PESEL DATAUR ADRES NUMERDZ	Klucz główny

DZIAŁ	Klucz obcy
NAZWADZ NUMERDZ PESELKIEROWNIKA	Klucz główny

LOKALIZACJE_DZ	Klucz obcy
NUMERDZ LOKALIZACJADZ	Klucz główny

PROJEKT	Klucz obcy
NAZWAPROJ NUMERPROJ LOKALIZACJAPROJ NRDZ	Klucz główny

PRACUJE_NAD	Klucz obcy Klucz obcy
PESEL NUMERPROJ GODZINY	Klucz główny

Stan bazy

PRACOWNIK

IMIENAZWPRAC	PESEL	DATAUR	ADRES	NUMERDZ
Szewczyk, Jan	65010912345	1965-01-09	ul. Kręta 4/3, Szczecin	5
Wieszczycki, Franciszek	55120834598	1955-12-08	os. T. Kościuszki 4a/11, Szczecin	5
Zalewska, Alicja	68011932514	1968-01-19	os. Piastowskie 2/23, Swarzędz	4
Wojtczak, Janina	41062013258	1941-06-20	ul. Kwiatowa 78, Mosina	4
Napierski, Robert	62091502054	1962-09-15	ul. Podgórska 7, Police	5
Englert, Joanna	72073110039	1972-07-31	ul. Wielka 40, Szczecin	5
Janiszewski, Albert	69032923149	1969-03-29	os. Jana Pawła II 13/4, Szczecin	4
Bąk, Józef	37111045873	1937-11-10	os. Centrum 45, Szczecin	1

DZIAŁ

NAZWADZ	NUMERDZ	PESELKIEROWNIKA
Badawczy	5	33344445555
Administracja	4	98765432109
Dyrekcja	1	88886665555

LOKALIZACJE DZ

NUMERDZ	LOKALIZACJADZ
1	Warszawa
4	Gliwice
5	Kielce
5	Kraków
5	Warszawa

PRACUJE_NAD

PESEL	NUMERPROJ	GODZINY
12345678901	1	32,5
12345678901	2	7,5
66688844444	3	40,0
45345345345	1	20,0
45345345345	2	20,0
33344445555	2	10,0
33344445555	3	10,0
33344445555	10	10,0
33344445555	20	10,0
99988887777	30	30,0
99988887777	10	10,0
98798798798	10	35,0
98798798798	30	5,0
98765432109	30	20,0
98765432109	20	15,0
88886665555	20	Null

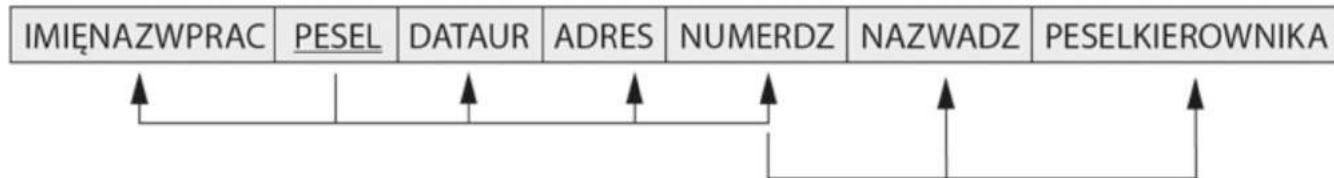
PROJEKT

NAZWAPROJ	NUMERPROJ	LOKALIZACJAPROJ	NRDZ
ProduktX	1	Kielce	5
ProduktY	2	Kraków	5
ProduktZ	3	Warszawa	5
Komputeryzacja	10	Gliwice	4
Reorganizacja	20	Warszawa	1
Nowe zyski	30	Gliwice	4

Ad. 1 - Naruszenie zasady

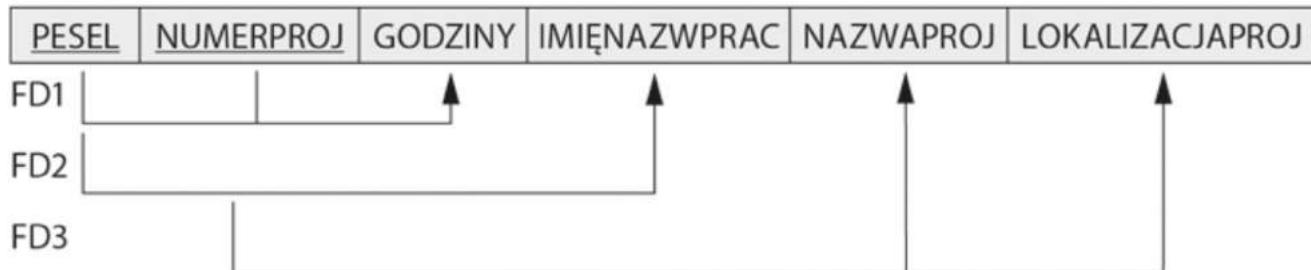
(a)

PRAC_DZ



(b)

PRAC_PROJ



Nadmiarowe informacje w krotkach

PRACOWNIK

IMIĘNAZWPRAC	PESEL	DATAUR	ADRES	NUMERDZ
Szewczyk, Jan	65010912345	1965-01-09	ul. Kręta 4/3, Szczecin	5
Wieszczycki, Franciszek	55120834598	1955-12-08	os. T. Kościuszki 4a/11, Szczecin	5
Zalewska, Alicja	68011932514	1968-01-19	os. Piastowskie 2/23, Swarzędz	4
Wojtczak, Janina	41062013258	1941-06-20	ul. Kwiatowa 78, Mosina	4
Napierski, Robert	62091502054	1962-09-15	ul. Podgórska 7, Police	5
Englert, Joanna	72073110039	1972-07-31	ul. Wielka 40, Szczecin	5
Janiszewski, Albert	69032923149	1969-03-29	os. Jana Pawła II 13/4, Szczecin	4
Bąk, Józef	37111045873	1937-11-10	os. Centrum 45, Szczecin	1

DZIAŁ

NAZWADZ	NUMERDZ	PESELKIEROWNIKA
Badawczy	5	33344445555
Administracja	4	98765432109
Dyrekcja	1	88886665555

LOKALIZACJE_DZ

NUMERDZ	LOKALIZACJADZ
1	Warszawa
4	Gliwice
5	Kielce
5	Kraków
5	Warszawa

PRACUJE_NAD

PESEL	NUMERPROJ	GODZINY
12345678901	1	32,5
12345678901	2	7,5
66688844444	3	40,0
45345345345	1	20,0
45345345345	2	20,0
33344445555	2	10,0
33344445555	3	10,0
33344445555	10	10,0
33344445555	20	10,0
99988877777	30	30,0
99988877777	10	10,0
98798798798	10	35,0
98798798798	30	5,0
98765432109	30	20,0
98765432109	20	15,0
88886665555	20	Null

PROJEKT

NAZWAPROJ	NUMERPROJ	LOKALIZACJAPROJ	NRDZ
ProduktX	1	Kielce	5
ProduktY	2	Kraków	5
ProduktZ	3	Warszawa	5
Komputeryzacja	10	Gliwice	4
Reorganizacja	20	Warszawa	1
Nowe zyski	30	Gliwice	4

Nadmiarowość

IMIĘNAZWPRAC	PESEL	DATAUR	ADRES	NUMERDZ	NAZWADZ	PESELKIEROWNIKA
Nowak, Jan	12345678901	1965-01-09	ul. Nowa 11, 00-900 Warszawa	5	Badawczy	33344445555
Kowalski, Fryderyk	33344445555	1955-12-08	Al. Niepodległości 23, 44-100 Gliwice	5	Badawczy	33344445555
Zielńska, Alicja	99988887777	1968-07-19	Pl. Wolności, 25-600 Kielce	4	Administracja	98765432109
Owiak, Maria	98765432109	1941-06-20	ul. Stara 22, 09-000 Warszawa	4	Administracja	98765432109
Głębocki, Adam	66688844444	1962-09-15	ul. Długa 34A, 70-300 Łomża	5	Badawczy	33344445555
Polakowski, Maciej	45345345345	1972-07-31	ul. Krótka 1, 50-550 Szczecin	5	Badawczy	33344445555
Kuc, Waldemar	98798798798	1969-03-29	Pl. Powstańców 10B, 00-111 Warszawa	4	Administracja	98765432109
Szymkowiak, Krystyna	88886665555	1937-11-10	ul. Szeroka 8, 26-100 Starachowice	1	Dyrekcja	88886665555

Nadmiarowość

NAZWAPROJ	PESEL	NUMERPROJ	GODZINY	IMIĘNAZWPRAC	NAZWAPROJ	LOKALIZACJAPROJ
ProduktX	12345678901	1	32,5	Nowak, Jan	ProduktX	Kielce
ProduktY	12345678901	2	7,5	Nowak, Jan	ProduktY	Kraków
ProduktZ	66688844444	3	40,0	Głębocki, Adam	ProduktZ	Warszawa
Komputeryzacja	45345345345	1	20,0	Polakowski, Maciej	ProduktX	Kielce
Komputeryzacja	45345345345	2	20,0	Polakowski, Maciej	ProduktY	Kraków
Komputeryzacja	33344445555	2	10,0	Kowalski, Fryderyk	ProduktY	Kraków
Komputeryzacja	33344445555	3	10,0	Kowalski, Fryderyk	ProduktZ	Warszawa
Komputeryzacja	33344445555	10	10,0	Kowalski, Fryderyk	Komputeryzacja	Gliwice
Reorganizacja	33344445555	20	10,0	Kowalski, Fryderyk	Reorganizacja	Warszawa
Nowe zyski	99988887777	30	30,0	Zielńska, Alicja	Nowe zyski	Gliwice
Nowe zyski	99988887777	10	10,0	Zielńska, Alicja	Komputeryzacja	Gliwice
Nowe zyski	98798798798	10	35,0	Kuc, Waldemar	Komputeryzacja	Gliwice
Nowe zyski	98798798798	30	5,0	Kuc, Waldemar	Nowe zyski	Gliwice
Nowe zyski	98765432109	30	20,0	Owiak, Maria	Nowe zyski	Gliwice
Nowe zyski	98765432109	20	15,0	Owiak, Maria	Reorganizacja	Warszawa
Nowe zyski	88886665555	20	Null	Szymkowiak, Krystyna	Reorganizacja	Warszawa

Ad 2. Nadmiarowe informacje w krotkach i anomalie aktualizacji (Codd 1972)

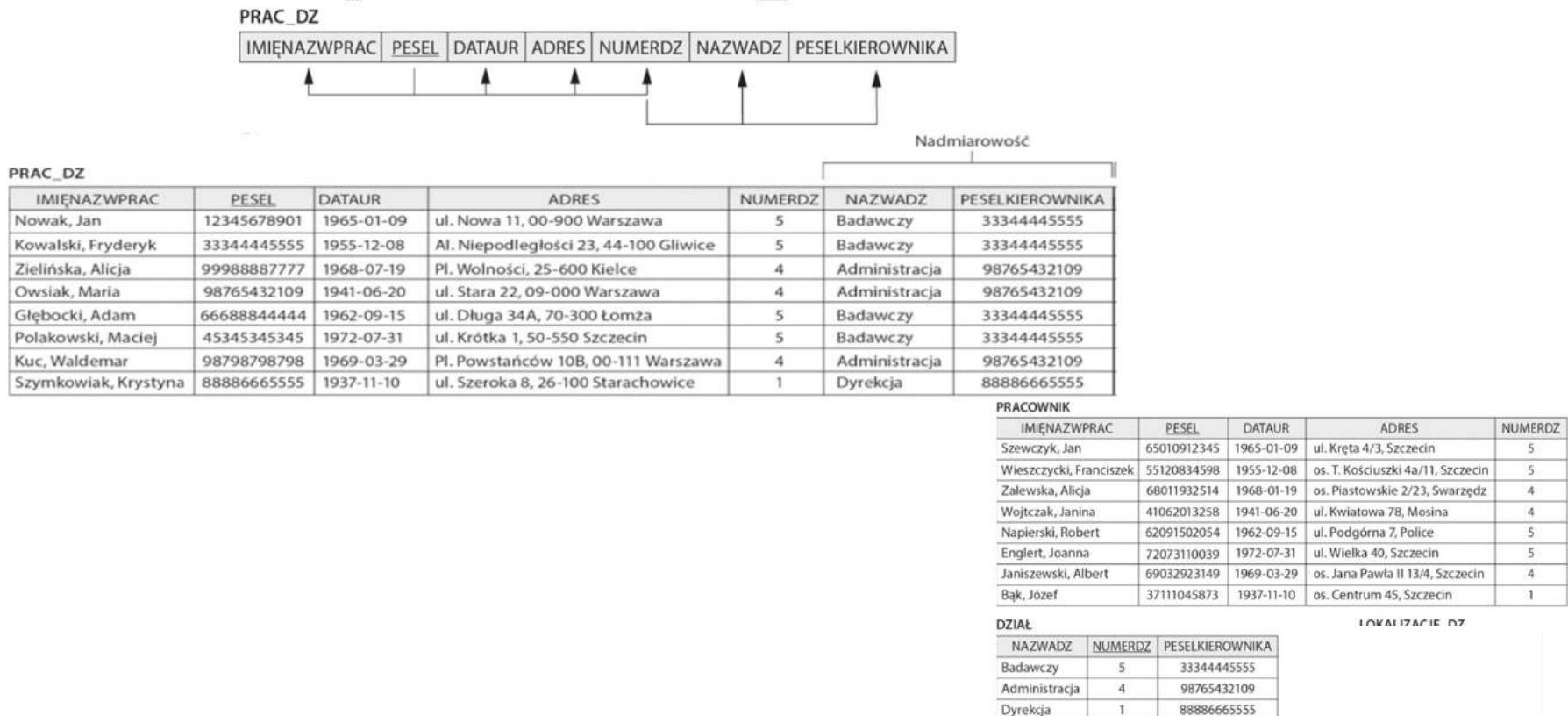
- » Anomalie wstawiania
- » Anomalie usuwania
- » Anomalie modyfikowania

Nadmiarowość						
PRAC_DZ						
IMIENAZWPRAC	PESEL	DATAUR	ADRES	NUMERDZ	NAZWADZ	PESELKIEROWNIKA
Nowak, Jan	12345678901	1965-01-09	ul. Nowa 11, 00-900 Warszawa	5	Badawczy	33344445555
Kowalski, Fryderyk	33344445555	1955-12-08	Al. Niepodległości 23, 44-100 Gliwice	5	Badawczy	33344445555
Zielńska, Alicja	99988887777	1968-07-19	Pl. Wolności, 25-600 Kielce	4	Administracja	98765432109
Owiak, Maria	98765432109	1941-06-20	ul. Stara 22, 09-000 Warszawa	4	Administracja	98765432109
Głębecki, Adam	66688844444	1962-09-15	ul. Długa 34A, 70-300 Łomża	5	Badawczy	33344445555
Polakowski, Maciej	45345345345	1972-07-31	ul. Krótka 1, 50-550 Szczecin	5	Badawczy	33344445555
Kuc, Waldemar	98798798798	1969-03-29	Pl. Powstańców 10B, 00-111 Warszawa	4	Administracja	98765432109
Szymkowiak, Krystyna	88886665555	1937-11-10	ul. Szeroka 8, 26-100 Starachowice	1	Dyrekcja	88886665555

PRAC_PROJ			Nadmiarowość	Nadmiarowość	
PESEL	NUMERPROJ	GODZINY	IMIENAZWPRAC	NAZWAPROJ	LOKALIZACJAPROJ
12345678901	1	32,5	Nowak, Jan	ProduktX	Kielce
12345678901	2	7,5	Nowak, Jan	ProduktY	Kraków
66688844444	3	40,0	Głębecki, Adam	ProduktZ	Warszawa
45345345345	1	20,0	Polakowski, Maciej	ProduktX	Kielce
45345345345	2	20,0	Polakowski, Maciej	ProduktY	Kraków
33344445555	2	10,0	Kowalski, Fryderyk	ProduktY	Kraków
33344445555	3	10,0	Kowalski, Fryderyk	ProduktZ	Warszawa
33344445555	10	10,0	Kowalski, Fryderyk	Komputeryzacja	Gliwice
33344445555	20	10,0	Kowalski, Fryderyk	Reorganizacja	Warszawa
99988887777	30	30,0	Zielńska, Alicja	Nowe zyski	Gliwice
99988887777	10	10,0	Zielńska, Alicja	Komputeryzacja	Gliwice
98798798798	10	35,0	Kuc, Waldemar	Komputeryzacja	Gliwice
98798798798	30	5,0	Kuc, Waldemar	Nowe zyski	Gliwice
98765432109	30	20,0	Owiak, Maria	Nowe zyski	Gliwice
98765432109	20	15,0	Owiak, Maria	Reorganizacja	Warszawa
88886665555	20	Null	Szymkowiak, Krystyna	Reorganizacja	Warszawa

Anomalie wstawiania

- » Wstawienie nowego pracownika do PRAC_DZ
- » Wstawienie nowego działu do PRAC_DZ



Anomalie usuwania

» Usunięcie z relacji PRAC_DZ krotkę reprezentującą ostatniego pracownika pracującego w danym dziale

PRAC_DZ

IMIĘNAZWPRAC	PESEL	DATAUR	ADRES	NUMERDZ	NAZWADZ	PESELKIEROWNIKA



PRAC_DZ

IMIĘNAZWPRAC	PESEL	DATAUR	ADRES	NUMERDZ	NAZWADZ	PESELKIEROWNIKA
Nowak, Jan	12345678901	1965-01-09	ul. Nowa 11, 00-900 Warszawa	5	Badawczy	33344445555
Kowalski, Fryderyk	33344445555	1955-12-08	Al. Niepodległości 23, 44-100 Gliwice	5	Badawczy	33344445555
Zielińska, Alicja	99988887777	1968-07-19	Pl. Wolności, 25-600 Kielce	4	Administracja	98765432109
Owiak, Maria	98765432109	1941-06-20	ul. Stara 22, 09-000 Warszawa	4	Administracja	98765432109
Głobocki, Adam	66688844444	1962-09-15	ul. Dluga 34A, 70-300 Łomża	5	Badawczy	33344445555
Polakowski, Maciej	45345345345	1972-07-31	ul. Krótka 1, 50-550 Szczecin	5	Badawczy	33344445555
Kuc, Waldemar	98798798798	1969-03-29	Pl. Powstańców 10B, 00-111 Warszawa	4	Administracja	98765432109
Szymkowiak, Krystyna	88886665555	1937-11-10	ul. Szeroka 8, 26-100 Starachowice	1	Dyrekcja	88886665555

PRACOWNIK

IMIĘNAZWPRAC	PESEL	DATAUR	ADRES	NUMERDZ
Szewczyk, Jan	65010912345	1965-01-09	ul. Kręta 4/3, Szczecin	5
Wieszczycki, Franciszek	55120834598	1955-12-08	os. T. Kościuszki 4a/11, Szczecin	5
Zalewska, Alicja	68011932514	1968-01-19	os. Piastowskie 2/23, Swarzędz	4
Wojtczak, Janina	41062013258	1941-06-20	ul. Kwiatowa 78, Mosina	4
Napierski, Robert	62091502054	1962-09-15	ul. Podgórska 7, Police	5
Englert, Joanna	72073110039	1972-07-31	ul. Wielka 40, Szczecin	5
Janiszewski, Albert	69032923149	1969-03-29	os. Jana Pawła II 13/4, Szczecin	4
Bąk, Józef	37111045873	1937-11-10	os. Czerwonego Pucharu 1	1

DZIAŁ

NAZWADZ	NUMERDZ	PESELKIEROWNIKA
Badawczy	5	33344445555
Administracja	4	98765432109
Dyrekcja	1	88886665555

Anomalie modyfikowania

- » Zmiana wartości jednego z atrybutów określonego działu, np. kierownika działu o nr 5

(a)

PRAC_DZ



...

Nadmiarowość

IMIĘNAZWPRAC	PESEL	DATAUR	ADRES	NUMERDZ	NAZWADZ	PESELKIEROWNIKA
Nowak, Jan	12345678901	1965-01-09	ul. Nowa 11, 00-900 Warszawa	5	Badawczy	33344445555
Kowalski, Fryderyk	33344445555	1955-12-08	Al. Niepodległości 23, 44-100 Gliwice	5	Badawczy	33344445555
Zielńska, Alicja	99988887777	1968-07-19	Pl. Wolności, 25-600 Kielce	4	Administracja	98765432109
Owsiąk, Maria	98765432109	1941-06-20	ul. Stara 22, 09-000 Warszawa	4	Administracja	98765432109
Głębocki, Adam	66688844444	1962-09-15	ul. Długa 34A, 70-300 Łomża	5	Badawczy	33344445555
Polakowski, Maciej	45345345345	1972-07-31	ul. Krótka 1, 50-550 Szczecin	5	Badawczy	33344445555
Kuc, Waldemar	98798798798	1969-03-29	Pl. Powstańców 10B, 00-111 Warszawa	4	Administracja	98765432109
Szymkowiak, Krystyna	88886665555	1937-11-10	ul. Szeroka 8, 26-100 Starachowice	1	Dyrekcja	88886665555

Anomalie aktualizacji cd

- » Niepożądane
- » Utrudniają zachowanie spójności danych
- » Wymagają zbędnych aktualizacji

Wskazówka 2: Schematy relacji należy projektować tak, aby w relacjach nie występowały żadne anomalie wstawiania, usuwania i modyfikowania.

W razie występowania pewnych anomalii należy je dokładnie określić i zapewnić, aby programy dokonujące aktualizacji w bazie danych działały poprawnie

Ad 3. Wartości *null* w krotkach

- » Jeżeli wiele atrybutów nie ma zastosowania w przypadku wszystkich krotek relacji, otrzymujemy wiele wartości *null* w tych krotkach
 - marnowanie miejsca
 - problemy ze zrozumieniem znaczenia atrybutów oraz z określaniem operacjiłączenia na poziomie logicznym
 - nieprzewidywalność operacji agregujących

Ad 3. Wartości *null* w krotkach c.d.

» Wiele interpretacji wartości *null*:

- atrybut *nie ma zastosowania* w przypadku danej krotki (np. stan_wizy może nie dotyczyć studentów z Polski);
- wartość atrybutu w przypadku danej krotki jest *nieznana* (np. data_ur pracownika może być nieznana);
- wartość jest *znana, ale nieobecna*, to znaczy nie została jeszcze zarejestrowana (np. numer_domowy pracownika może istnieć, ale nie jest jeszcze określony lub zarejestrowany)

Wskazówka 3. W jak największym stopniu należy unikać umieszczania w relacji atrybutów, które często mogą przyjmować wartość *null*.

Jeżeli nie da się uniknąć występowania wartości *null*, należy zapewnić, aby występowały tylko w przypadkach wyjątkowych, a nie w przypadku większości krotek relacji.

Egzamin

- » Egzamin będzie w trybie zdalnym na UPEL – kurs wykładowy: [**Podstawy Baz Danych WYKŁAD 2020-2021**](#)
- » Ze względu na liczbę osób będzie on w formie testowej - jednokrotnego lub wielokrotnego wyboru.
- » Terminy:
 - **4 luty** (czwartek) 10.00 - 1 termin
 - **11 luty** (czwartek) 10.00 - 1 termin poprawkowy
 - **18 luty** (czwartek) 10.00 - 2 termin poprawkowy

Ad 4. Generowanie fałszywych krotek

(a)

PRAC_LOK

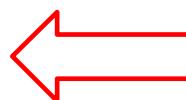
IMIĘNAZWPRAC	LOKALIZACJAP
--------------	--------------

Klucz główny

PRAC_PROJ1

PESEL	NUMERPROJ	GODZINY	NAZWAPROJ	LOKALIZACJAP
-------	-----------	---------	-----------	--------------

Klucz główny



PRAC_PROJ

PESEL	NUMERPROJ	GODZINY	IMIĘNAZWPRAC	NAZWAPROJ	LOKALIZACJAPROJ
-------	-----------	---------	--------------	-----------	-----------------

FD1

FD2

FD3

(b)

PRAC_LOK

IMIĘNAZWPRAC	LOKALIZACJAP
Nowak, Jan	Kielce
Nowak, Jan	Kraków
Głębocki, Adam	Warszawa
Polakowski, Maciej	Kielce
Polakowski, Maciej	Kraków
Kowalski, Fryderyk	Kraków
Kowalski, Fryderyk	Warszawa
Kowalski, Fryderyk	Gliwice
Zielińska, Alicja	Gliwice
Kuc, Waldemar	Gliwice
Owsiak, Maria	Gliwice
Owsiak, Maria	Warszawa
Szymkowiak, Krystyna	Warszawa

PRAC_PROJ1

PESEL	NUMERPROJ	GODZINY	NAZWAPROJ	LOKALIZACJAP
12345678901	1	32,5	ProduktX	Kielce
12345678901	2	7,5	ProduktY	Kraków
66688844444	3	40,0	ProduktZ	Warszawa
45345345345	1	20,0	ProduktX	Kielce
45345345345	2	20,0	ProduktY	Kraków
33344445555	2	10,0	ProduktY	Kraków
33344445555	3	10,0	ProduktZ	Warszawa
33344445555	10	10,0	Komputeryzacja	Gliwice
33344445555	20	10,0	Reorganizacja	Warszawa
99988887777	30	30,0	Nowe zyski	Gliwice
99988887777	10	10,0	Zielińska, Alicja	Komputeryzacja
99988887777	10	10,0	Zielińska, Alicja	Gliwice
98798798798	10	10,0	Komputeryzacja	Gliwice
98798798798	30	35,0	Komputeryzacja	Gliwice
98798798798	30	5,0	Nowe zyski	Gliwice
98765432109	30	20,0	Nowe zyski	Gliwice
98765432109	20	15,0	Reorganizacja	Warszawa
88886665555	20	NULL	Reorganizacja	Warszawa

PRAC_PROJ

PESEL	NUMERPROJ	GODZINY	Nadmiarowość		LOKALIZACJAPROJ
			IMIĘNAZWPRAC	NAZWAPROJ	
12345678901	1	32,5	Nowak, Jan	ProduktX	Kielce
12345678901	2	7,5	Nowak, Jan	ProduktY	Kraków
66688844444	3	40,0	Głębocki, Adam	ProduktZ	Warszawa
45345345345	1	20,0	Polakowski, Maciej	ProduktX	Kielce
45345345345	2	20,0	Polakowski, Maciej	ProduktY	Kraków
33344445555	2	10,0	Kowalski, Fryderyk	ProduktY	Kraków
33344445555	3	10,0	Kowalski, Fryderyk	ProduktZ	Warszawa
33344445555	10	10,0	Kowalski, Fryderyk	Komputeryzacja	Gliwice
33344445555	20	10,0	Kowalski, Fryderyk	Reorganizacja	Warszawa
99988887777	30	30,0	Zielińska, Alicja	Nowe zyski	Gliwice
99988887777	10	10,0	Zielińska, Alicja	Komputeryzacja	Gliwice
98798798798	10	35,0	Kuc, Waldemar	Komputeryzacja	Gliwice
98798798798	30	5,0	Kuc, Waldemar	Nowe zyski	Gliwice
98765432109	30	20,0	Owsiak, Maria	Nowe zyski	Gliwice
98765432109	20	15,0	Owsiak, Maria	Reorganizacja	Warszawa
88886665555	20	NULL	Szymkowiak, Krystyna	Reorganizacja	Warszawa

(a)

PRAC_LOK

IMIĘNAZWPRAC	LOKALIZACJAP
--------------	--------------

Klucz główny

PRAC_PROJ1

PESEL	NUMERPROJ	GODZINY	NAZWAPROJ	LOKALIZACJAP
-------	-----------	---------	-----------	--------------

Klucz główny

(b)

PRAC_LOK

IMIĘNAZWPRAC	LOKALIZACJAP
Nowak, Jan	Kielce
Nowak, Jan	Kraków
Głobocki, Adam	Warszawa
Polakowski, Maciej	Kielce
Polakowski, Maciej	Kraków
Kowalski, Fryderyk	Kraków
Kowalski, Fryderyk	Warszawa
Kowalski, Fryderyk	Gliwice
Zielińska, Alicja	Gliwice
Kuc, Waldemar	Gliwice
Owsiak, Maria	Gliwice
Owsiak, Maria	Warszawa
Szymkowiak, Krystyna	Warszawa

PRAC_PROJ1

PESEL	NUMERPROJ	GODZINY	NAZWAPROJ	LOKALIZACJAP
12345678901	1	32,5	ProduktX	Kielce
12345678901	2	7,5	ProduktY	Kraków
66688844444	3	40,0	ProduktZ	Warszawa
45345345345	1	20,0	ProduktX	Kielce
45345345345	2	20,0	ProduktY	Kraków
33344445555	2	10,0	ProduktY	Kraków
33344445555	3	10,0	ProduktZ	Warszawa
33344445555	10	10,0	Komputeryzacja	Gliwice
33344445555	20	10,0	Reorganizacja	Warszawa
99988887777	30	30,0	Nowe zyski	Gliwice
99988887777	10	10,0	Komputeryzacja	Gliwice
98798798798	10	35,0	Komputeryzacja	Gliwice
98798798798	30	5,0	Nowe zyski	Gliwice
98765432109	30	20,0	Nowe zyski	Gliwice
98765432109	20	15,0	Reorganizacja	Warszawa
88886665555	20	NULL	Reorganizacja	Warszawa

PRAC_PROJ

PESEL	NUMERPROJ	GODZINY	IMIĘNAZWPRAC	NAZWAPROJ	LOKALIZACJAPROJ
12345678901	1	32,5	ProduktX	Kielce	Nowak, Jan
12345678901	1	32,5	ProduktX	Kielce	Polakowski, Maciej
12345678901	2	7,5	ProduktY	Kraków	Nowak, Jan
* 12345678901	2	7,5	ProduktY	Kraków	Polakowski, Maciej
* 12345678901	2	7,5	ProduktY	Kraków	Kowalski, Fryderyk
66688844444	3	40,0	ProduktZ	Warszawa	Głobocki, Adam
* 66688844444	3	40,0	ProduktZ	Warszawa	Kowalski, Fryderyk
* 45345345345	1	20,0	ProduktX	Kielce	Nowak, Jan
45345345345	1	20,0	ProduktX	Kielce	Polakowski, Maciej
* 45345345345	2	20,0	ProduktY	Kraków	Nowak, Jan
45345345345	2	20,0	ProduktY	Kraków	Polakowski, Maciej
* 45345345345	2	20,0	ProduktY	Kraków	Kowalski, Fryderyk
* 33344445555	2	10,0	ProduktY	Kraków	Nowak, Jan
33344445555	2	10,0	ProduktY	Kraków	Polakowski, Maciej
* 33344445555	2	10,0	ProduktY	Kraków	Kowalski, Fryderyk
* 33344445555	3	10,0	ProduktZ	Warszawa	Głobocki, Adam
33344445555	3	10,0	ProduktZ	Warszawa	Kowalski, Fryderyk
* 33344445555	10	10,0	Komputeryzacja	Gliwice	Kowalski, Fryderyk
* 33344445555	20	10,0	Reorganizacja	Warszawa	Głobocki, Adam
* 33344445555	20	10,0	Reorganizacja	Warszawa	Kowalski, Fryderyk

*

+

+

Ad 4. Generowanie fałszywych krotek c.d.

- » **Wskazówka 4.** Schematy relacji należy projektować tak, aby mogły być łączane z użyciem warunków równości atrybutów, które są kluczami głównymi lub kluczami obcymi, w sposób gwarantujący, że nie będą generowane fałszywe krotki.

Należy unikać relacji zawierających pasujące atrybuty, które nie stanowią kombinacji (klucz obcy, klucz główny), ponieważ łączanie po takich atrybutach może powodować otrzymywanie fałszywych krotek.

Podsumowanie wskazówek projektowych

- » Problemy, których występowanie można stwierdzić bez użycia dodatkowych narzędzi analitycznych, to:
 - Anomalie powodujące wykonywanie nadmiarowych działań w czasie operacji wstawiania do relacji lub jej modyfikowania, oraz które mogą powodować przypadkową utratę informacji w czasie operacji usuwania z relacji.
 - Marnotrawstwo przestrzeni związane z występowaniem wartości null oraz wynikające z tego trudności w wykonywaniu selekcji, operacji agregujących oraz złączeń.
 - Generowanie niepoprawnych i fałszywych danych w czasie złączeń relacji głównych na podstawie dopasowywania atrybutów, które nie tworzą prawidłowego związku (klucz obcy, klucz główny).

Co dalej?

- » Formalne pojęcia i zagadnienia teoretyczne do precyzyjnieszego zdefiniowania warunków *poprawności* i *niepoprawności pojedynczych schematów relacji*.
 - zależności funkcyjne
 - postaci normalne: 1 NF, 2 NF, 3 NF i Boyce'a-Codda (BCNF)
 - uznane i przyjęte standardy jakości w projektowaniu baz relacyjnych
- » Strategia tworzenia dobrego projektu to:
 - odpowiednia dekompozycja źle zaprojektowanej relacji w celu uzyskania wyższych postaci normalnych

Zależności funkcyjne

Definicja zależności funkcyjnej

- » Niech schemat relacyjnej bazy danych posiada n atrybutów A_1, A_2, \dots, A_n .
Przyjmijmy, że całą bazę danych opisuje schemat pojedynczej **relacji uniwersalnej** $R = \{A_1, A_2, \dots, A_n\}$.
- » **Zależność funkcyjna** (ang. *functional dependency*) **FD**
 - $X \rightarrow Y$, między dwoma zbiorami atrybutów X i Y , które są podzbiorami zbioru R , określa **więzy** (ang. *constraint*) względem możliwych krotek, które mogą tworzyć stan r relacji R .
 - Więzy określają, że dla dowolnych dwóch krotek t_1 i t_2 w r , takich że $t_1[X] = t_2[X]$, musi również zachodzić równość $t_1[Y] = t_2[Y]$.
 - wartości komponentu Y krotki w r zależą od (są *determinowane przez*) wartości komponentu X
 - wartości komponentu X krotki jednoznacznie (**funkcyjnie**) *determinują* wartości komponentu Y
 - istnieje zależność funkcyjna od X do Y lub że Y jest **funkcyjnie zależne** od X

Definicja zależności funkcyjnej c.d.

- » X funkcjnie determinuje Y w schemacie relacji R wtedy i tylko wtedy, gdy dwie krotki stanu $r(R)$, które są zgodne co do swoich wartości X , zgadzają się co do swoich wartości Y
- » Jeżeli więzy nałożone na relację R określają, że nie może występować więcej niż jedna krotka o danej wartości X w dowolnej realizacji relacji $r(R)$
 - to znaczy, kiedy X jest **kluczem kandydującym** relacji R – implikuje to występowanie zależności funkcyjnej $X \rightarrow Y$ dla dowolnego podzbioru atrybutów Y relacji R
 - jeżeli X jest kluczem kandydującym relacji R to $X \rightarrow R$
- » Jeżeli $X \rightarrow Y$ w R , nie musi oznaczać, że $Y \rightarrow X$ w R

Zależności funkcyjne

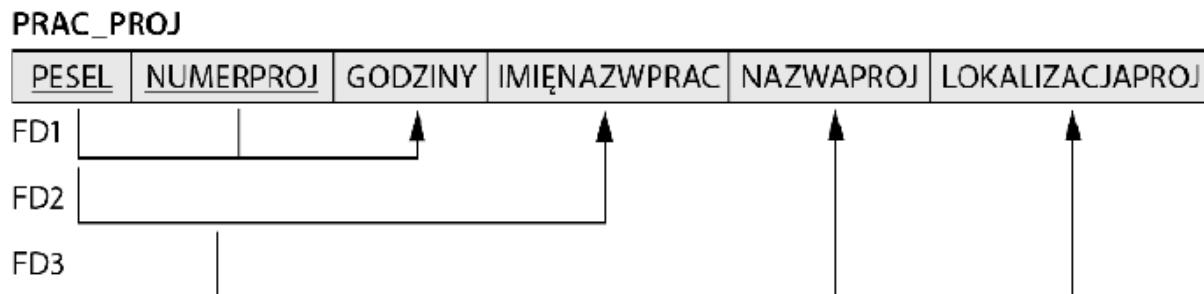
- » Zależność funkcyjna - właściwość *semantyki* (znaczenia) atrybutów
 - do określenia zależności funkcyjnych wykorzystywana znajomość semantyki atrybutów relacji R
- » zależności funkcyjne powinny być zachowane dla *wszystkich* stanów relacji (rozszerzeń) r relacji R
- » Rozszerzenia relacji $r(R)$ spełniające więzy zależności funkcyjnych są określane mianem **dopuszczalnych stanów relacji** (ang. *legal relation states*) lub **dopuszczalnych rozszerzeń** (ang. *legal extensions*) relacji R
- » Głównym zastosowaniem zależności funkcyjnych jest dalsze opisanie schematu relacji R poprzez określenie na jej atrybutach więzów, które muszą być zawsze zachowane

Zależności funkcyjne c.d.

- » Pewne zależności funkcyjne można określić bez odwoływania się do określonych relacji, jako właściwość atrybutów o dobrze znanym znaczeniu
 - np. {WOJEWÓDZTWO, NUMER_PRAWA_JAZDY} → PESEL
- » Ale pewne zależności funkcyjne będą znikać w razie występowania zmian związków
 - np. KOD_POCZTOWY → NUMER_KIERUNKOWY występowała kiedyś w USA ale obecnie już nie obowiązuje.

Przykłady zależności funkcyjnych

- » Na podstawie semantyki atrybutów i relacji wiemy, że powinny być zachowane następujące zależności funkcyjne:
 - a. PESEL → IMIĘNAZWPRAC
 - b. NUMERPROJ → {NAZWAPROJ, LOKALIZACJAPROJ}
 - c. {PESEL, NUMERPROJ} → GODZINY



Zależności funkcyjne – trudność określania

- » Zależność funkcyjna jest właściwością schematu relacji R , a nie pewnego konkretnego dopuszczalnego stanu r relacji R
 - zależność funkcyjna *nie może* być automatycznie wywnioskowana na podstawie danego rozszerzenia relacji r , ale jawnie zdefiniowana przez osobę znającą semantykę atrybutów relacji R .

UCZY

WYKŁADOWCA	PRZEDMIOT	PODRĘCZNIK
Nowak	Struktury danych	Bartram
Nowak	Zarządzanie danymi	Al.-Nour
Jankowski	Kompilatory	Hoffman
Wiśniewski	Struktury danych	Augenthaler

- PODRĘCZNIK → PRZEDMIOT ?
- WYKŁADOWCA → PRZEDMIOT ? PODRĘCZNIK → WYKŁADOWCA ?
PRZEDMIOT → PODRĘCZNIK ?

Zależności funkcyjne - przykład

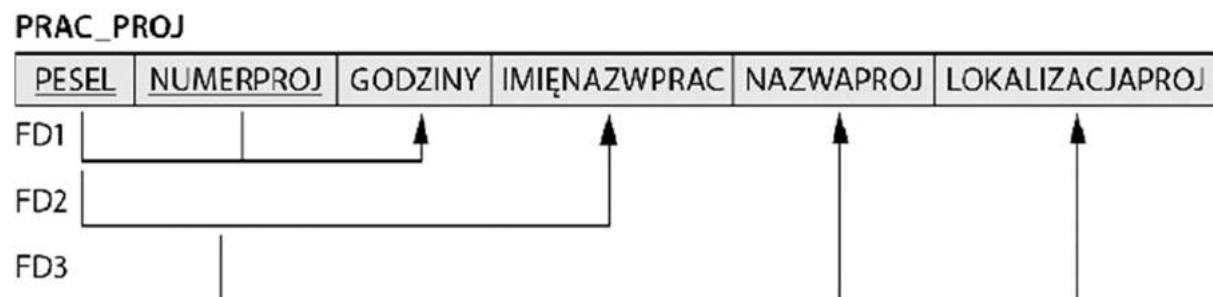
- » Bez znajomości znaczenia atrybutów i relacji między nimi nie można ustalić, które zależności funkcyjne są spełnione, a które nie
- można tylko stwierdzić, że określona zależność funkcyjna *może* występować, jeśli jest spełniona w danym rozszerzeniu
 - bez zrozumienia znaczenia istotnych atrybutów nie można zagwarantować jej występowania
 - jeśli jednak występują krotki naruszające daną zależność funkcyjną, można z przekonaniem stwierdzić, że *nie jest ona spełniona*.

$(B \rightarrow C, C \rightarrow B, \{A, B\} \rightarrow C, \{A, B\} \rightarrow D \text{ i } \{C, D\} \rightarrow B)$
 $(A \rightarrow B, B \rightarrow A, D \rightarrow C)$

A	B	C	D
a1	b1	c1	d1
a1	b2	c2	d2
a2	b2	c2	d3
a3	b3	c4	d3

Zależności funkcyjne – notacja schematyczna

- » F - zbiór zależności funkcyjnych określonych w schemacie relacji R .
- » Projektant schematu zwykle określa zależności funkcyjne, które są *semantycznie oczywiste*
 - dla *wszystkich* dopuszczalnych realizacji relacji spełniających zależności ze zbioru F jest zachowanych również wiele innych zależności funkcyjnych
 - pozostałe zależności można wywnioskować lub wydedukować na podstawie zależności funkcyjnych ze zbioru F - reguły wnioskowania



Postaci normalne oparte na kluczach głównych

Normalizacja relacji

» Codd (1972)

- polega na poddaniu schematu relacji serii testów, określających, czy spełnia on wymagania określonej **postaci normalnej** (ang. *normal form NF*)
- schemat zstępujący – brak zgodności z określonymi warunkami NF = dekompozycja
 - projektowanie relacyjne przez analizę
- początkowo trzy NF
 - bardziej rygorystyczna forma trzeciej = Boyce'a-Codda (BCNF)
 - oparte na zależnościach funkcyjnych występujących między atrybutami relacji
- później 4NF, 5NF
 - bazują na pojęciach zależności wielowartościowych oraz zależności złączeniowych

Normalizacja danych

- » Proces analizowania określonych schematów relacji w oparciu o ich zależności funkcyjne oraz klucze główne w celu otrzymania pożądanych właściwości:
 - 1) minimalizacja nadmiarowości
 - 2) minimalizacja anomalii wstawiania, usuwania i aktualizowania
- » Niepoprawne schematy relacji, które nie spełniają określonych warunków — **testów postaci normalnej** (ang. *normal form tests*), są rozkładane na mniejsze schematy relacji, które z powodzeniem przechodzą testy, a zatem mają odpowiednie właściwości.

Procedura normalizacji

- » Oferuje projektantom baz danych następujące elementy:
 - formalną strukturę analizy schematów relacji w oparciu o ich klucze oraz zależności funkcyjne występujące między ich atrybutami;
 - szereg testów postaci normalnych, które można wykonywać w odniesieniu do pojedynczych schematów relacji, dzięki czemu relacyjna baza danych może zostać **znormalizowana** do dowolnego wymaganego poziomu.
- » Pojęcie **postaci normalnej** relacji odnosi się do warunków najwyższej postaci normalnej, jakie spełnia, a przez to określa poziom, do jakiego została znormalizowana.

Procedura normalizacji c.d.

- » Proces normalizacji przez dekompozycję musi również potwierdzać istnienie dodatkowych właściwości, które, w ujęciu całościowym, posiadają schematy relacji:
 - **właściwość złączenia bezstratnego** (ang. *lossless join property*), inaczej **właściwość złączenia nieaddytywnego** (ang. *nonadditive join property*)
 - gwarantuje, że problem generowania fałszywych krotek, nie występuje w przypadku schematów relacji utworzonych po dekompozycji;
 - **właściwość zachowania zależności** (ang. *dependency preservation property*)
 - każda zależność funkcyjna jest reprezentowana w pewnej pojedynczej relacji wynikającej z dekompozycji.
- » Właściwość złączenia nieaddytywnego niezmiernie istotna
 - **musi być zachowana bezwzględnie**
 - właściwość zachowania zależności, choć pożądana, czasem może być pomijana

Praktyczne zastosowania postaci normalnych

- » 4NF, 5NF – praktyczne wykorzystanie problematyczne
 - więzy, na których bazują – trudne do zrozumienia
- » Projektowanie baz danych, zgodnie z normami obowiązującymi we współczesnym świecie - do poziomu 3NF, BCNF
- » Projektanci *nie muszą* przeprowadzać normalizacji do najwyższej możliwej postaci normalnej
 - relacje można pozostawić w niższej postaci normalnej, na przykład 2NF, ze względów wydajnościowych
 - ale: problemy w postaci konieczności radzenia sobie z anomaliami
- » Proces zachowywania złączenia wyższych postaci normalnych jako relacji, która znajduje się w niższej postaci normalnej, nosi nazwę **denormalizacji** (ang. *denormalization*).

Definicja kluczy

- » **Nadklucz/Superklucz** (ang. *superkey*) schematu relacji $R = \{A_1, A_2, \dots, A_n\}$ to zbiór atrybutów $S \subseteq R$ o tej właściwości, że żadne dwie krotki t_1 i t_2 w dowolnym dopuszczalnym stanie r relacji R nie spełniają równości $t_1[S] = t_2[S]$.
- » **Klucz** (ang. *key*) K jest nadkluczem posiadającym dodatkową właściwość, która określa, że usunięcie dowolnego atrybutu z K powoduje, że K przestaje być nadkluczem.
 - jeżeli posiadamy klucz $K = \{A_1, A_2, \dots, A_n\}$ relacji R , to $K - \{A_i\}$ nie jest kluczem relacji R dla dowolnego A_i , gdzie $1 \leq i \leq k$.
- » Jeżeli schemat relacji posiada więcej niż jeden klucz, każdy z nich jest **kluczem kandydującym** (ang. *candidate key*).
- » Jeden z kluczy kandydujących zostaje *arbitralnie* określony jako **klucz główny** (ang. *primary key*), zaś pozostałe określa się mianem kluczy drugorzędnych.
- » Każdy schemat relacji musi posiadać klucz główny.
- » Jeśli nie jest znany klucz kandydujący relacji, całą relację można traktować jako domyślny nadklucz.

Atrybut podstawowy/niepodstawowy

- » Atrybut schematu relacji R określa się mianem **atrybutu podstawowego** (ang. *prime attribute*) relacji R , jeżeli jest składową *penego klucza kandydującego* relacji R .
- » Atrybut określa się jako **niepodstawowy** (ang. *nonprime*), jeżeli nie jest atrybutem podstawowym, to znaczy jeśli nie jest składową żadnego klucza kandydującego.



Pierwsza postać normalna 1NF

- » **Pierwsza postać normalna** (ang. *first normal form*, 1NF) jest obecnie traktowana jako element formalnej definicji relacji w podstawowym modelu relacyjnym
 - historycznie zdefiniowana jako warunek zapobiegania występowaniu atrybutów wielowartościowych, atrybutów złożonych oraz ich kombinacji.
- » Dziedzina atrybutu musi zawierać wyłącznie *wartości niepodzielne* (proste) oraz że wartość dowolnego atrybutu krotki musi być *pojedynczą wartością* pochodzącą z dziedziny tego atrybutu.
 - zabrania występowania zbioru wartości, krotki wartości lub ich kombinacji jako wartości atrybutu *pojedynczej krotki*
 - zabrania występowania *relacji w relacjach* lub *relacji jako wartości atrybutu w krotkach*
- » Jedyne wartości atrybutów dopuszczalne w ramach postaci 1NF to **pojedyncze wartości niepodzielne** (ang. *atomic values*).

1NF – przykład relacji naruszającej

(a)

DZIAŁ

NAZWADZ	<u>NUMERDZ</u>	PESELKIEROWNIKA	LOKALIZACJEDZ



(b)

DZIAŁ

NAZWADZ	<u>NUMERDZ</u>	PESELKIEROWNIKA	LOKALIZACJEDZ
Badawczy	5	33344445555	{Kielce, Kraków, Warszawa}
Administracja	4	98765432109	{Gliwice}
Dyrekcja	1	88886665555	{Warszawa}

1NF - przykład

<u>NAZWADZ</u>	<u>NUMERDZ</u>	<u>PESELKIEROWNIKA</u>	<u>LOKALIZACJEDZ</u>

DZIAŁ

<u>NAZWADZ</u>	<u>NUMERDZ</u>	<u>PESELKIEROWNIKA</u>	<u>LOKALIZACJEDZ</u>
Badawczy	5	33344445555	{Kielce, Kraków, Warszawa}
Administracja	4	98765432109	{Gliwice}
Dyrekcja	1	88886665555	{Warszawa}

LOKALIZACJE_DZ

<u>NUMERDZ</u>	<u>LOKALIZACJADZ</u>
1	Warszawa
4	Gliwice
5	Kielce
5	Kraków
5	Warszawa

DZIAŁ_1NF

<u>NAZWADZ</u>	<u>NUMERDZ</u>	<u>PESELKIEROWNIKA</u>	<u>LOKALIZACJEDZ</u>
Badawczy	5	33344445555	Kielce
Badawczy	5	33344445555	Kraków
Badawczy	5	33344445555	Warszawa
Administracja	4	98765432109	Gliwice
Dyrekcja	1	88886665555	Warszawa

1NF a relacje zagnieżdżone

- » 1NF zabrania również występowania atrybutów wielowartościowych, które same są złożone
 - **relacje zagnieżdżone** (ang. *nested relations*) - każda krotka może posiadać relację *w sobie*
- » Normalizacja do 1NF:
 - usuwamy atrybuty relacji zagnieżdżonej, przenosząc je do nowej relacji, do której *propagujemy klucz główny*
 - klucz główny nowej relacji stanowi połączenie klucza częściowego z kluczem głównym oryginalnej relacji.

(a)

PRAC_PROJ

		PROJEKTY	
PESEL	IMIĘNAZWPRAC	NUMERPROJ	GODZINY

(b)

PRAC_PROJ

PESEL	IMIĘNAZWPRAC	NUMERPROJ	GODZINY
12345678901	Nowak, Jan	1	32,5
		2	7,5
33344445555	Kowalski, Fryderyk	3	40,0
99988887777	Zielińska, Alicja	1	20,0
		2	20,0
98765432109	Owiak, Maria	2	10,0
		3	10,0
		10	10,0
		20	10,0
66688844444	Głębocki, Adam	30	30,0
		10	10,0
45345345345	Polakowski, Maciej	10	35,0
		30	5,0
98798798798	Kuc, Waldemar	30	20,0
		20	15,0
88886665555	Szymkowiak, Krystyna	20	NULL

(c)

PRAC_PROJ1

PESEL	IMIĘNAZWPRAC
-------	--------------

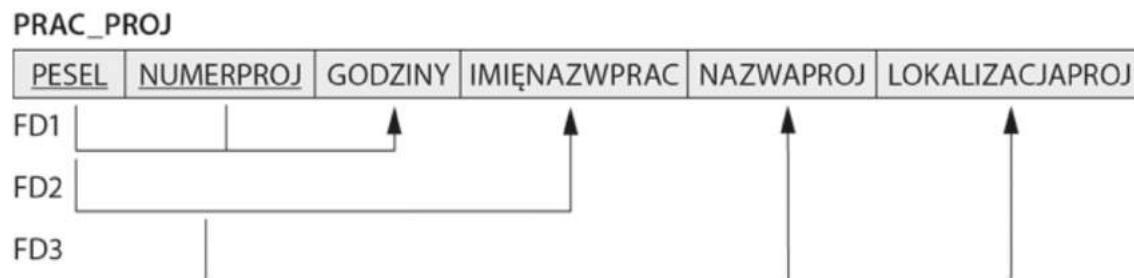
PRAC_PROJ2

PESEL	NUMERPROJ	GODZINY
-------	-----------	---------

Usuwanie zagnieżdżania

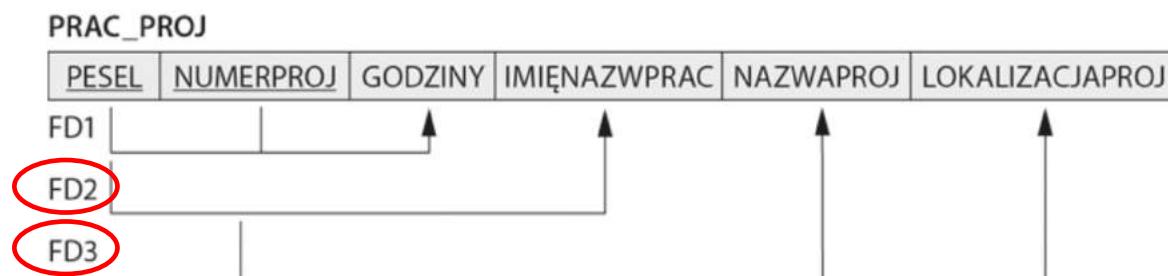
Druga postać normalna 2NF

- » Zależność funkcyjna $X \rightarrow Y$ jest zupełną zależnością funkcyjną, jeżeli usunięcie, jeżeli usunięcie dowolnego atrybutu A ze zbioru X sprawia, że zależność przestaje obowiązywać
 - dla dowolnego atrybutu $A \in X, (X - \{A\})$ nie determinuje funkcyjnie Y
- » Zależność funkcyjna $X \rightarrow Y$ jest zależnością częściową, jeżeli ze zbioru X można usunąć pewien atrybut $A \in X$ a zależność wciąż będzie zachowana
 - $A \in X, (X - \{A\}) \rightarrow Y$
- » $\{\text{PESEL, NUMERPROJ}\} \rightarrow \text{GODZINY}$ – pełna
- » $\{\text{PESEL, NUMERPROJ}\} \rightarrow \text{IMIENAZWPRAC}$ - częściowa



Druga postać normalna 2NF c.d.

- » Definicja Schemat relacji R znajduje się w drugiej postaci normalnej, jeżeli każdy niepodstawowy atrybut A należący do R jest *zupełnie zależny funkcyjnie* od klucza głównego relacji R
- » Test dotyczący 2NF wiąże się ze sprawdzeniem zależności funkcyjnych, których atrybuty lewostronne są częścią PK
 - jeżeli PK zawiera 1 atrybut – test niepotrzebny



Normalizacja do 2 NF

Schemat, który nie jest w 2NF może zostać znormalizowany do wielu relacji 2NF, w których atrybuty nie podstawowe są powiązane tylko częścią PK, od którego są zupełnie zależne funkcyjnie.

(a)

PRAC_PROJ

PESEL	NUMERPROJ	GODZINY	IMIĘNAZWPRAC	NAZWAPROJ	LOKALIZACJAPROJ
FD1			↑	↑	↑
FD2				↑	↑
FD3					↑

Normalizacja do drugiej postaci normalnej

PP1

PESEL	NUMERPROJ	GODZINY
FD1		↑

PP2

PESEL	IMIĘNAZWPRAC
FD2	↑

PP3

NUMERPROJ	NAZWAPROJ	LOKALIZACJAPROJ
FD3	↑	↑

Trzecia postać normalna 3NF

- » Zależność funkcyjna $X \rightarrow Y$ w schemacie relacji R jest zależnością przechodnią, jeżeli istnieje taki zbiór atrybutów Z , który nie tworzy ani klucza kandydującego, ani podzbioru żadnego klucza w relacji R oraz zarówno zależność $X \rightarrow Z$ jak i $Z \rightarrow Y$ jest spełniona
- » PESEL \rightarrow PESELKIEROWNIKA przechodnia przez NUMERDZ

PRAC_DZ

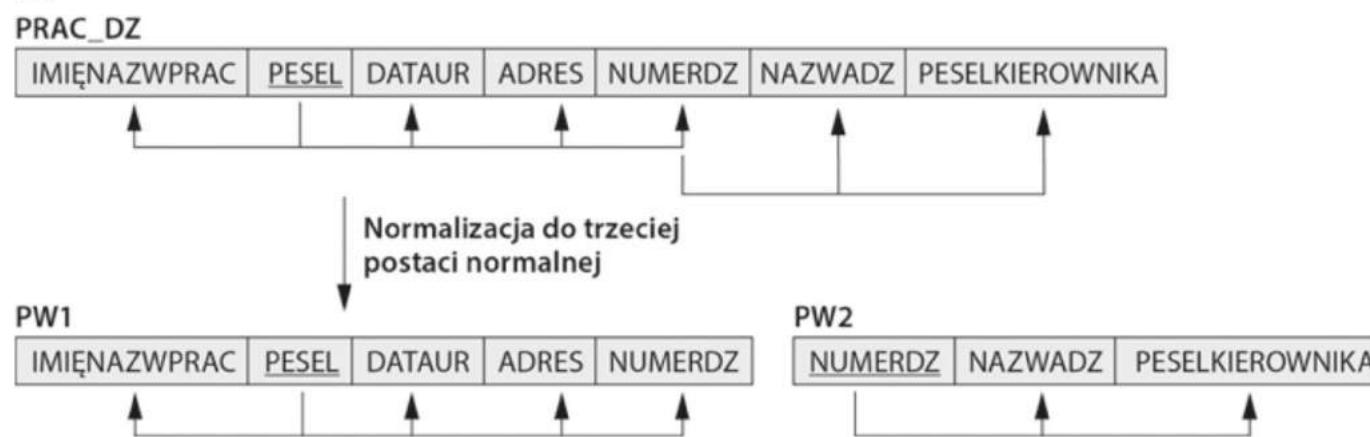
IMIĘNAZWPRAC	PESEL	DATAUR	ADRES	NUMERDZ	NAZWADZ	PESELKIEROWNIKA



The diagram illustrates the dependencies in the PRAC_DZ relation. Each attribute is shown with an arrow pointing upwards to the PESEL attribute, indicating that all other attributes depend on the primary key.

Trzecia postać normalna 3NF c.d.

- » **Definicja** Schemat relacji R jest w *trzeciej postaci normalnej*, jeżeli jest w 2NF oraz żaden niepodstawowy atrybut relacji R nie jest przechodnio zależny od PK



Zależności problematyczne

- » Dowolna zależność funkcyjna, której lewa strona jest częścią (podzbiorem właściwym) klucza głównego, lub dowolna zależność funkcyjna, której lewa strona jest atrybutem niekluczowym
 - normalizacja = rozłożenie oryginalnej relacji na nowe relacje

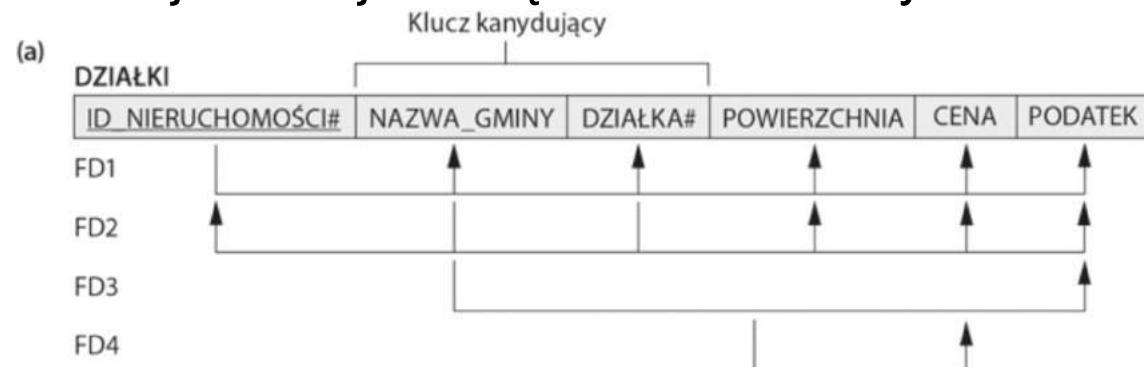
Postaci normalne oparte na PK i metody normalizacji

Postać normalna	Test	Receptura (normalizacja)
Pierwsza (1NF)	Relacja powinna mieć tylko atrybuty niepodzielne oraz nie mieć relacji zagnieżdżonych.	Tworzymy nowe relacje dla każdego atrybutu relacji zagnieżdżonej nie będącego atrybutem niepodzielnym.
Druga (2NF)	W przypadku relacji, których klucz główny zawiera wiele atrybutów, żaden atrybut niekluczowy nie powinien być zależny funkcjnie od części klucza głównego.	Rozkładamy i określamy nową relację dla każdego klucza częściowego z jego zależnym atrybutem (atrbutami). Zapewniamy, aby została zachowana relacja z oryginalnym kluczem głównym oraz wszelkimi atrybutami, które są od niego zupełnie zależne funkcjnie.
Trzecia (3NF)	Relacja nie powinna mieć żadnego atrybutu niekluczowego zdeterminowanego funkcjnie przez inny atrybut niekluczowy (lub zbiór takich atrybutów). Oznacza to, że nie powinna występować żadna zależność przechodnia atrybutu niekluczowego od klucza głównego.	Rozkładamy i określamy relację zawierającą atrybut(y) niekluczowy(e), który(e) determinuje(q) inny(e) atrybut(y) niekluczowy(e).

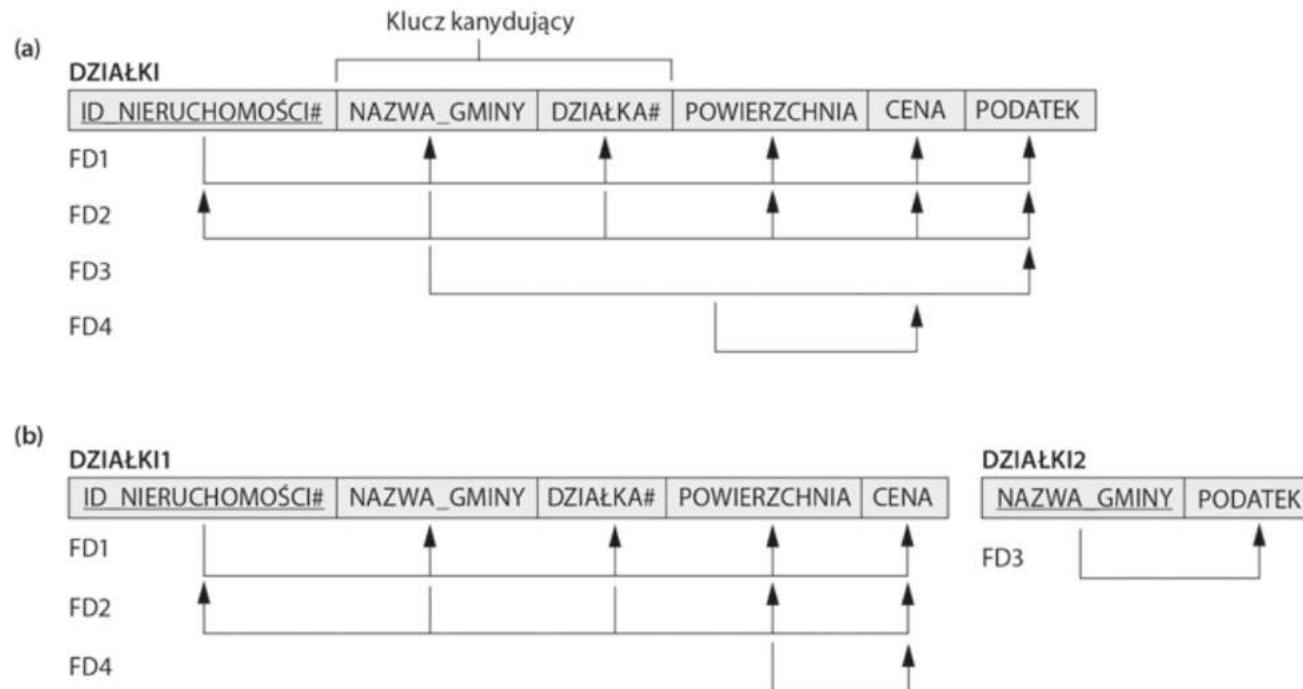
Definicje ogólna 2NF

- » *Atrybut podstawowy* – atrybut będący częścią dowolnego klucza kandydującego
- » **Definicja** Schemat relacji R znajduje się w 2NF, jeżeli żaden nie podstawowy atrybut A relacji R nie jest częściowo zależny od dowolnego klucza relacji R

- » FD3 narusza 2NF



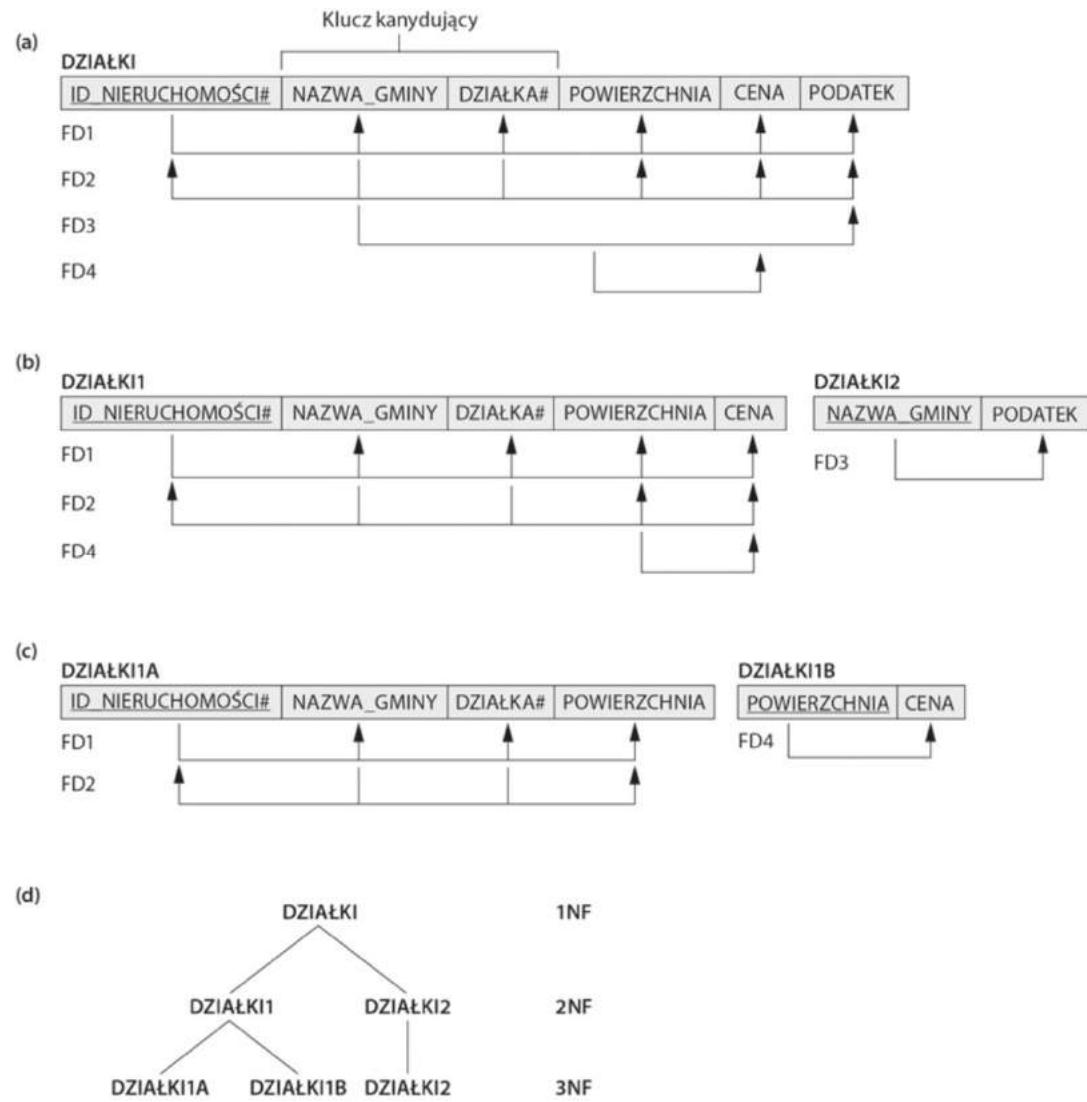
Normalizacja do 2NF



Definicje ogólna 3NF

» **Definicja** Schemat relacji R znajduje się w 3NF, jeżeli zawsze, kiedy *nietrywialna* zależność funkcyjna $X \rightarrow A$ jest zachowana w R , to albo

- (a) X jest nadkluczem w R , albo
- (b) A jest atrybutem podstawowym w relacji R

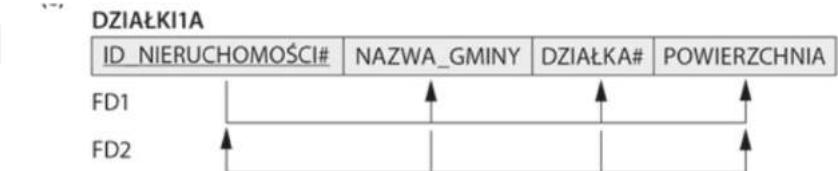
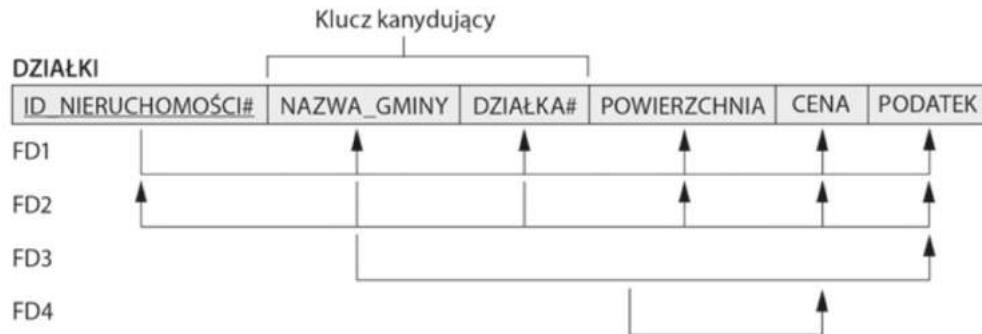


Interpretacja definicji ogólnej 3NF

- » Schemat relacji R narusza definicję ogólną 3NF, jeżeli w relacji R jest zachowana zależność funkcyjna $X \rightarrow A$ i naruszone są oba warunki (a) i (b)
- » Pierwszy warunek pozwala wykryć dwa rodzaje problematycznych zależności:
 - atrybut niepodstawowy determinuje inny atrybut nie podstawowy – występuje wtedy zwykle zależność przechodnia naruszająca 3NF
 - podzbiór właściwy klucza relacji R funkcyjnie determinuje atrybut nie podstawowy – występuje wtedy zależność częściowa naruszająca 2NF
- » **Alternatywna definicja** Schemat relacji R znajduje się w 3NF jeżeli każdy nie podstawowy atrybut relacji R spełnia oba warunki:
 - jest zupełnie zależny funkcyjnie od każdego klucza w R
 - jest nieprzechodnio zależny od każdego klucza w R

Postać normalna Boyce'a-Codda BCNF

- » Zaproponowana jako uproszczona wersja 3NF, ale okazała się bardziej restrykcyjna
 - każda relacja, która jest w BCNF jest w 3NF – niekoniecznie odwrotnie



Rozmiary działek w gminie Sandomierz: 5,6,7,8,9,10 arów

Rozmiary działek w gminie Połczyn: 11,12,...,19 i 20 arów

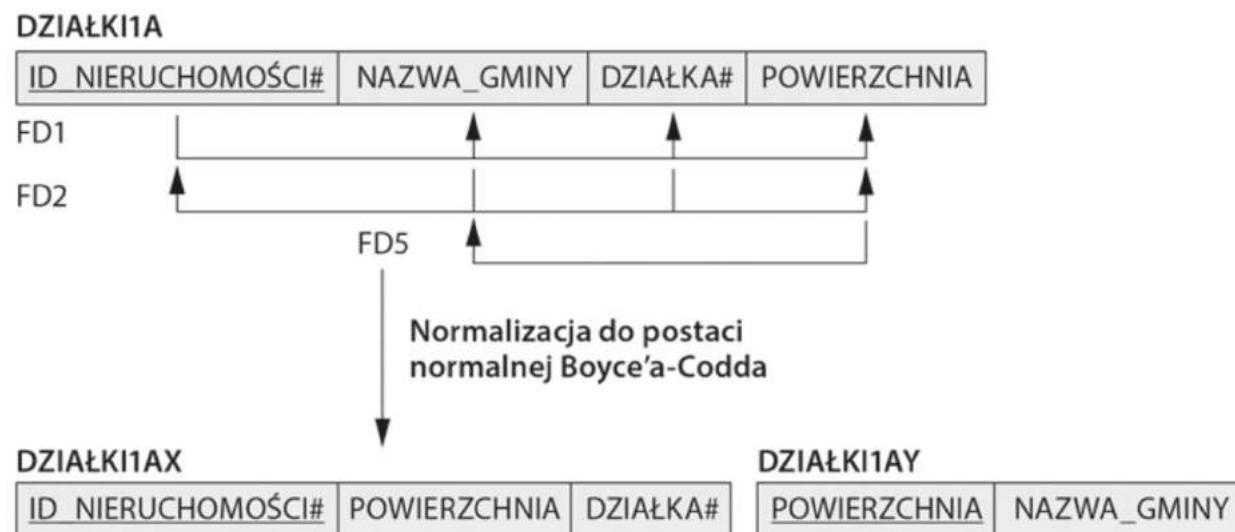
Dodatkowa zależność:

FD5: POWIERZCHNIA \rightarrow NAZWA_GMINY

Oddzielna relacja $R(\text{POWIERZCHNIA}, \text{NAZWA_GMINY})$

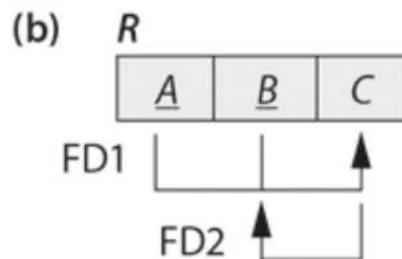
Postać normalna Boyce'a-Codda BCNF c.d.

- » **Definicja** Schemat relacji R znajduje się w postaci *normalnej Boyce'a-Codda (BCNF)*, jeżeli zawsze, kiedy nietrywialna zależność funkcyjna $X \rightarrow A$ jest zachowana w R , to X jest nadkluczem w R



3NF a BCNF

- » Większość schematów relacji, które są w 3NF, są również w BCNF
- » Tylko wtedy, gdy zależność $X \rightarrow A$ jest zachowana w R , a X nie jest nadkluczem oraz A jest atrybutem podstawowym, relacja R znajduje się w 3NF ale nie w BCNF



- » Projekt relacyjnej BD powinien osiągać postać BCNF lub 3NF

Dekompozycja relacji niebędących w BCNF

» Relacja UCZY – 3NF, ale nie BCNF

FD1: {STUDENT,PRZEDMIOT} -> WYKŁADOWCA

FD2: WYKŁADOWCA -> PRZEDMIOT

{STUDENT,PRZEDMIOT} – klucz kandydujący

UCZY

STUD	PRZEDMIOT	WYKŁADOWCA
Głębocki	Bazy danych	Robakiewicz
Nowak	Bazy danych	Nowosielski
Nowak	Systemy operacyjne	Anielewicz
Nowak	Teoria	Szuman
Owiak	Bazy danych	Robakiewicz
Owiak	Systemy operacyjne	Ostrowski
Kowalski	Bazy danych	Omieciński
Zielińska	Bazy danych	Nowosielski
Głębocki	Systemy operacyjne	Anielewicz

Dekompozycja relacji niebędących w BCNF c.d.

- » Możliwe rozkłady:
 1. R1 {STUDENT, WYKŁADOWCA} i R2{STUDENT, PRZEDMIOT}
 2. R1 {PRZEDMIOT, WYKŁADOWCA} i R2{PRZEDMIOT ,STUDENT}
 3. R1 {WYKŁADOWCA, PRZEDMIOT} i R2{WYKŁADOWCA, STUDENT}
- » Utrata FD1
- » W procesie normalizacji staramy się zapewnić dwie cechy dekompozycji:
 - **właściwość złączenia nieaddytywnego**
 - właściwość zachowania zależności (-)

Dekompozycja relacji niebędących w BCNF c.d.

- » Dekompozycja $D = \{R_1, R_2\}$ relacji R posiada właściwość złączenia bezstratnego (nieaddytywnego) w kontekście zbioru zależności funkcyjnych F na relacji R wtedy i tylko wtedy, gdy jest spełniony jeden z warunków:
 - zależność funkcyjna $((R_1 \cap R_2) \rightarrow (R_1 - R_2))$ należy do domknięcia F^+
 - zależność funkcyjna $((R_1 \cap R_2) \rightarrow (R_2 - R_1))$ należy do domknięcia F^+

» $R1 \{WYKŁADOWCA, PRZEDMIOT\}$ i $R2\{WYKŁADOWCA, STUDENT\}$

$R_1 \cap R_2 = WYKŁADOWCA$

$R_1 - R_2 = PRZEDMIOT$

- » Poprawna dekompozycja relacji UCZY na relacje zgodne z BCNF:
- » UCZY1(WYKŁADOWCA, PRZEDMIOT) I UCZY2(WYKŁADOWCA, STUDENT)

Dekompozycja relacji niebędących w BCNF c.d.

- » Relację R , która nie jest w BCNF, można rozłożyć w sposób zgodny z właściwością złączenia nieaddytywnego za pomocą procedury dzielącej R stopniowo na zbiór relacji w postaci BCNF:
 - niech R będzie relacją niebędącą w postaci BCNF, $X \subseteq R$, a $X \rightarrow A$, to zależność funkcyjna powodująca naruszenie BCNF. R można rozdzielić na dwie relacje:
 - $R-A$
 - XA
 - jeżeli $R-A$ lub XA nie są w postaci BCNF, proces należy powtórzyć

Zależności wielowartościowe i 4NF

- » Pracownik NAZWPRAC, pracuje nad projektem NAZWAPROJ i posiada członka rodziny IMIECZŁRODZ

NAZWPRAC	NAZWAPROJ	IMIECZŁRODZ
Nowak	X	Jan
Nowak	Y	Anna
Nowak	X	Anna
Nowak	Y	Jan

- » Relacja p kluczu całościowym, bez zależności funkcyjnych, BCNF

Zależności wielowartościowe

- » Niektóre relacje posiadają więzy, których nie da się wyrazić za pomocą zależności funkcyjnych – dlatego nie naruszają BCNF -> zależności wielowartościowe (MVD), 4NF
 - zawsze wtedy gdy dwie niezależne relacje $A:B$ i $A:C$ typu 1:N występuje w tej samej relacji $R(A,B,C)$ mogą pojawić się MVD

Zależności wielowartościowe c.d.

- » **Definicja** Zależność wielowartościowa $X \rightarrow\rightarrow Y$ określona na relacji R , gdzie X i Y są podzbiorami relacji R , określa następujące więzy na dowolnym stanie r relacji R :
- jeżeli w r istnieją dwie krotki t_1 i t_2 , takie, że $t_1[X] = t_2[X]$, to w r powinny również istnieć dwie krotki t_3 i t_4 o następujących właściwościach (Z oznacza zbiór $(R - (X \cup Y))$):
 - $t_3[X] = t_4[X] = t_1[X] = t_2[X]$;
 - $t_3[Y] = t_1[Y]$ oraz $t_4[Y] = t_2[Y]$;
 - $t_3[Z] = t_2[Z]$ oraz $t_4[Z] = t_1[Z]$
- » Kiedy jest zachowana zależność $X \rightarrow\rightarrow Y$ – mówimy, że X determinuje wielokrotnie Y
- ze względu na symetryczność jaka występuje w definicji, kiedy w relacji R zachodzi $X \rightarrow\rightarrow Y$, zachodzi również $X \rightarrow\rightarrow Z$
 - stąd $X \rightarrow\rightarrow Y$ implikuje, że $X \rightarrow\rightarrow Z$, co zapisuje się $X \rightarrow\rightarrow Y \sqsubset Z$

NAZWPRAC	NAZWAPROJ	IMIĘCZŁRODZ
Nowak	X	Jan
Nowak	Y	Anna
Nowak	X	Anna
Nowak	Y	Jan

Trywialne/nietrywialne zależności

- » Zależność wielowartościowa $X \rightarrow\rightarrow Y$ jest trywialną zależnością wielowartościową, jeżeli

- Y jest podzbiorem X lub
- $X \cup Y = R$

PROJEKTY_PRAC

NAZWPRAC	NAZWAPROJ
Nowak	X
Nowak	Y

PRAC_PODWŁADNI

NAZWPRAC	IMIĘCZŁRODZ
Nowak	Jan
Nowak	Anna

- » Zależność wielowartościowa nie spełniająca ani a) ani b)
 - nietrywialna zależność wielowartościowa

Nietrywialne zależności wielowartościowe

- » Jeżeli w relacji występuje nietrywialna zależność wielowartościowa może się okazać, że w krotkach trzeba powtarzać nadmiarowe wartości

PRAC

<u>NAZWPRAC</u>	<u>NAZWAPROJ</u>	<u>IMIECZŁRODZ</u>
Nowak	X	Jan
Nowak	Y	Anna
Nowak	X	Anna
Nowak	Y	Jan

- » Niepożądana nadmiarowość
 - ale PRAC jest w BCNF
 - trzeba zdefiniować bardziej rygorystyczną od BCNF postać normalną
 - relacje zawierające nietrywialne zależności wielowartościowe zwykle są relacjami o kluczu całościowym

Czwarta postać normalna 4NF

- » **Definicja** Schemat relacji R znajduje się w *czwartej postaci normalnej* w kontekście zbioru zależności funkcyjnych F (zależności funkcyjne/wielowartościowe), jeżeli dla każdej nietrywialnej zależności wielowartościowej $X \rightarrow\!\!\!\rightarrow Y$ w domknięciu F^+ zbiór X jest nadkluczem relacji R
- relacja z kluczem całościowym zawsze jest w postaci BCNF, ponieważ nie zawiera zależności funkcyjnych
 - relacja z kluczem całościowym, która nie ma zależności funkcyjnych, ale obejmuje zależność wielowartościową nie jest w postaci 4NF
 - relację, która z powodu nietrywialnej zależności wielowartościowej nie jest w 4NF, trzeba rozłożyć, aby przekształcić ją w zbiór relacji zgodnych z 4NF
 - dekompozycja umożliwia wyeliminowanie nadmiarowości powodowanej zależnością wielowartościową

Dekompozycja do 4NF

PRAC

NAZWPRAC	NAZWAPROJ	IMIĘCZŁRODZ
Nowak	X	Jan
Nowak	Y	Anna
Nowak	X	Anna
Nowak	Y	Jan

NAZWPRAC	NAZWAPROJ
Nowak	X
Nowak	Y

PROJEKTY_PRAC

NAZWPRAC	IMIĘCZŁRODZ
Nowak	Jan
Nowak	Anna

PRAC_CZŁRODZINY



AGH

Struktury przechowywania danych

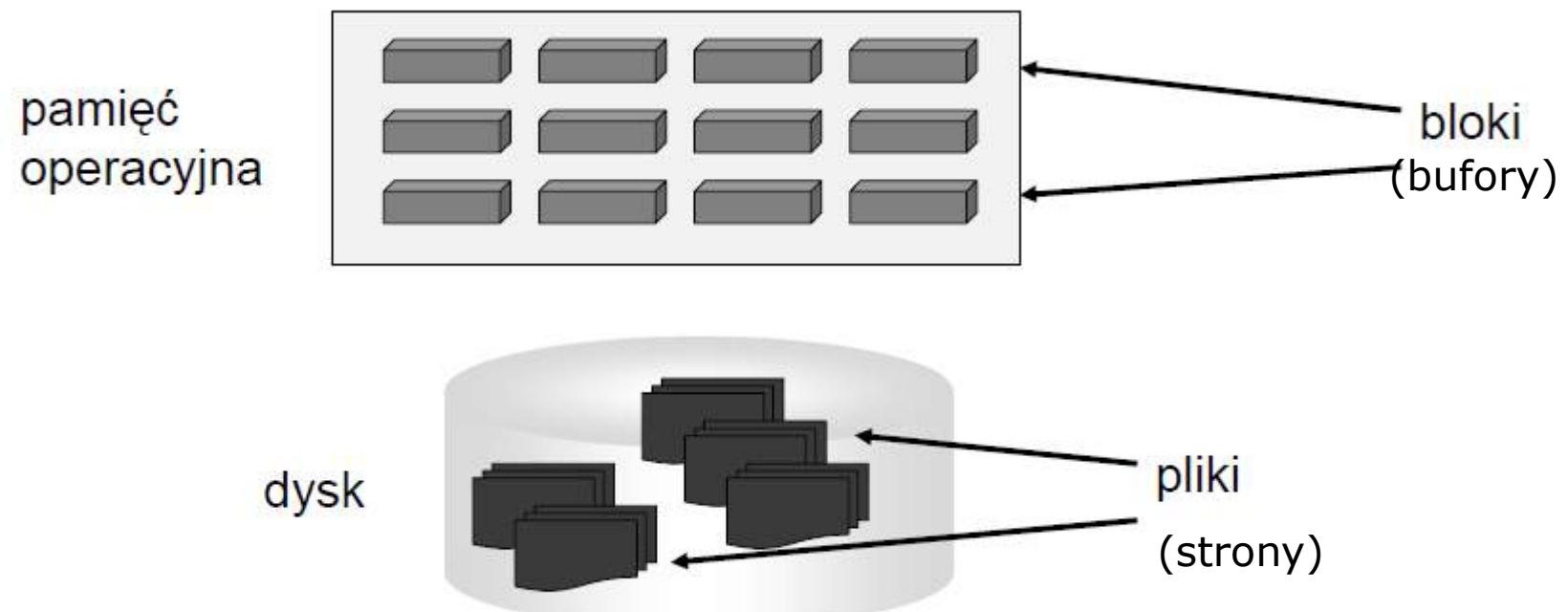


AGH

- » Dane do przetwarzania – pamięć główna
- » Trwałe dane w BD są przechowywane w pamięci zewnętrznej:
 - ze względu na rozmiar danych
 - odporność pamięci zewnętrznej na awarie
 - niski koszt przechowywania
- » Buforowanie bloków dyskowych
 - dane odczytywane z plików do bloków (buforów) BD
 - udostępniane użytkowników BD (odczyt/modyfikacje)
 - zapis do plików na dysku buforów BD

» Media fizyczne tworzą hierarchię pamięci składającą się z:

- pamięci operacyjnej o organizacji blokowej
- pamięci zewnętrznej o organizacji plikowej



Organizacja pliku

- » *Blok* może zawierać wiele rekordów
 - założenie:
 - rekord nie jest dłuższy od bloku
 - każdy rekord zawiera się w całości w bloku
- » Najprostsze podejście:
 - założenie, że wielkość rekordu jest stała
 - każdy plik ma rekordy tylko jednego typu
 - różne pliki do różnych relacji

Rekordy stałej długości - przykład

- » Plik zawiera rekordy *instructor*, każdy rekord:

```
type instructor = record
    ID varchar(5);
    name varchar(20);
    dept_name varchar(20);
    salary numeric(8,2);
end
```

- » każdy rekord zajmuje 53 bajty (1 znak= 1 bajt, numeric = 8 bajtów)

Rekordy stałej długości

- Podejście:
 - Przechowuj rekord i zaczynając od bajtu $n * (i - 1)$, gdzie n jest rozmiarem każdego rekordu.
 - Dostęp do rekordu prosty, ale rekordy mogą zajmować wiele bloków
 - o podejście: pozostawić puste miejsce w bloku
- Usuwanie rekordu i (trudne):
podejścia:
 - przesuń rekordy $i + 1, \dots, n$ do $i, \dots, n - 1$
 - przesuń rekord n do i
 - nie przesuwaj rekordów, ale zlinkuj wszystkie wolne rekordy w *free list*

instructor

record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Usunięcie rekordu 3 i kompaktowanie

record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Usunięcie rekordu 3 i przesunięcie ostatniego

record 0
record 1
record 2
record 11
record 4
record 5
record 6
record 7
record 8
record 9
record 10

10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
98345	Kim	Elec. Eng.	80000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000

Free List

- Przechowaj adres pierwszego skasowanego rekordu w nagłówku pliku.
- Użyj pierwszy rekord do przechowania adresu drugiego skasowanego rekordu itd.
- Można traktować przechowywane adresy jako **wskaźniki** gdyż wskazują na lokalizację rekordu (*free list*).
- Wstawianie nowego rekordu: wykorzystanie pierwszego z *free list* oraz aktualizacja adresu w nagłówku

header				
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1				
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4				
record 5	33456	Gold	Physics	87000
record 6				
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Reprezentacja rekordu (atrybuty stałej i zmiennej długości)

- Przyczyny występowania rekordów o zmiennej długości:
 - przechowywanie różnych typów rekordów w pliku
 - typy rekordów pozwalające na zmienną długość dla jednego lub większej liczby pól (**varchar**)
- Różne techniki do implementacji rekordów zmiennej długości.
- Problemy do rozwiązania:
 - jak reprezentować pojedynczy rekord aby można było prosto wyekstrahować pojedyncze atrybuty
 - jak przechowywać rekordy zmiennej długości w bloku, tak by rekordy z bloku było łatwo wyekstrahować



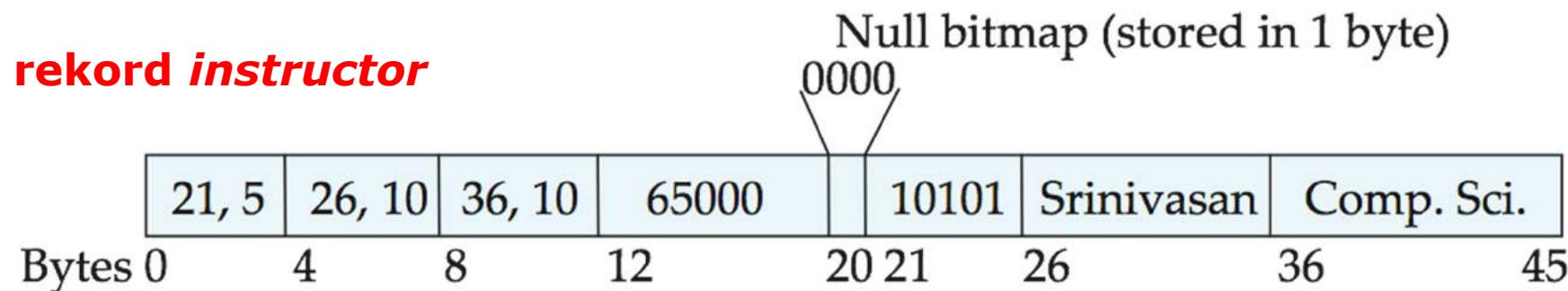
AGH

Reprezentacja rekordu (atrybuty stałej i zmiennej długości) (1)

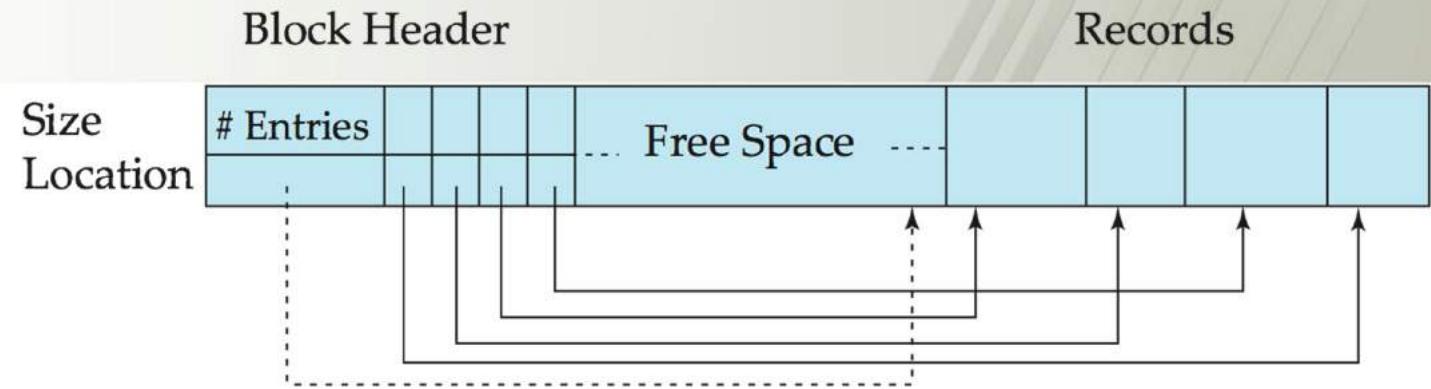
```
type instructor = record (ID varchar(5);  
                           name varchar(20);  
                           dept_name varchar(20);  
                           salary numeric(8,2);end
```

- Złożona z 2 części:
 - część początkowa z atrybutami stałej długości
 - dane dla atrybutów zmiennej długości
- Atrybuty stałej długości (liczby, daty, stringi) – alokowane tyle bajtów ile wymaga długość
- Atrybuty zmiennej długości (varchar) reprezentowane w początkowej części rekordu przez parę: (offset, długość), z aktualnymi danymi przechowywanymi po wszystkich atrybutach stałej długości
- Wartości *null* reprezentowane przez *null bitmapę* (czasem na początku)

rekord *instructor*



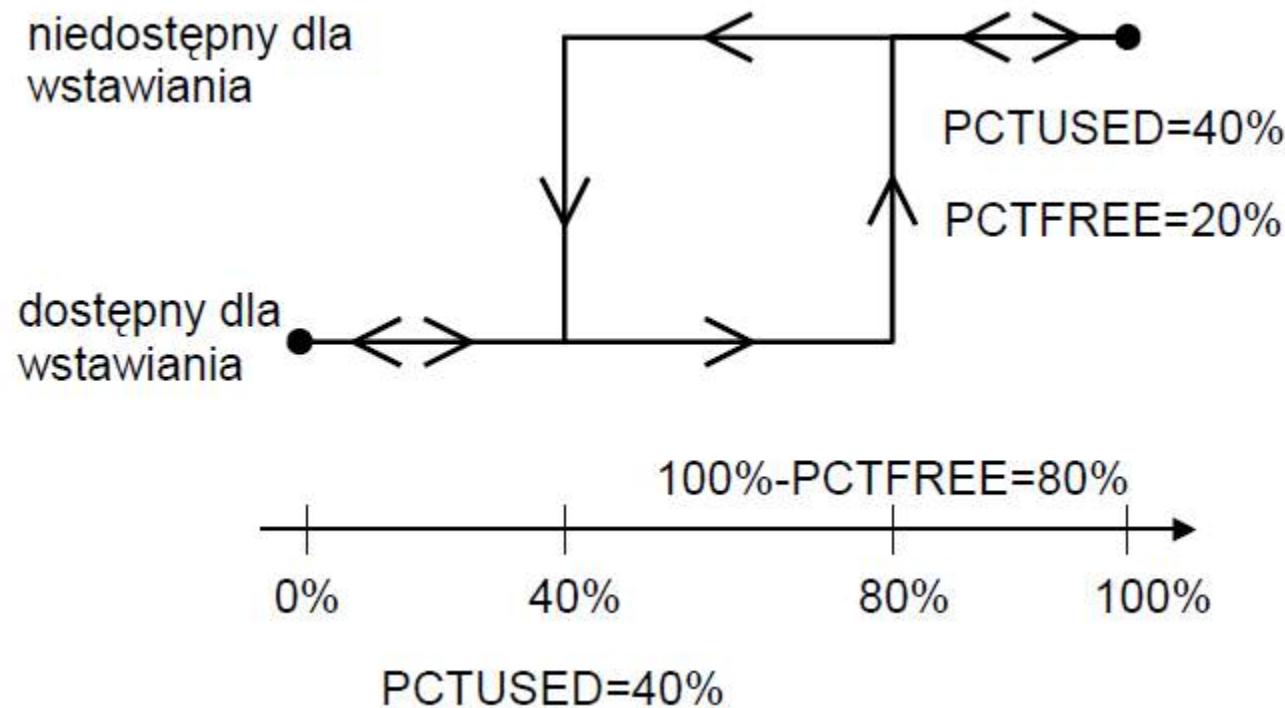
Przechowywanie rekordów zmiennej długości w bloku: Struktura Slotowa Strony (Slotted Page Structure)



- Nagłówek **slotted page** zawiera:
 - liczba rekordów
 - koniec wolnej przestrzeni w bloku
 - tablica z lokalizacjami i rozmiarem każdego rekordu
- Wstawianie rekordów
- Usuwanie rekordów – inne przesuwane wraz z aktualizacją nagłówka (małe koszty)
- Brak konieczności przechowywania wskaźników bezpośrednio do rekordów

Zarządzanie rozmiarem bloku danych

- » Utrzymywanie wolnej pamięci w bloku dla potencjalnej modyfikacji
 - PCTUSED – kiedy do bloku można wstawiać rekordy (%rozmiaru bloku)
 - PCTFREE – ile % rozmiaru bloku pozostanie wolne



Jak organizować zbiór rekordów w pliku

- **Nieuporządkowane** – rekordy przechowywane w dowolnej kolejności
- **Uporządkowane (sekwencyjne)** – rekordy wstawiane w porządku sekwencyjnym, w oparciu o wartość klucza wyszukiwania
- **Haszowe** – funkcja haszująca wyliczana na określonym atrybutie w każdym rekordzie; wynik wskazuje w którym bloku w pliku należy umieścić rekord
- **Multitable clustering (klastrowanie wielotablicowe)** rekordy wielu różnych relacji mogą być przechowywane w tym samym pliku
 - Motywacja: przechowywanie powiązanych rekordów w tym samym bloku minimalizuje I/O

Organizacja pliku sekwencyjnego

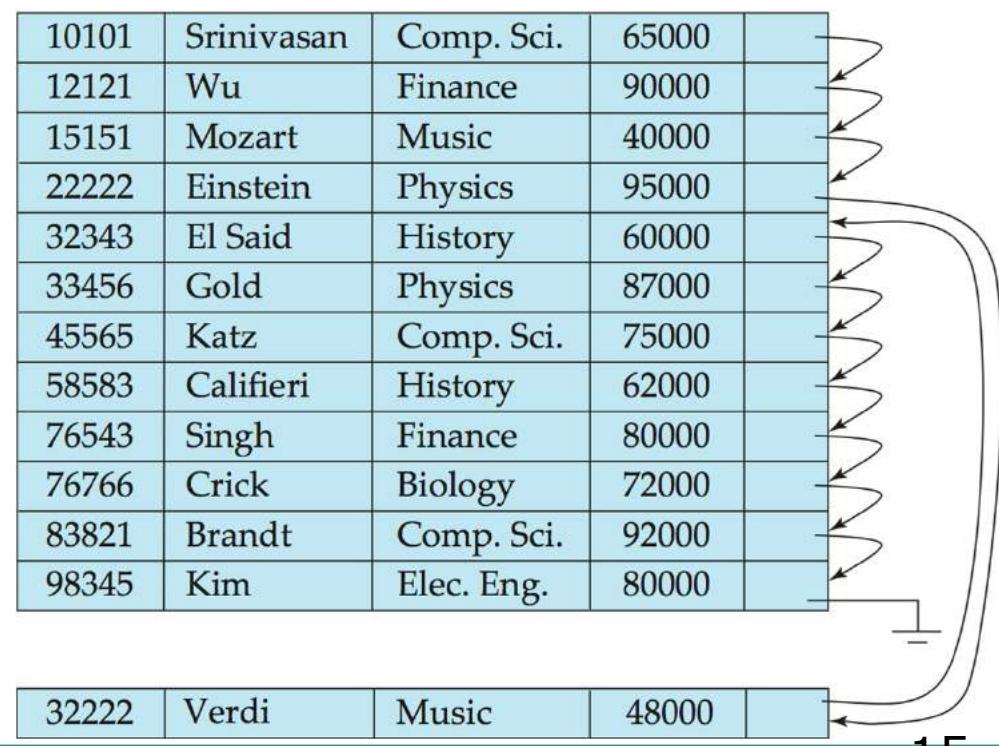
- Przydatne dla aplikacji wymagających sekwencyjnego przetwarzania całego pliku
- Rekordy w pliku fizycznie uporządkowane wg klucza wyszukiwania (**search-key**)
- Pointer wskazuje na następny rekord wg klucza wyszukiwania

10101	Srinivasan	Comp. Sci.	65000	
12121	Wu	Finance	90000	
15151	Mozart	Music	40000	
22222	Einstein	Physics	95000	
32343	El Said	History	60000	
33456	Gold	Physics	87000	
45565	Katz	Comp. Sci.	75000	
58583	Califieri	History	62000	
76543	Singh	Finance	80000	
76766	Crick	Biology	72000	
83821	Brandt	Comp. Sci.	92000	
98345	Kim	Elec. Eng.	80000	



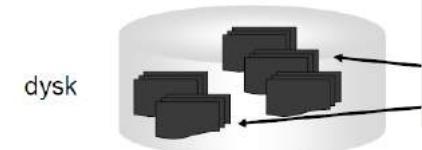
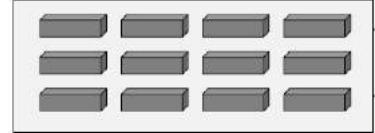
Organizacja pliku sekwencyjnego c.d.

- Usuwanie – wykorzystanie łańcucha wskaźników
- Wstawianie – zlokalizuj pozycję, gdzie rekord powinien być wstawiony
 - jeżeli jest wolne miejsce, wstaw tam
 - jeżeli brak wolnego miejsca, wstaw rekord do **overflow block**
 - w każdym przypadku, łańcuch wskaźników musi być uaktualniony
- Potrzeba okresowej reorganizacji pliku w celu odtworzenia porządku sekwencyjnego
- Trudne zarządzanie



Dostęp do pamięci

pamięć operacyjna



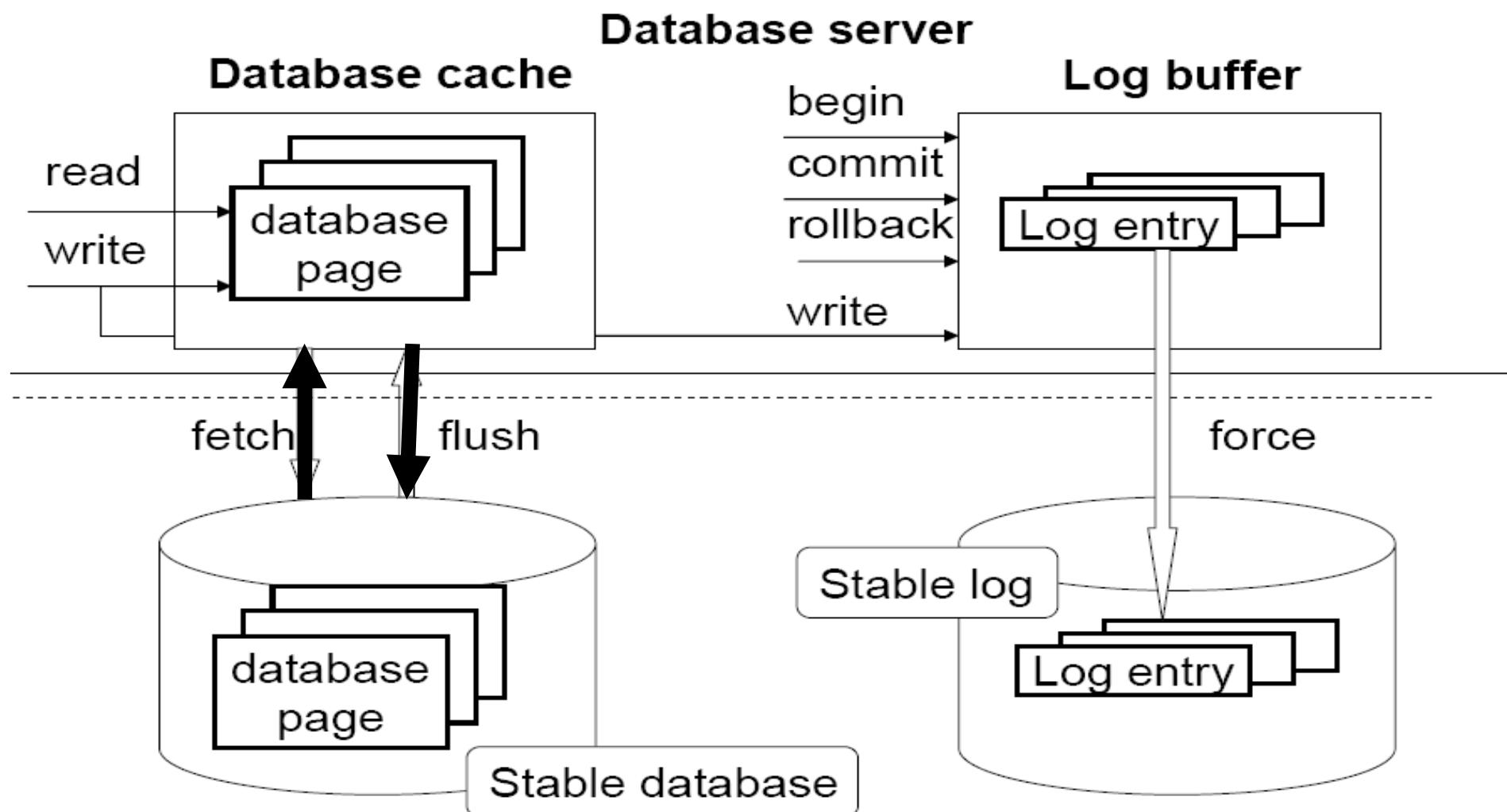
- Plik bazy danych podzielony na stałej długości jednostki pamięci zwane **blokami**
 - jednostki zarówno alokacji pamięci jak i transferu danych.
- Cel SZBD: minimalizacja liczby transferów bloków między dyskiem a pamięcią
 - podejście: przechowywanie jak największej liczby bloków w pamięci głównej.
- **Bufor (pula buforów)** – porcja pamięci głównej przeznaczona do przechowywania kopii bloków dyskowych.
- **Zarządca bufora (buffer manager)** – podsystem odpowiedzialny za alokację przestrzeni buforów w pamięci głównej.

Zarządca bufora

- Programy wysyłają żądanie do zarządcy bufora gdy potrzebują blok z dysku.
 1. Jeżeli blok jest już w buforze, zarządca zwraca adres bloku w pamięci głównej
 2. Jeżeli nie, to zarządca:
 1. alokuje przestrzeń w puli buforów dla bloku
 1. Wymiana (wyrzucenie) jakiegoś innego bloku, w razie potrzeby, aby zrobić miejsce na nowy blok
 2. Zastąpiony blok zapisywany na dysk tylko wtedy gdy został zmodyfikowany od ostatniego czasu gdy został zapisany/pobrany na/z dysk
 2. czyta blok z dysku do bufora i zwraca adres bloku w pamięci głównej żądającemu.

- Gdy brakuje miejsca w buforze – najczęściej usuwany blok najdawniej używany (**LRU strategy**)
- Przypięty blok (**pinned block**) – blok pamięci, którego nie można zapisać z powrotem na dysk.
- Zarządcą buforów wspiera również **wymuszony zapis (forced output)** bloków, nawet jeżeli nie brakuje miejsca w buforze

Buforowanie bloków dyskowych





AGH

Tworzenie indeksu

- » Wiele zapytań odnosi się tylko do niewielkiej części rekordów w tabeli.
- » Nieefektywne jest odczytywanie przez system każdego rekordu w celu znalezienia rekordu o określonej wartości
- » **Indeks** na atrybucie relacji to struktura danych, która umożliwia systemowi bazodanowemu efektywne znajdowanie tych krotek w relacji, które mają określoną wartość dla tego atrybutu bez przeglądania wszystkich krotek relacji.
- » Polecenie **create index** do tworzenia indeksu

create index <name> on <relation-name> (attribute);



AGH

Tworzenie indeksu przykład

- » **create table student**
*(ID varchar (5),
name varchar (20) not null,
dept_name varchar (20),
tot_cred numeric (3,0) default 0,
primary key (ID))*
- » **create index studentID_index on student(ID)**
- » Zapytanie:

```
select *
from student
where ID = '12345'
```

może być wykonane przy użyciu indeksu bez przeglądania wszystkich rekordów relacji *student*



AGH

Rodzaje indeksów

» Klastrowy (*clustered*, grupujący)

```
CREATE UNIQUE CLUSTERED INDEX nazwa_indeksu ON  
nazwa_tabeli(kolumna)
```

- podobny do książki telefonicznej
- nie jest osobnym obiektem
- narzuca tabelom, jak mają zapisywać fizycznie rekordy na kolejnych stronach
- po utworzeniu indeksu struktura stron ulega reorganizacji wg wskazań indeksu
- 1 indeks klastrujący/1 tabelę
- max na 1 kolumnie
- zazwyczaj PK jako indeks klastrowy



AGH

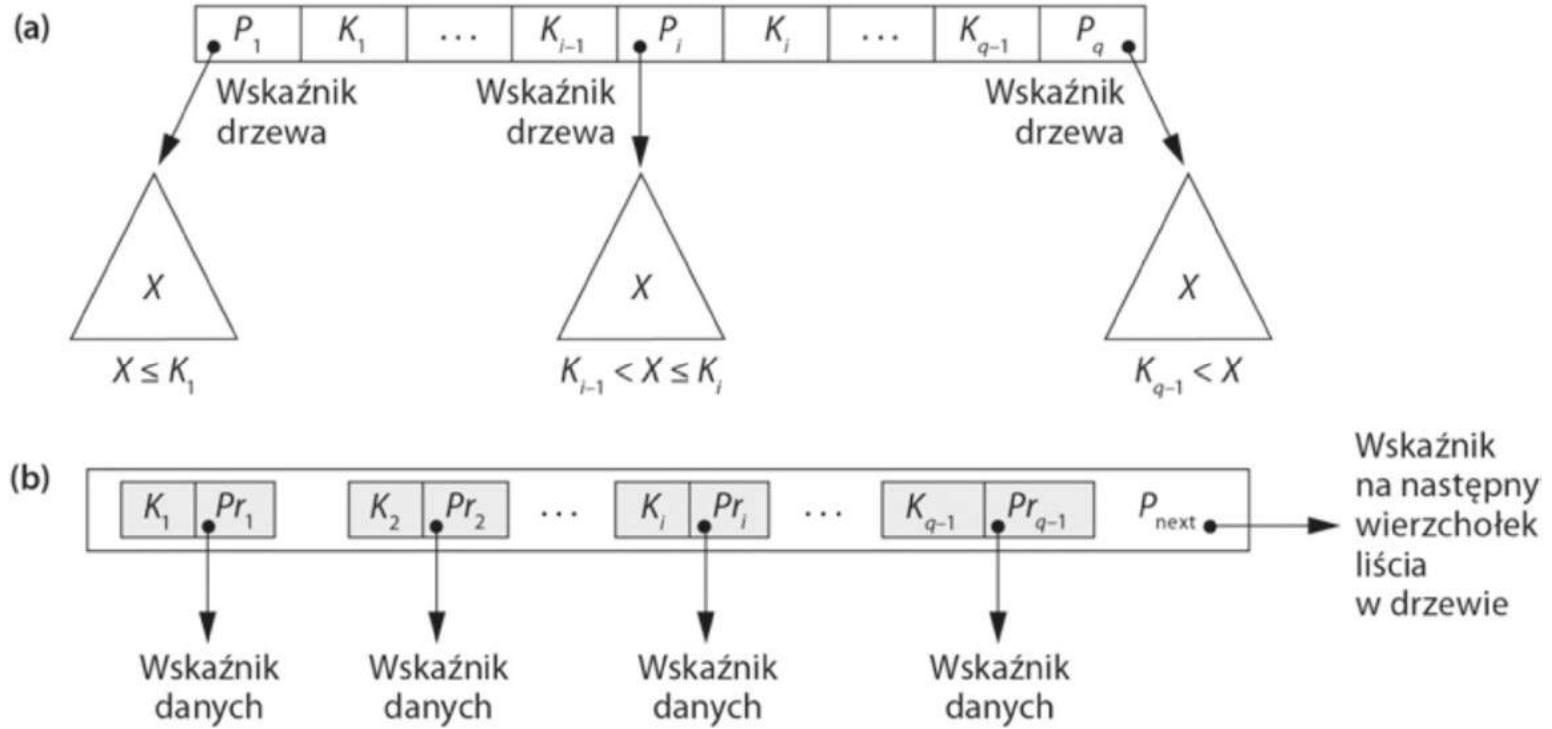
Rodzaje indeksów c.d.

» Nieklastrowy (*nonclustered*, niegrupujący)

```
CREATE NONCLUSTERED INDEX nazwa_indeksu ON  
nazwa_tabeli(kolumna)
```

- zapisywany jako osobny obiekt (jak indeks słów kluczowych na końcu książki)
- wskazuje na dokładnie 1 rekord w tabeli
- może być zakładany na >1 kolumnie
- nie wpływa na organizację fizyczną tabeli => wiele indeksów/tabelę

Struktura B+-drzewa



Uwagi o wykorzystywaniu indeksów

- » Indeksy powinno zakładać się na kolumnach rzadko modyfikowanych.
- » Kolumny typu PRIMARY/FOREIGN KEY powinny zawierać indeksy: klastrowy i nieklastrowy.
- » Efektywność wykorzystania indeksów zależy od selektywności zapytania – jak duży procent wierszy przeszukiwanej tabeli zwraca zapytanie:
 - jeżeli zapytanie zwraca nie więcej, niż 1% wierszy, warto używać indeksów
 - jeżeli zapytanie zwraca np. więcej, niż 20% wierszy, należy zastanowić się na indeksami: strategia pełnego odczytu danych z tabeli może okazać się bardziej efektywna
 - wpływ na to co się dzieje może mieć wiele czynników: np historia usuwania/dodawania nowych wierszy do tabeli, a nawet historii odczytywań danych z tabeli przez bieżący wątek kliencki.

Uwagi o wykorzystywaniu indeksów c.d.

- » Samo zakładanie indeksów na wielkich tabelach może być kosztowne
- » Dodatkowo każdy indeks powoduje, że proces dodawania/usuwania krotek z tabeli może być znacznie wolniejszy
- » Indeksów nie należy więc zakładać “bo tak”, ale tylko wówczas, gdy ich przewidywane wykorzystanie przyspieszy działanie bazy, to znaczy działanie często wykonywanych/kluczowych zapytań

Dostrajanie indeksu

- » Przyczyny:
 - niektóre zapytania są wykonywane zbyt długo z powodu braku indeksu
 - niektóre indeksy w ogóle nie są używane
 - niektóre indeksy są zbyt często aktualizowane, ponieważ indeks dotyczy często modyfikowanego atrybutu
- » stosowanie w zapytaniach wskazówek będących sugerowanymi alternatywami lub sygnałami dla procesora i optymalizatora zapytań, określającymi, jak wykonać zapytanie (np. Oracle)

```
SELECT /*+INDEX(PRACOWNIK PRAC_NRDZ_INDEKS)*/ PESEL,  
      PENSJA,NRDZ  
  FROM PRACOWNIK  
 WHERE NRDZ<10
```



AGH

Wprowadzenie do bezpieczeństwa baz danych

Kwestie bezpieczeństwa

- » *Kwestie prawne i etyczne dotyczące dostępności informacji*
- » *Polityki bezpieczeństwa na poziomie państwowym, instytucjonalnym lub firmowym dotyczące danych, które nie mogą być publicznie ujawniane (wyciągi bankowe, dane medyczne,...)*
- » *Kwestie systemowe, takie jak systemowe poziomy zabezpieczeń, na których wdrażane są różne funkcje kontroli dostępu (zabezpieczenia realizowane na poziomie sprzętu, systemu operacyjnego, SZBD...)*
- » W niektórych organizacjach: potrzeba wprowadzenia wielu różnych poziomów dostępu oraz klasyfikacji w stosunku do ich wszystkich danych i użytkowników

Zagrożenia wobec baz danych

- » *Utrata integralności*
 - dane mają być zabezpieczone przed niewłaściwymi modyfikacjami (tworzenie, dodawanie, aktualizacje, zmianę statusu danych, usuwanie)
- » *Utrata dostępności*
 - możliwość uzyskania przez człowieka lub program dostępu do obiektów, do których odczytywania jest uprawiony
- » *Utrata poufności*
 - ochrona danych przed nieautoryzowanymi ujawnieniami

Bezpieczeństwo baz danych – nieodizolowane zagadnienie

- » Baza danych – część sieci usług (aplikacji, serwerów WWW, zapór, mechanizmów przerywania połączeń SSL, systemów monitorowania bezpieczeństwa)
- » Do zapewnienia bezpieczeństwa danych w bazie danych stosuje się 4 podstawowe środki kontroli:
 - *kontrola dostępu*
 - *kontrola wnioskowania*
 - *kontrola przepływu*
 - *szafrowanie danych*

Podsystem uwierzytelniania i zabezpieczeń (SZBD)

- » Odpowiedzialny za zapewnienie bezpieczeństwa fragmentów bazy danych przed nieautoryzowanym dostępem
- » Mechanizmy zabezpieczeń bazy danych:
 - *dyspozycyjny mechanizm bezpieczeństwa*
 - nadawanie użytkownikom uprawnień określających dostęp do określonych plików, rekordów i atrybutów w określonym trybie (odczytu, dodawania, usuwania lub modyfikacji)
 - *obowiązkowy mechanizm bezpieczeństwa*
 - wprowadzanie wielu poziomów bezpieczeństwa poprzez przypisywanie danych i użytkowników do różnych klas (poziomów) bezpieczeństwa
 - mechanizm oparty o role

Środki kontroli

1. *Kontrola dostępu*: konta i hasła użytkowników
2. *Kontrola wnioskowania*: dostęp do danych statystycznych, ale nie do szczegółowych danych
3. *Kontrola przepływu*: taki przepływ danych, który pozwalałby na dostęp do nich nieautoryzowanemu użytkownikowi – kanały ukryte
4. *Szyfrowanie danych*: nieautoryzowany użytkownik po uzyskaniu dostępu do systemu bazy danych nie będzie mógł ich odszyfrować, w odróżnieniu od autoryzowanych użytkowników – szyfrowanie z użyciem klucza publicznego i podpisów cyfrowych

Bezpieczeństwo a administrator bazy danych (DBA)

- » Konto administratora (systemowe, superużytkownika)
 - 1. tworzenie kont (ograniczanie dostępu do całego systemu bazy danych)
 - 2. nadawanie uprawnień (określanie dyspozycyjnej autoryzacji)
 - 3. odbieranie uprawnień (określanie dyspozycyjnej autoryzacji)
 - 4. określanie poziomów bezpieczeństwa (określanie autoryzacji obowiązkowej)

Ochrona dostępu, konta użytkowników, audyty bazy danych

- » Stworzenie konta i hasła
 - programy komputerowe traktowane jak użytkownicy
- » Kontrolowanie listy użytkowników oraz ich kont i haseł poprzez stworzenie zaszyfrowanej tabeli lub pliku:
 - *IdentyfikatorKonta* i *Hasło*
- » Przechowywanie zapisu wszystkich operacji wykonanych przez danego użytkownika podczas sesji logowania: *dziennik systemowy*
- » Przy podejrzeniu jakiegokolwiek zfałszowania danych w bazie danych – *audyt bazy danych*

Dane wrażliwe

- » *Poziom wrażliwości danych:* miara znaczenia danych przypisywana im przez właściciela w celu określenia potrzeby ochrony
- » Dane mogą zostać uznane za wrażliwe z kilku powodów:
 - dane z natury wrażliwe (wynagrodzenie pracownika)
 - dane z wrażliwego źródła
 - opisane jako wrażliwe
 - z wrażliwymi atrybutami lub rekordami
 - wrażliwe w połączeniu z wcześniej ujawnionymi danymi



AGH

Typy ujawnień

- » Czy poszczególnym użytkownikom lub ich grupom należy zapewnić dostęp do określonego elementu tabeli lub elementu danych
- » Czynniki:
 - *dostępność danych* - blokady
 - *akceptowalność dostępu*
 - *wiarygodność uwierzytelnienia*
- » *Zabezpieczenia* – środki chroniące dane przed uszkodzeniem i zapewniające odpowiednią kontrolę dostępu do danych
- » *Precyzja* – ochrona wszystkich wrażliwych danych przy jednoczesnym ujawnianiu lub udostępnianiu możliwie wielu zwykłych danych
- » *Cel*: zapewnienie pełnego bezpieczeństwa przy maksymalnej precyzji

Dyspozycyjna kontrola dostępu – nadawanie/odbieranie uprawnień

- » SQL2 – identyfikator uwierzytelnienia (=użytkownik=konto)
- » Poziomy nadawania uprawnień:
 - *poziom konta* – DBA określa uprawnienia przypisane do konta, niezależnie od relacji w bazie danych
 - *poziom relacji* (tabeli) – DBA może kontrolować dostęp do każdej relacji lub pespektywy w strukturze bazy danych

Uprawnienia na poziomie konta

- » Dotyczą możliwości samego konta, mogą zawierać uprawnienia:
 - CREATE SCHEMA, CREATE TABLE
 - CREATE VIEW
 - ALTER
 - DROP
 - MODIFY
 - SELECT
- » Uprawnienia dotyczą całego konta
 - brak CREATE TABLE – nie będzie mógł tworzyć żadnych relacji

Uprawnienia na poziomie relacji

- » Dotyczą rzeczywistych i wirtualnych relacji
- » Określają jakie typy komend mogą być wykonywane przez określonych użytkowników na określonych relacjach
 - dotyczą również pojedynczych kolumn
 - SQL2 – uprawnienia tylko na poziomie relacji i atrybutów
- » Nadawanie/cofanie poszczególnych uprawnień wg modelu macierzy dostępu M
 - wiersze M przedstawiają *podmioty* uprawnień (użytkowników, konta, programy)
 - kolumny odpowiadają ich *przedmiotom* (relacjom, kolumnom, perspektywom, operacjom)
 - $M(i,j)$ określa typ uprawnień (odczyt, zapis, zmiana) podmiotu i do przedmiotu j

Przypisywanie uprawnień

- » Każda relacja R ma określonego właściciela
 - nadawane są mu wszystkie uprawnienia do niej
 - może on przekazać swoje uprawnienia do posiadanych relacji innym użytkownikom poprzez *nadanie* uprawnień ich kontom
- » Dla każdej relacji R można przypisać następujące uprawnienia:
 - SELECT (odczyt) do relacji R
 - MODIFY do relacji R – UPDATE, DELETE, INSERT
 - REFERENCES do relacji R
- » Do stworzenia perspektywy użytkownik musi mieć uprawnienia SELECT do *wszystkich* relacji wykorzystywanych w jego definicji

Określanie uprawnień przy użyciu perspektywy

- » Jeżeli właściciel A relacji R chce udostępnić użytkownikowi B tylko niektóre atrybuty swojej relacji, to A może utworzyć perspektywę V zawierającą wybrane atrybuty R
 - a następnie nadać użytkownikowi B uprawnienia SELECT do perspektywy V
- » Ograniczenia dostępu B do niektórych krotek R
 - nowa perspektywa V' zawierająca tylko te krotki R które mają być dostępne dla B



AGH

Cofanie uprawnień

- » Pewne uprawnienia tylko czasowo
- » REVOKE

Propagacja uprawnień

- » B ma nadane przez A uprawnienia z opcją GRANT
 - następnie nadaje swoje uprawnienia do relacji R kolejnemu użytkownikowi C również z opcją GRANT
 - uprawnienia do R mogą być propagowane na inne konta bez wiedzy właściciela relacji
 - jeżeli A anuluje uprawnienia nadane B , to wszystkie propagowane za pośrednictwem B uprawnienia powinny zostać *automatycznie cofnięte* przez system
- » Użytkownik może dostać uprawnienia do tej samej relacji z dwóch różnych źródeł

Przykład

» konta: A1, A2, A3, A4 ale tylko A1 będzie miał możliwość tworzenia relacji w bazie danych:

- GRANT CREATETAB TO A1 – uprawnienie na poziomie konta

» SQL2:

- CREATE SCHEMA PRZYKŁAD AUTHORIZATION A1

PRACOWNIK						
IMIĘNAZWISKO	PESEL	DATAUR	ADRES	PŁEĆ	PENSJA	NRDZ

DZIAŁ		
NUMERDZ	NAZWADZ	PESELKIEROWNIKA

- A1: GRANT INSERT, DELETE ON PRACOWNIK,DZIAŁ TO A2
- A1: GRANT SELECT ON PRACOWNIK,DZIAŁ TO A3 WITH GRANT OPTION
- A3: GRANT SELECT ON PRACOWNIK TO A4
- A1: REVOKE SELECT ON PRACOWNIK FROM A3
- Wycofanie uprawnień dla A3 i A4



AGH

Przykład c.d.

- » A1 chce nadać uprawnienia A3 do odczytu tylko atrybutów IMIĘNAZWISKO, DATAUR i ADRES, ale dla krotek o wartości NRDZ równej 5:

```
A1: CREATE VIEW A3PRACOWNIK AS  
      SELECT IMIĘNAZWISKO, DATAUR, ADRES  
            FROM PRACOWNIK  
           WHERE NRDZ=5
```

- A1: GRANT SELECT ON A3PRACOWNIK TO A3 WITH GRANT OPTION
- A1: GRANT UPDATE ON PRACOWNIK(PENSJA) TO A4

Obowiązkowa kontrola dostępu

- » Mechanizm zabezpieczeń dyspozycyjnych (nadawanie/odbieranie uprawnień) – tradycyjnie główny mechanizm kontroli dostępu w RSBD
 - „wszystko albo nic”
- » W wielu aplikacjach istnieje potrzeba wprowadzenia *dodatkowej polityki bezpieczeństwa*
 - klasyfikacja danych i użytkowników w oparciu o klasy bezpieczeństwa – obowiązkowa kontrola dostępu – często łączona z zabezpieczeniami dyspozycyjnymi
 - ścisłe tajne, tajne, poufne, jawne

Dyspozycyjny a obowiązkowy model bezpieczeństwa

- » Dyspozycyjna
 - zalety: wysoki poziom elastyczności
 - wada: podatność na ataki (konie trojańskie zaszyte w kodzie aplikacji)
 - model nie kontroluje rozpowszechniania ani używania danych po uzyskaniu do nich dostępu przez użytkownika
- » Obowiązkowa
 - zaleta: zapewnia wysoki stopień zabezpieczenia poprzez zablokowanie niepożądanego przepływu danych
 - wada: zbyt sztywne, bo wymagają przydzielania wszystkich podmiotów i przedmiotów działania bazy danych do określonych klas bezpieczeństwa

Kontrola dostępu oparta na rolach

- » Przypisywanie uprawnień określonym rołom (a nie poszczególnym użytkownikom)
 - powiązanie użytkowników z rolami
- » CREATE ROLE (DROP ROLE)
 - nadawanie/odbieranie uprawnień rołom: GRANT/REVOKE
- » Do roli można przypisać wiele osób
- » Użytkownicy nawiązują sesje, w ramach których mogą aktywować podzbior przypisanych im ról
 - z każdą sesją może być powiązanych wiele ról, ale tylko 1 użytkownik



AGH

Podział zadań

- » Cel: zapobiec wykonywaniu przez 1 użytkownika operacji wymagających udziału przynajmniej 2 (zapobiegnie oszustwom)
- » *Wzajemnie wykluczanie się ról*
 - nie mogą być jednocześnie wykorzystywane przez tego samego użytkownika
 - *statyczne* (na etapie uwierzytelniania)
 - *dynamiczne* (na etapie wykonywania)

Hierarchia ról

- » Naturalny sposób organizowania ról w celu odzwierciedlenia łańcuchów odpowiedzialności i władzy w organizacji
 - mniej ważne role u dołu połączone z coraz istotniejszymi w miarę przesuwania się w górę hierarchii
 - częściowy porządek – jeżeli użytkownik ma daną rolę, automatycznie otrzymuje role z niższych poziomów hierarchii

GRANT ROLE pracownik_etatowy TO typ_pracownika1

GRANT ROLE stażysta TO typ_pracownika2



AGH

Systemy oparte o role - cechy

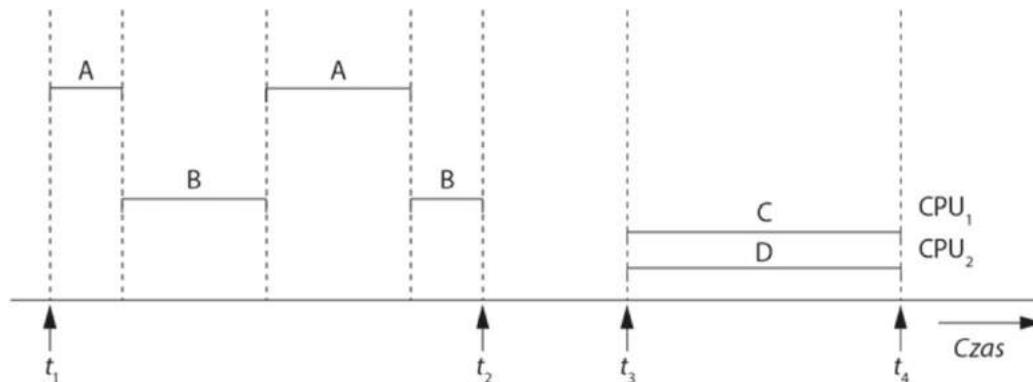
- » Czasowe ograniczenia

- czas i okres trwania aktywacji ról
 - uruchamianie roli na pewien okres poprzez aktywację innej roli
 - przypisywanie ról kolejnym etapom ciągu prac

Wprowadzenie do problematyki przetwarzania transakcji

Klasyfikacja systemów bazodanowych

- » Według liczby użytkowników, którzy mogą z systemu korzystać współbieżnie:
 - jednoużytkownikowe
 - wieloużytkownikowe
- » Pojęcie *wieloprogramowości*
 - przetwarzanie z przeplotem
- » Kilką procesorów sprzętowych – *równoległe* przetwarzanie



Transakcja

- » Wykonywany program, który tworzy logiczną jednostkę przetwarzania w bazie danych
 - składa się z jednej lub wielu operacji dostępu do bazy danych (wstawianie/usuwanie/modyfikowanie/pobieranie danych)
 - mogą być zawarte w programie aplikacji lub określone interaktywnie za pomocą SQL (np. poprzez jawne *begin transaction...end transaction*)
- » Jeżeli operacje bazodanowe należące do transakcji nie aktualizują bazy danych – *transakcja tylko do odczytu*
 - pozostałe: *transakcje do odczytu i zapisu*

Przetwarzanie transakcji

- » Baza danych – zbiór *nazwanych elementów danych*
 - rekord w bazie danych, cały blok dyskowy, wartość pojedynczego pola rekordu z bazy danych
 - ma unikatową nazwę
 - rozmiar elementu danych: *ziarnistość*
- » Uproszczony model bazy danych – podstawowe operacje dostępu do bazy, które mogą należeć do transakcji:
 - *odczytaj_element(X)*: wczytuje element bazy danych o nazwie X do zmiennej programowej (X)
 - *zapisz_element(X)*: zapisuje wartość zmiennej programowej X w elemencie bazy danych o nazwie X

Wykonanie podstawowych operacji dostępu do bazy danych

» *odczytaj_element(X)*

1. znalezienie adresu bloku dyskowego, który zawiera element X
2. skopiowanie bloku dyskowego do bufora w pamięci głównej
3. skopiowanie elementu X z bufora do zmiennej programowej X

» *zapisz_element(X)*

1. znalezienie adresu bloku dyskowego, który zawiera element X
2. skopiowanie tego bloku do bufora w pamięci głównej
3. skopiowanie elementu X ze zmiennej programowej o nazwie X do odpowiedniej lokalizacji w buforze
4. przeniesienie zaktualizowanego bloku z bufora z powrotem na dysk (od razu lub z pewnym opóźnieniem)

Transakcje - przykład

(a)

 T_1

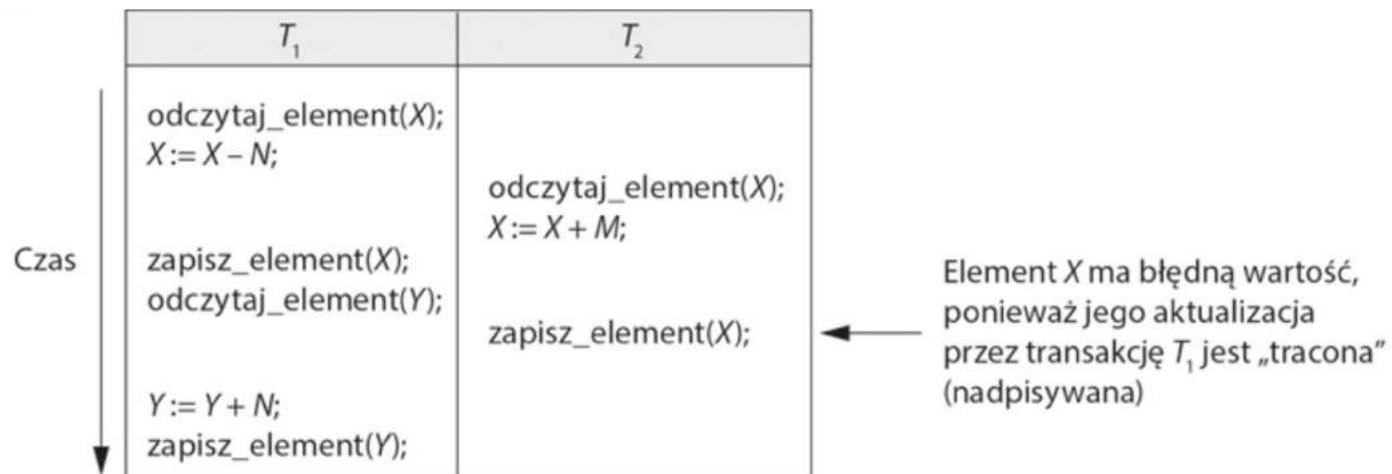
```
odczytaj_element(X);
X := X - N;
zapisz_element(X);
odczytaj_element(Y);
Y := Y + N;
zapisz_element(Y);
```

 T_2

```
odczytaj_element(X);
X := X + M;
zapisz_element(X);
```

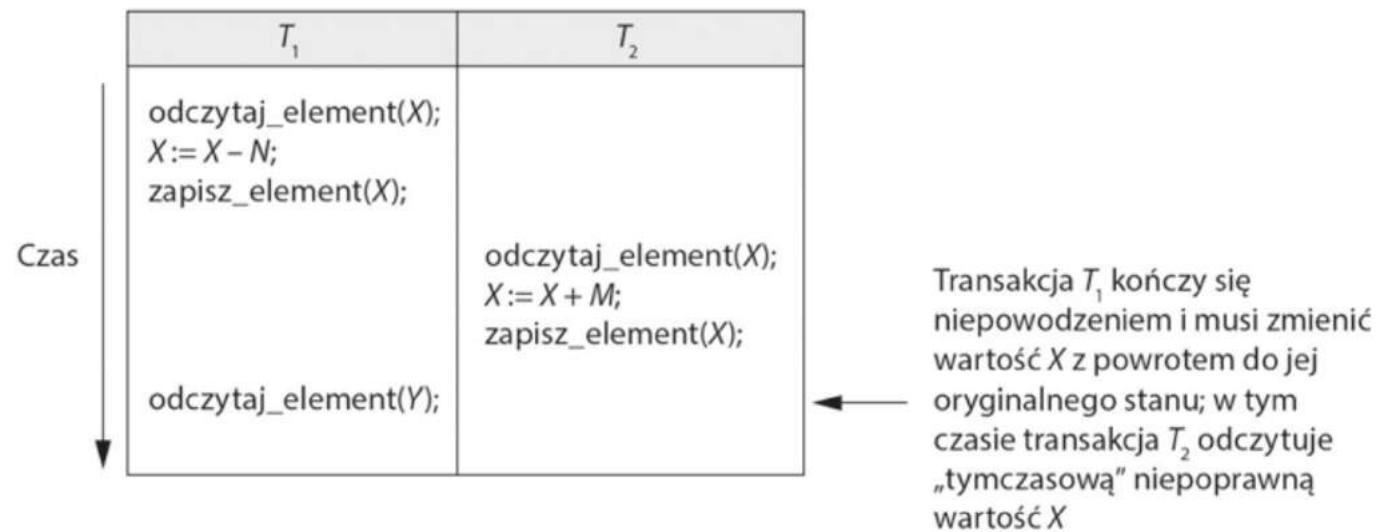
Problemy przy współbieżnym wykonywaniu transakcji

- » *Problem utraconej aktualizacji* – dwie transakcje uzyskujące dostęp do tych samych elementów są związane z przeplotem ich operacji



Problemy przy współbieżnym wykonywaniu transakcji c.d.

» *Problem aktualizacji tymczasowej (odczytu zmodyfikowanego)*



Problemy przy współbieżnym wykonywaniu transakcji c.d.

- » *Problem błędnej sumy* – jeżeli jedna transakcja oblicza wartość funkcji agregującej podsumowania na wielu rekordach, kiedy inną transakcją aktualizuje niektóre z tych rekordów, funkcja aggregująca może obliczyć pewne wartości przed tym, jak zostaną zaktualizowane, a inne już po ich aktualizacji.

T_1	T_3
<pre>odczytaj_element(X); X := X - N; zapisz_element(X); odczytaj_element(Y); Y := Y + N; zapisz_element(Y);</pre>	<pre>suma := 0; odczytaj_element(A); suma := suma + A; ⋮ odczytaj_element(X); suma := suma + X; odczytaj_element(Y); suma := suma + Y;</pre>

Transakcja T_3 odczytuje X po odjęciu N i odczytuje Y przed dodaniem N ; w wyniku otrzymuje błędą sumę (o N)

Problemy przy współbieżnym wykonywaniu transakcji c.d.

- » *Problem odczytu niepowtarzalnego* – transakcja T odczytuje element dwukrotnie i element ten zostaje zmieniony przez inną transakcję T' między tymi dwoma odczytami
 - T otrzymuje *różne wartości* dla swoich dwóch różnych odczytów tego samego elementu

Rola SZBD

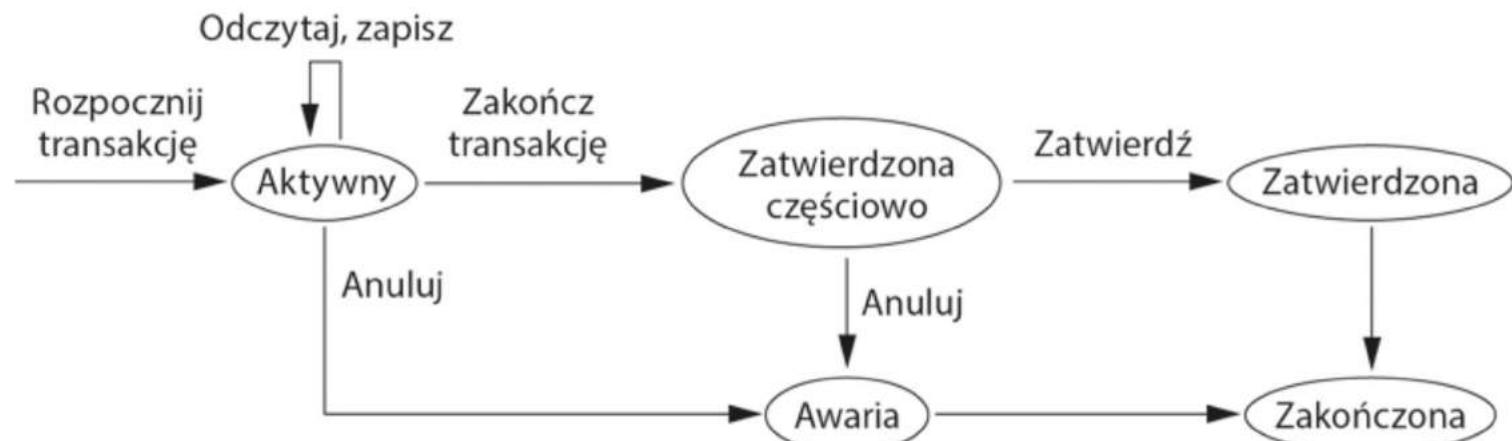
- » SZBD odpowiedzialny za zapewnienie aby wszystkie operacje zostały zakończone z powodzeniem
 - efekty ich działania zapisane na stałe w bazie danych (transakcja została *zatwierdzona*)
 - albo
 - transakcja nie miała wpływu na bazę danych i inne transakcje (transakcja *anulowana*)
- » SZBD nie może dopuścić do sytuacji gdy pewne operacje transakcji T zostaną uwzględnione w bazie danych, a inne nie
 - *cała transakcja* jest logiczną jednostką przetwarzania w bazie danych

Rodzaje awarii

- » Awaria: transakcyjne, systemowe, nośnikowe
- » Przyczyny awarii:
 1. *Awaria komputera (załamanie systemu)*
 2. *Błąd transakcyjny lub systemowy*
 3. *Błędy lokalne lub wyjątki wykryte przez transakcję*
 4. *Wymuszanie sterowania współbieżnego*
 5. Awaria dysku
 6. Problemy i katastrofy fizyczne

Stany transakcji – diagram przejść stanów

- » ROCZPOCZNIJ_TRANSAKCJĘ (BEGIN_TRANSACTION) – oznaczenie początku
- » ODCZYTAJ (READ) lub ZAPISZ (WRITE) – operacje odczytu/zapisu na elementach BD
- » ZAKOŃCZ_TRANSAKCJE (END_TRANSACTION) – wszystkie operacje ODCZYTAJ i ZAPISZ zostały zakończone oraz zakończono wykonywanie transakcji
- » ZATWIERDŹ_TRANSAKCJĘ (COMMIT_TRANSACTION) – *udane zakończenie* transakcji – wszystkie zmiany dokonane przez transakcję mogą być bezpiecznie zatwierdzone (*commit*)
- » WYCOFAJ (ROLLBACK), ANULUJ (ABORT) – *nieudane zakończenie* transakcji – wszystkie zmiany spowodowane przez transakcję w BD muszą zostać wycofane (*rollback*)



Dziennik systemowy

- » Plik sekwencyjny – dane tylko dodawane
 - bufory dziennika w pamięci głównej – przechowują najnowszy fragment tego pliku
 - gdy się zapełni – bufor dołączany na końcu pliku dziennika na dysku
 - okresowa archiwizacja
- » Rodzaje wpisów (rekordy dziennika)
 1. [rozpocznij_transakcję, T]
 2. [zapisz_element, $T, X, stara_{wartość}, nowa_{wartość}$]
 3. [odczytaj_element, T, X]
 4. [zatwierdź, T]
 5. [anuluj, T]
 - brak rejestracji odczytu

Dziennik systemowy c.d.

- » W razie awarii systemu można dokonać odtworzenia do spójnego stanu bazy danych poprzez sprawdzenie dziennika
 - można *cofnąć* efekty operacji T poprzez przejrzenia dziennika i odtworzenie wartości wszystkich elementów zmienionych poprzez operacje zapisu T na wartości określone przez parametr *stara_wartość*
 - może być również konieczne *ponowienie* operacji transakcji, jeżeli wszystkie jej aktualizacje zostaną zarejestrowane w dzienniku, ale wystąpi uszkodzenie zanim będzie możliwe upewnienie się, że wszystkie wartości *nowa_wartość* zostały trwale zapisane w BD na dysku

Punkt zatwierdzenia transakcji

- » Transakcja T osiąga swój *punkt zatwierdzenia* gdy:
 - wszystkie jej operacje zostaną w powodzeniu wykonane (efekty na trwale zapisane w BD) oraz
 - wyniki działania wszystkich operacji T zostaną zarejestrowane w dzienniku (rekord [zatwierdź, T])
- » Gdy awaria:
 - należy przeszukać wstecz dziennik w poszukiwaniu wszystkich T , które zapisały rekord [rozpocznij_transakcję, T], ale nie zapisały [zatwierdź, T]
 - może trzeba będzie je wycofać w celu usunięcia skutków ich działania
- » *Zapis wymuszony* bufora dziennika przed zatwierdzeniem transakcji

Pożądane właściwości transakcji

- » ACID – wymuszane przez mechanizm terowania współbieżnego oraz metody odtwarzania SZBD
 - *Niepodzielność* (atomicity) – transakcja jest niepodzielną jednostką przetwarzania: jest wykonywana albo w całości albo wcale
 - *Zachowanie spójności* (consistency preservation) – transakcja powinna zachować spójność: jeżeli zostaje wykonana od początku do końca bez zakłóceń ze strony innych transakcji, przenosi BD z jednego stanu spójnego do innego
 - *Izolacja* (isolation) – transakcja powinna wyglądać tak jakby była wykonywana w izolacji od innych transakcji: wykonywanie transakcji nie powinno kolidować ze współbieżnym wykonywaniem innych transakcji
 - *Trwałość* (durability) – zmiany zastosowane względem BD przez zatwierdzone transakcje muszą być trwałe – nie mogą zostać utracone w wyniku jakiejkolwiek awarii

Poziomy izolacji

- » Transakcja posiada zerowy *poziom izolacji* jeżeli nie nadpisuje zmodyfikowanych odczytów transakcji
- » *Izolacja poziomu pierwszego* – brak utraconych aktualizacji
- » *Izolacja poziomu drugiego* – nie posiada ani utraconych aktualizacji, ani zmodyfikowanych odczytów
- » *Izolacja poziomu trzeciego* (prawdziwa izolacja) – poziom drugi + odczyty powtarzalne
- » *Izolacja snapshotów*

Harmonogramy (historie) transakcji

- » *Harmonogram S* zbioru n transakcji T_1, T_2, \dots, T_n jest uporządkowaniem operacji transakcji
 - operacje z różnych transakcji mogą się przeplatać w S , ale
 - dla każdej transakcji T_i należącej do S operacje tej transakcji w S muszą występować w tej samej kolejności w jakiej występują w T_i

	T_1	T_2
Czas	\downarrow odczytaj_element(X); $X := X - N$; zapisz_element(X); odczytaj_element(Y); $Y := Y + N$; zapisz_element(Y);	\downarrow odczytaj_element(X); $X := X + M$; zapisz_element(X);

$S_a: r_1(X); r_2(X); w_1(X); r_1(Y); w_2(X); w_1(Y);$

	T_1	T_2
Czas	\downarrow odczytaj_element(X); $X := X - N$; zapisz_element(X); \downarrow odczytaj_element(Y);	\downarrow odczytaj_element(X); $X := X + M$; zapisz_element(X);

$S_b: r_1(X); w_1(X); r_2(X); w_2(X); r_1(Y); a_1;$

Operacje powodujące konflikt w harmonogramie

- » Dwie operacje w harmonogramie są w stanie *konfliktu*, jeżeli spełniają wszystkie warunki:
 1. należą do *różnych transakcji*
 2. uzyskują dostęp do *tego samego elementu X*
 3. *przynajmniej jedna jest operacją zapisz_element(X)*
$$S_a: r_1(X): r_2(X): w_1(X): r_1(Y): w_2(X): w_1(Y)$$
- » Dwie operacje powodują konflikt, jeżeli zmiana ich kolejności może skutkować zmianą wyniku
 - $(r_1(X): w_2(X))$ na $(w_2(X): r_1(X))$ – *konflikt odczytu i zapisu*
 - $(w_1(X): w_2(X))$ na $(w_2(X): w_1(X))$ – *konflikt zapisu i zapisu*

Charakterystyka harmonogramów wg możliwości odtwarzania

- » Odtwarzanie harmonogramów po awarii transakcji może być łatwe, złożone albo niemożliwe
- » Harmonogramy
 - odtwarzalne
 - bezkaskadowe
 - ścisłe

Harmonogramy odtwarzalne

» Warunek odtwarzalności harmonogramu

- harmonogram S jest odtwarzalny, jeżeli żadna transakcja T w S nie jest zatwierdzona do momentu aż wszystkie transakcje T' , które zapisyły element odczytywany przez transakcję T zostaną zatwierdzone
- Transakcja T czyta z transakcji T' w harmonogramie S jeżeli pewien element X jest najpierw zapisywany przez T' a później odczytywany przez T
- Transakcja T' nie powinna zostać anulowana zanim transakcja T odczyta element X i nie powinny występować żadne transakcje zapisujące X po dokonaniu zapisu przez T' a przed wykonaniem odczytu przez T

$S_a: r_1(X); r_2(X); w_1(X); r_1(Y); w_2(X); w_1(Y);$

$S_b: r_1(X); w_1(X); r_2(X); w_2(X); r_1(Y); a_1$

Harmonogramy odtwarzalne c.d.

$S'_a: r_1(X); r_2(X); w_1(X); r_1(Y); w_2(X); c_2; w_1(Y); c_1;$

$S_c: r_1(X); w_1(X); r_2(X); r_1(Y); w_2(X); c_2; a_1;$

$S_d: r_1(X); w_1(X); r_2(X); r_1(Y); w_2(X); w_1(Y); c_1; c_2;$

$S_e: r_1(X); w_1(X); r_2(X); r_1(Y); w_2(X); w_1(Y); a_1; a_2;$

Harmonogramy bezkaskadowe

- » *Wycofanie (anulowanie) kaskadowe*

- niezatwierdzona transakcja musi zostać wycofana, ponieważ odczytała element transakcji, która zakończyła się niepowodzeniem
 - $S_e: r_1(X); w_1(X); r_2(X); r_1(Y); w_2(X); w_1(Y); a_1; a_2;$

- » *Harmonogram bezkaskadowy*

- każda transakcja w harmonogramie odczytuje tylko te elementy, które zostały zapisane przez zatwierdzone transakcje

Harmonogramy ścisłe

- » Transakcje nie mogą *ani odczytywać, ani zapisywać* elementu X do momentu aż zostanie zatwierdzona (anulowana) ostatnia transakcja, która go zapisała
 - upraszczają proces odtwarzania: proces cofnięcia operacji `zapisz_element(X)` anulowanej transakcji polega na zwykłym odtworzeniu *obrazu pierwotnego* (before image), czyli *stara_wartość*, elementu danych X
 $S_f: w_1(X,5); w_2(X,8); a_1;$
- » Każdy harmonogram ścisły jest bezkaskadowy
- » Wszystkie bezkaskadowe są odtwarzalne
- » Większość protokołów odtwarzania dopuszcza tylko harmonogramy ścisłe