# Dobry projekt bazy danych

## Aspekty

# Dobry projekt bazy danych

## Aspekty

NULL-e — krotki powinny pasować do schematu tabeli,

# Dobry projekt bazy danych

## Aspekty

NULL-e — krotki powinny pasować do schematu tabeli,

Redundancja — informacja nie powinna być zapisywana wielokrotnie,

# Dobry projekt bazy danych

## Aspekty

- **NULL-e** — krotki powinny pasować do schematu tabeli,
- **Redundancja** — informacja nie powinna być zapisywana wielokrotnie,
- **Kontrola więzów** — sprawdzanie własności klucza, unikalności i innych więzów powinno być łatwe,

# Dobry projekt bazy danych

## Aspekty

**NULL-e** — krotki powinny pasować do schematu tabeli,

**Redundancja** — informacja nie powinna być zapisywana wielokrotnie,

**Kontrola więzów** — sprawdzanie własności klucza, unikalności i innych więzów powinno być łatwe,

**Obliczanie złączeń** — jest trudne, więc nie należy rozdrabniać zbytnio bazy.

# Dobry projekt bazy danych

## Aspekty

**NULL-e** — krotki powinny pasować do schematu tabeli,

**Redundancja** — informacja nie powinna być zapisywana wielokrotnie,

**Kontrola więzów** — sprawdzanie własności klucza, unikalności i innych więzów powinno być łatwe,

**Obliczanie złączeń** — jest trudne, więc nie należy rozdrabniać zbytnio bazy.

# Dobry projekt bazy danych

## Aspekty

**NULL-e** — krotki powinny pasować do schematu tabeli,

**Redundancja** — informacja nie powinna być zapisywana wielokrotnie,

**Kontrola więzów** — sprawdzanie własności klucza, unikalności i innych więzów powinno być łatwe,

**Obliczanie złączeń** — jest trudne, więc nie należy rozdrabniać zbytnio bazy.

OSOBA(ido,tytul,indeks,nazwisko,adres),

# Dobry projekt bazy danych

## Aspekty

- NULL-e — krotki powinny pasować do schematu tabeli,
- Redundancja — informacja nie powinna być zapisywana wielokrotnie,
- Kontrola więzów — sprawdzanie własności klucza, unikalności i innych więzów powinno być łatwe,
- Obliczanie złączeń — jest trudne, więc nie należy rozdrabniać zbytnio bazy.

OSOBA(ido,tytul,indeks,nazwisko,adres),

- Student ma indeks i nie ma tytułu; pracownik — odwrotnie:

# Dobry projekt bazy danych

## Aspekty

- **NULL-e** — krotki powinny pasować do schematu tabeli,
- **Redundancja** — informacja nie powinna być zapisywana wielokrotnie,
- **Kontrola więzów** — sprawdzanie własności klucza, unikalności i innych więzów powinno być łatwe,
- **Obliczanie złączeń** — jest trudne, więc nie należy rozdrabniać zbytnio bazy.

OSOBA(ido,tytul,indeks,nazwisko,adres),

- Student ma indeks i nie ma tytułu; pracownik — odwrotnie: PR(ido,tytuł,nazwisko,adres), ST(ido,indeks,nazwisko,adres)

# Dobry projekt bazy danych

## Aspekty

**NULL-e** — krotki powinny pasować do schematu tabeli,

**Redundancja** — informacja nie powinna być zapisywana wielokrotnie,

**Kontrola więzów** — sprawdzanie własności klucza, unikalności i innych więzów powinno być łatwe,

**Obliczanie złączeń** — jest trudne, więc nie należy rozdrabniać zbytnio bazy.

OSOBA(ido,tytul,indeks,nazwisko,adres), GRUPA(idg,idk,idp,termin,limit,sala)

- Student ma indeks i nie ma tytułu; pracownik — odwrotnie: PR(ido,tytuł,nazwisko,adres), ST(ido,indeks,nazwisko,adres)
- Grupa ma kilka terminów zajęć — kilkakrotne wpisanie grupy do tabeli oznacza redundancję (limit, idk,idp):

# Dobry projekt bazy danych

## Aspekty

NULL-e — krotki powinny pasować do schematu tabeli,

Redundancja — informacja nie powinna być zapisywana wielokrotnie,

Kontrola więzów — sprawdzanie własności klucza, unikalności i innych więzów powinno być łatwe,

Obliczanie złączeń — jest trudne, więc nie należy rozdrabniać zbytnio bazy.

OSOBA(ido,tytul,indeks,nazwisko,adres), GRUPA(idg,idk,idp,termin,limit,sala)

- Student ma indeks i nie ma tytułu; pracownik — odwrotnie: PR(ido,tytuł,nazwisko,adres), ST(ido,indeks,nazwisko,adres)
- Grupa ma kilka terminów zajęć — kilkakrotne wpisanie grupy do tabeli oznacza redundancję (limit, idk,idp): GR(idg,idk,idp,limit), TS(idg,termin,sala)

# Dobry projekt bazy danych

## Aspekty

- NULL-e — krotki powinny pasować do schematu tabeli,
- Redundancja — informacja nie powinna być zapisywana wielokrotnie,
- Kontrola więzów — sprawdzanie własności klucza, unikalności i innych więzów powinno być łatwe,
- Obliczanie złączeń — jest trudne, więc nie należy rozdrabniać zbytnio bazy.

OSOBA(ido,tytul,indeks,nazwisko,adres), GRUPA(idg,idk,idp,termin,limit,sala)

- Student ma indeks i nie ma tytułu; pracownik — odwrotnie: PR(ido,tytuł,nazwisko,adres), ST(ido,indeks,nazwisko,adres)
- Grupa ma kilka terminów zajęć — kilkakrotne wpisanie grupy do tabeli oznacza redundancję (limit, idk,idp): GR(idg,idk,idp,limit), TS(idg,termin,sala)
- Łatwo sprawdzić:
  - grupa ma jednego prowadzącego
  - studenci mają unikalne indeksy

# Dobry projekt bazy danych

## Aspekty

- **NULL-e** — krotki powinny pasować do schematu tabeli,
- **Redundancja** — informacja nie powinna być zapisywana wielokrotnie,
- **Kontrola więzów** — sprawdzanie własności klucza, unikalności i innych więzów powinno być łatwe,
- **Obliczanie złączeń** — jest trudne, więc nie należy rozdrabniać zbytnio bazy.

OSOBA(ido,tytul,indeks,nazwisko,adres), GRUPA(idg,idk,idp,termin,limit,sala)

- Student ma indeks i nie ma tytułu; pracownik — odwrotnie: PR(ido,tytuł,nazwisko,adres), ST(ido,indeks,nazwisko,adres)
- Grupa ma kilka terminów zajęć — kilkakrotne wpisanie grupy do tabeli oznacza redundancję (limit, idk,idp): GR(idg,idk,idp,limit), TS(idg,termin,sala)
- Łatwo sprawdzić:
  - grupa ma jednego prowadzącego
  - studenci mają unikalne indeksy
- Trudno sprawdzić, że identyfikatory osób są unikalne.

# Dobry projekt bazy danych

## Aspekty

**NULL-e** — krotki powinny pasować do schematu tabeli,

**Redundancja** — informacja nie powinna być zapisywana wielokrotnie,

**Kontrola więzów** — sprawdzanie własności klucza, unikalności i innych więzów powinno być łatwe,

**Obliczanie złączeń** — jest trudne, więc nie należy rozdrabniać zbytnio bazy.

OSOBA(ido,tytul,indeks,nazwisko,adres), GRUPA(idg,idk,idp,termin,limit,sala)

- Student ma indeks i nie ma tytułu; pracownik — odwrotnie: PR(ido,tytuł,nazwisko,adres), ST(ido,indeks,nazwisko,adres)
- Grupa ma kilka terminów zajęć — kilkakrotne wpisanie grupy do tabeli oznacza redundancję (limit, idk,idp): GR(idg,idk,idp,limit), TS(idg,termin,sala)
- Łatwo sprawdzić:
  - grupa ma jednego prowadzącego
  - studenci mają unikalne indeksy
- Trudno sprawdzić, że identyfikatory osób są unikalne.
- Wyznaczenie terminu i sali zajęć grupy wymaga złączenia.

# Definicja i przykład

## Definition (Zależność funkcyjna)

Dla relacji $R = A_1 A_2 \ldots A_k$ oraz zbiorów jej atrybutów $\alpha, \beta \subseteq \{A_1 A_2 \ldots A_k\}$ zachodzi zależność funkcyjna $\alpha \to \beta$, jeżeli dla każdego stanu $r$ relacji $R$ zachodzi:

$$(\forall t_1, t_2 \in r)((t_1.\alpha = t_2.\alpha) \; \Rightarrow \; (t_1.\beta = t_2.\beta))$$

# Definicja i przykład

## Definition (Zależność funkcyjna)

Dla relacji $R = A_1 A_2 \ldots A_k$ oraz zbiorów jej atrybutów $\alpha, \beta \subseteq \{A_1 A_2 \ldots A_k\}$ zachodzi zależność funkcyjna $\alpha \rightarrow \beta$, jeżeli dla każdego stanu $r$ relacji $R$ zachodzi:

$$(\forall t_1, t_2 \in r)((t_1.\alpha = t_2.\alpha) \;\Rightarrow\; (t_1.\beta = t_2.\beta))$$

W relacjach PR(ido,tytuł,nazwisko,adres), ST(ido,indeks,nazwisko,adres) i GR(idg,idk,idp,limit), TS(idg,termin,sala) zachodzą zależności

# Definicja i przykład

## Definition (Zależność funkcyjna)

Dla relacji $R = A_1 A_2 \ldots A_k$ oraz zbiorów jej atrybutów $\alpha, \beta \subseteq \{A_1 A_2 \ldots A_k\}$ zachodzi zależność funkcyjna $\alpha \rightarrow \beta$, jeżeli dla każdego stanu $r$ relacji $R$ zachodzi:

$$(\forall t_1, t_2 \in r)((t_1.\alpha = t_2.\alpha) \;\Rightarrow\; (t_1.\beta = t_2.\beta))$$

W relacjach PR(ido,tytuł,nazwisko,adres), ST(ido,indeks,nazwisko,adres) i GR(idg,idk,idp,limit), TS(idg,termin,sala) zachodzą zależności

- w PR: *ido* $\rightarrow$ *nazwisko*, *adres*, *tytu*,

# Definicja i przykład

## Definition (Zależność funkcyjna)

Dla relacji $R = A_1 A_2 \ldots A_k$ oraz zbiorów jej atrybutów $\alpha, \beta \subseteq \{A_1 A_2 \ldots A_k\}$ zachodzi zależność funkcyjna $\alpha \to \beta$, jeżeli dla każdego stanu $r$ relacji $R$ zachodzi:

$$(\forall t_1, t_2 \in r)((t_1.\alpha = t_2.\alpha) \; \Rightarrow \; (t_1.\beta = t_2.\beta))$$

W relacjach PR(ido,tytuł,nazwisko,adres), ST(ido,indeks,nazwisko,adres) i GR(idg,idk,idp,limit), TS(idg,termin,sala) zachodzą zależności

- w PR: *ido* $\to$ *nazwisko*, *adres*, *tytu*,
- w ST: *ido* $\to$ *nazwisko*, *adres*, *indeks* oraz *indeks* $\to$ *ido*, *nazwisko*, *adres*;

# Definicja i przykład

## Definition (Zależność funkcyjna)

Dla relacji $R = A_1 A_2 \dots A_k$ oraz zbiorów jej atrybutów $\alpha, \beta \subseteq \{A_1 A_2 \dots A_k\}$ zachodzi zależność funkcyjna $\alpha \to \beta$, jeżeli dla każdego stanu $r$ relacji $R$ zachodzi:

$$(\forall t_1, t_2 \in r)((t_1.\alpha = t_2.\alpha) \ \Rightarrow \ (t_1.\beta = t_2.\beta))$$

W relacjach PR(ido,tytuł,nazwisko,adres), ST(ido,indeks,nazwisko,adres) i GR(idg,idk,idp,limit), TS(idg,termin,sala) zachodzą zależności

- w PR: *ido* $\to$ *nazwisko*, *adres*, *tytu*,
- w ST: *ido* $\to$ *nazwisko*, *adres*, *indeks* oraz *indeks* $\to$ *ido*, *nazwisko*, *adres*;
- w GR: *idg* $\to$ *idk*, *idp*, *limit*

# Definicja i przykład

## Definition (Zależność funkcyjna)

Dla relacji $R = A_1A_2 \ldots A_k$ oraz zbiorów jej atrybutów $\alpha, \beta \subseteq \{A_1A_2 \ldots A_k\}$ zachodzi zależność funkcyjna $\alpha \to \beta$, jeżeli dla każdego stanu $r$ relacji $R$ zachodzi:

$$(\forall t_1, t_2 \in r)((t_1.\alpha = t_2.\alpha) \;\Rightarrow\; (t_1.\beta = t_2.\beta))$$

W relacjach PR(ido,tytuł,nazwisko,adres), ST(ido,indeks,nazwisko,adres) i GR(idg,idk,idp,limit), TS(idg,termin,sala) zachodzą zależności

- w PR: *ido* $\to$ *nazwisko*, *adres*, *tytu*,
- w ST: *ido* $\to$ *nazwisko*, *adres*, *indeks* oraz *indeks* $\to$ *ido*, *nazwisko*, *adres*;
- w GR: *idg* $\to$ *idk*, *idp*, *limit*
- w TS: *termin*, *sala* $\to$ *idg*

# Definicja i przykład

## Definition (Zależność funkcyjna)

Dla relacji $R = A_1 A_2 \ldots A_k$ oraz zbiorów jej atrybutów $\alpha, \beta \subseteq \{A_1 A_2 \ldots A_k\}$ zachodzi zależność funkcyjna $\alpha \to \beta$, jeżeli dla każdego stanu $r$ relacji $R$ zachodzi:

$$(\forall t_1, t_2 \in r)((t_1.\alpha = t_2.\alpha) \;\Rightarrow\; (t_1.\beta = t_2.\beta))$$

W relacjach PR(ido,tytuł,nazwisko,adres), ST(ido,indeks,nazwisko,adres) i GR(idg,idk,idp,limit), TS(idg,termin,sala) zachodzą zależności

- w PR: $ido \to nazwisko, adres, tytu$,
- w ST: $ido \to nazwisko, adres, indeks$ oraz $indeks \to ido, nazwisko, adres$;
- w GR: $idg \to idk, idp, limit$
- w TS: $termin, sala \to idg$

**Spostrzeżenie:** Jeśli $K$ jest kluczem $R$, to $K \to R$.

# Klucz relacji

## Definition (Klucz relacji)

Kluczem relacji $R$ nazywamy taki podzbiór $K$ jej atrybutów, który:

- wyznacza funkcyjnie wszystkie atrybuty $R$, czyli $K \to R$ oraz
- jest minimalnym zbiorem o tej własności, czyli $(\forall L \subsetneq K)\neg(L \to R)$

# Klucz relacji

## Definition (Klucz relacji)

Kluczem relacji $R$ nazywamy taki podzbiór $K$ jej atrybutów, który:

- wyznacza funkcyjnie wszystkie atrybuty $R$, czyli $K \to R$ oraz
- jest minimalnym zbiorem o tej własności, czyli $(\forall L \subsetneq K)\neg(L \to R)$

## Definition

Nadklucz reacji — dowolny zbiór atrybutów zawierający klucz relacji,

Klucz główny — jeden z kluczy relacji,

Klucz alternatywny — klucz relacji inny niż klucz główny,

Atrybut główny — atrybut (dowolnego) klucza relacji.

# Klucz relacji

## Definition (Klucz relacji)

Kluczem relacji $R$ nazywamy taki podzbiór $K$ jej atrybutów, który:

- wyznacza funkcyjnie wszystkie atrybuty $R$, czyli $K \to R$ oraz
- jest minimalnym zbiorem o tej własności, czyli $(\forall L \subsetneq K)\neg(L \to R)$

## Definition

Nadklucz reacji — dowolny zbiór atrybutów zawierający klucz relacji,

Klucz główny — jeden z kluczy relacji,

Klucz alternatywny — klucz relacji inny niż klucz główny,

Atrybut główny — atrybut (dowolnego) klucza relacji.

W relacji ST(ido,indeks,nazwisko,adres)

- Kluczem jest *indeks* i kluczem jest *ido*,
- Nadkluczami są: $\{indeks, nazwisko\}$ lub $\{ido, indeks, adres\}$,
- Jako klucz główny możemy wybrać *indeks*,
- Atrybuty główne to $\{ido, indeks\}$

# Klucz relacji

## Definition (Klucz relacji)

Kluczem relacji $R$ nazywamy taki podzbiór $K$ jej atrybutów, który:

- wyznacza funkcyjnie wszystkie atrybuty $R$, czyli $K \rightarrow R$ oraz
- jest minimalnym zbiorem o tej własności, czyli $(\forall L \subsetneq K) \neg (L \rightarrow R)$

## Definition

Nadklucz reacji — dowolny zbiór atrybutów zawierający klucz relacji,

Klucz główny — jeden z kluczy relacji,

Klucz alternatywny — klucz relacji inny niż klucz główny,

Atrybut główny — atrybut (dowolnego) klucza relacji.

W relacji ST(ido,indeks,nazwisko,adres) w relacji TS(idg,termin,sala)

- Kluczem jest *indeks* i kluczem jest *ido*,
- Nadkluczami są: $\{indeks, nazwisko\}$ lub $\{ido, indeks, adres\}$,
- Jako klucz główny możemy wybrać *indeks*,
- Atrybuty główne to $\{ido, indeks\}$
- W relacji TS kluczem jest $\{termin, sala\}$

# Reguły dla zależności funkcyjnych

## Definition (Aksjomaty Armstronga i in.)

Dla relacji *R* i zbiorów jej atrybutów $\alpha, \beta, \gamma \subseteq R$ zachodzi:

# Reguły dla zależności funkcyjnych

## Definition (Aksjomaty Armstronga i in.)

Dla relacji *R* i zbiorów jej atrybutów $\alpha, \beta, \gamma \subseteq$ *R* zachodzi:

- **Zwrotność** $(\beta \subseteq \alpha) \;\Rightarrow\; (\alpha \to \beta)$ (zależności trywialne)
- **Rozszerzanie** $(\alpha \to \beta) \;\Rightarrow\; (\alpha\gamma \to \beta\gamma)$
- **Przechodniość** $(\alpha \to \beta \wedge \beta \to \gamma) \;\Rightarrow\; (\alpha \to \gamma)$

# Reguły dla zależności funkcyjnych

## Definition (Aksjomaty Armstronga i in.)

Dla relacji *R* i zbiorów jej atrybutów $\alpha, \beta, \gamma \subseteq$ *R* zachodzi:

- **Zwrotność** $(\beta \subseteq \alpha) \ \Rightarrow \ (\alpha \to \beta)$ (zależności trywialne)
- **Rozszerzanie** $(\alpha \to \beta) \ \Rightarrow \ (\alpha\gamma \to \beta\gamma)$
- **Przechodniość** $(\alpha \to \beta \wedge \beta \to \gamma) \ \Rightarrow \ (\alpha \to \gamma)$
- **Sumowanie** $(\alpha \to \beta \wedge \alpha \to \gamma) \ \Rightarrow \ (\alpha \to \beta\gamma)$
- **Rozkładanie** $(\alpha \to \beta\gamma) \ \Rightarrow \ (\alpha \to \beta \wedge \alpha \to \gamma)$

# Reguły dla zależności funkcyjnych

## Definition (Aksjomaty Armstronga i in.)

Dla relacji *R* i zbiorów jej atrybutów $\alpha, \beta, \gamma \subseteq R$ zachodzi:

- Zwrotność $(\beta \subseteq \alpha) \Rightarrow (\alpha \to \beta)$ (zależności trywialne)
- Rozszerzanie $(\alpha \to \beta) \Rightarrow (\alpha\gamma \to \beta\gamma)$
- Przechodniość $(\alpha \to \beta \wedge \beta \to \gamma) \Rightarrow (\alpha \to \gamma)$
- Sumowanie $(\alpha \to \beta \wedge \alpha \to \gamma) \Rightarrow (\alpha \to \beta\gamma)$
- Rozkładanie $(\alpha \to \beta\gamma) \Rightarrow (\alpha \to \beta \wedge \alpha \to \gamma)$

## Definition (Domknięcie zbioru zależności i zbioru atrybutów)

Dla relacji *R* i jej zbioru zależności funkcyjnych *F*:

- $F^+$ — **domknięciem zbioru zależności** *F* nazywamy zbiór wszystkich zależności wyprowadzalnych z *F*.
- $(\alpha)^+_F$ — **domknięciem zbioru atrybutów** $\alpha \subseteq R$ **względem** *F* nazywamy zbiór atrybutów, które można wyprowadzić z $\alpha$ za pomocą *F*.

# Twierdzenie o znaczeniu Aksjomatów Armstronga

## Definition (Aksjomaty Armstronga)

Dla relacji *R* i zbiorów jej atrybutów $\alpha, \beta, \gamma \subseteq R$ zachodzi:

- **Zwrotność** $(\beta \subseteq \alpha) \Rightarrow (\alpha \to \beta)$ (zależności trywialne)
- **Rozszerzanie** $(\alpha \to \beta) \Rightarrow (\alpha\gamma \to \beta\gamma)$
- **Przechodniość** $(\alpha \to \beta \wedge \beta \to \gamma) \Rightarrow (\alpha \to \gamma)$

# Twierdzenie o znaczeniu Aksjomatów Armstronga

## Definition (Aksjomaty Armstronga)

Dla relacji *R* i zbiorów jej atrybutów $\alpha, \beta, \gamma \subseteq$ *R* zachodzi:

- Zwrotność $(\beta \subseteq \alpha) \Rightarrow (\alpha \rightarrow \beta)$ (zależności trywialne)
- Rozszerzanie $(\alpha \rightarrow \beta) \Rightarrow (\alpha\gamma \rightarrow \beta\gamma)$
- Przechodniość $(\alpha \rightarrow \beta \wedge \beta \rightarrow \gamma) \Rightarrow (\alpha \rightarrow \gamma)$

## Theorem

*Aksjomaty Armstronga stanowią zupełny, niesprzeczny i minimalny zbiór reguł pozwalający wyprowadzić ze zbioru zależności F każdą zależność funkcyjną prawdziwą w każdym stanie relacji, w którym spełnione są reguły F.*

# Twierdzenie o znaczeniu Aksjomatów Armstronga

## Definition (Aksjomaty Armstronga)

Dla relacji *R* i zbiorów jej atrybutów $\alpha, \beta, \gamma \subseteq R$ zachodzi:

Zwrotność $(\beta \subseteq \alpha) \;\Rightarrow\; (\alpha \to \beta)$ (zależności trywialne)

Rozszerzanie $(\alpha \to \beta) \;\Rightarrow\; (\alpha\gamma \to \beta\gamma)$

Przechodniość $(\alpha \to \beta \wedge \beta \to \gamma) \;\Rightarrow\; (\alpha \to \gamma)$

## Theorem

*Aksjomaty Armstronga stanowią zupełny, niesprzeczny i minimalny zbiór reguł pozwalający wyprowadzić ze zbioru zależności F każdą zależność funkcyjną prawdziwą w każdym stanie relacji, w którym spełnione są reguły F.*

## Wniosek

Tym samym zależności dowodliwe za pomocą aksjomatów Armstronga to zależności prawdziwe.

# Uwagi ad. domknięcia zbiorów zależności i atrybutów

# Uwagi ad. domknięcia zbiorów zależności i atrybutów

1. Zazwyczaj nie wyznaczamy $F^+$ — to zbiór duży i zawierający dużo nieciekawych informacji (np. zależności trywialne).

# Uwagi ad. domknięcia zbiorów zależności i atrybutów

1. Zazwyczaj nie wyznaczamy $F^+$ — to zbiór duży i zawierający dużo nieciekawych informacji (np. zależności trywialne).
2. Efektywne wyznaczanie $(\alpha)^+_F$ jest potrzebne — pozwala zadecydować, co jest kluczem, czy w relacji jest redundancja itp.

# Uwagi ad. domknięcia zbiorów zależności i atrybutów

1. Zazwyczaj nie wyznaczamy $F^+$ — to zbiór duży i zawierający dużo nieciekawych informacji (np. zależności trywialne).
2. Efektywne wyznaczanie $(\alpha)_F^+$ jest potrzebne — pozwala zadecydować, co jest kluczem, czy w relacji jest redundancja itp.
3. Mamy algorytm, który pozwala wyznaczać $\chi = (\alpha)_F^+$:
    - $\chi \leftarrow \alpha$ (zwrotność)
    - dopóki $\chi$ zmienia się:
        - znajdź $\beta \in \chi$ taki, że istnieje $\beta \to \gamma \in F$ oraz $\gamma \setminus \chi \neq \emptyset$
        - $\chi \leftarrow \chi \cup \gamma$ (zastosuj rozszerzanie i przechodniość)
    - zwróć $\chi$ jako wynik

# Uwagi ad. domknięcia zbiorów zależności i atrybutów

1. Zazwyczaj nie wyznaczamy $F^+$ — to zbiór duży i zawierający dużo nieciekawych informacji (np. zależności trywialne).
2. Efektywne wyznaczanie $(\alpha)_F^+$ jest potrzebne — pozwala zadecydować, co jest kluczem, czy w relacji jest redundancja itp.
3. Mamy algorytm, który pozwala wyznaczać $\chi = (\alpha)_F^+$:
   - $\chi \leftarrow \alpha$ (zwrotność)
   - dopóki $\chi$ zmienia się:
     - znajdź $\beta \in \chi$ taki, że istnieje $\beta \rightarrow \gamma \in F$ oraz $\gamma \setminus \chi \neq \emptyset$
     - $\chi \leftarrow \chi \cup \gamma$ (zastosuj rozszerzanie i przechodniość)
   - zwróć $\chi$ jako wynik
4. Mamy sposób, by porównywać zbiory zależności $F$ i $G$ dla tej samej relacji. Sprawdzamy, czy $F^+ = G^+$:
   - dla każdej zależności $\alpha \rightarrow \beta \in F$:
     - oblicz $\chi = (\alpha)_G^+$
     - jeśli $\beta \subseteq \chi$, to $\alpha \rightarrow \beta \in G^+$, w przeciwnym wypadku $F^+ \neq G^+$
   - powtórz to dla każdej zależności $\alpha \rightarrow \beta \in G$ i zbioru $F$

# Uwagi ad. domknięcia zbiorów zależności i atrybutów

1. Zazwyczaj nie wyznaczamy $F^+$ — to zbiór duży i zawierający dużo nieciekawych informacji (np. zależności trywialne).
2. Efektywne wyznaczanie $(\alpha)^+_F$ jest potrzebne — pozwala zadecydować, co jest kluczem, czy w relacji jest redundancja itp.
3. Mamy algorytm, który pozwala wyznaczać $\chi = (\alpha)^+_F$:
   - $\chi \leftarrow \alpha$ (zwrotność)
   - dopóki $\chi$ zmienia się:
     - znajdź $\beta \in \chi$ taki, że istnieje $\beta \rightarrow \gamma \in F$ oraz $\gamma \setminus \chi \neq \emptyset$
     - $\chi \leftarrow \chi \cup \gamma$ (zastosuj rozszerzanie i przechodniość)
   - zwróć $\chi$ jako wynik
4. Mamy sposób, by porównywać zbiory zależności $F$ i $G$ dla tej samej relacji. Sprawdzamy, czy $F^+ = G^+$:
   - dla każdej zależności $\alpha \rightarrow \beta \in F$:
     - oblicz $\chi = (\alpha)^+_G$
     - jeśli $\beta \subseteq \chi$, to $\alpha \rightarrow \beta \in G^+$, w przeciwnym wypadku $F^+ \neq G^+$
   - powtórz to dla każdej zależności $\alpha \rightarrow \beta \in G$ i zbioru $F$
5. Zależności funkcyjne powinny być kontrolowane przez SZBD. Dlatego dobrze, by było ich mało. Zbiór $F_{min}$ nazwiemy minimalnym pokryciem $F$ jeśli jest równoważny $F$ i nie zawiera zależności "nadmiarowych".

# Postać normalna Boyce-Codda

## Definition (Postać normalna Boyce-Codda, BCNF)

Relacja *R* ze zbiorem zależności funkcyjnych *F* jest w postaci normalnej Boyce-Codda, jeśli dla każdej nietrywialnej zależności $\alpha \rightarrow \beta$ ($\alpha \cap \beta = \emptyset$) zbiór $\alpha$ jest nadkluczem.

# Postać normalna Boyce-Codda

## Definition (Postać normalna Boyce-Codda, BCNF)

Relacja *R* ze zbiorem zależności funkcyjnych *F* jest w postaci normalnej Boyce-Codda, jeśli dla każdej nietrywialnej zależności $\alpha \to \beta$ ($\alpha \cap \beta = \emptyset$) zbiór $\alpha$ jest nadkluczem.

**Uwagi**:

1. Relacja w BCNF ma tylko zależności trywialne i wynikające z nadklucza.

# Postać normalna Boyce-Codda

## Definition (Postać normalna Boyce-Codda, BCNF)

Relacja *R* ze zbiorem zależności funkcyjnych *F* jest w postaci normalnej Boyce-Codda, jeśli dla każdej nietrywialnej zależności $\alpha \rightarrow \beta$ ($\alpha \cap \beta = \emptyset$) zbiór $\alpha$ jest nadkluczem.

**Uwagi**:

1. Relacja w BCNF ma tylko zależności trywialne i wynikające z nadklucza.
2. Kontrola zależności funkcyjnych w relacji w BCNF sprowadza się do kontroli własności klucza.

# Postać normalna Boyce-Codda

## Definition (Postać normalna Boyce-Codda, BCNF)

Relacja $R$ ze zbiorem zależności funkcyjnych $F$ jest w postaci normalnej Boyce-Codda, jeśli dla każdej nietrywialnej zależności $\alpha \to \beta$ ($\alpha \cap \beta = \emptyset$) zbiór $\alpha$ jest nadkluczem.

**Uwagi**:

1. Relacja w BCNF ma tylko zależności trywialne i wynikające z nadklucza.
2. Kontrola zależności funkcyjnych w relacji w BCNF sprowadza się do kontroli własności klucza.

**Przykłady**:

1. $R = ABCDE$ oraz $F = \{AB \to C, B \to D, C \to E, E \to C\}$ nie jest w BCNF. Kluczem jest $AB$. Tylko $AB \to C$ nie narusza BCNF. Pozostałe:
   - **zależność częściowa:** $B \to D$ wynika z podzbioru klucza,
   - **zależności przechodnie:** $C \to E$ i $E \to C$, których lewa strona nie należy do klucza.

# Postać normalna Boyce-Codda

## Definition (Postać normalna Boyce-Codda, BCNF)

Relacja $R$ ze zbiorem zależności funkcyjnych $F$ jest w postaci normalnej Boyce-Codda, jeśli dla każdej nietrywialnej zależności $\alpha \to \beta$ ($\alpha \cap \beta = \emptyset$) zbiór $\alpha$ jest nadkluczem.

**Uwagi**:

1. Relacja w BCNF ma tylko zależności trywialne i wynikające z nadklucza.
2. Kontrola zależności funkcyjnych w relacji w BCNF sprowadza się do kontroli własności klucza.

**Przykłady**:

1. $R = ABCDE$ oraz $F = \{AB \to C, B \to D, C \to E, E \to C\}$ nie jest w BCNF. Kluczem jest $AB$. Tylko $AB \to C$ nie narusza BCNF. Pozostałe:
   - **zależność częściowa:** $B \to D$ wynika z podzbioru klucza,
   - **zależności przechodnie:** $C \to E$ i $E \to C$, których lewa strona nie należy do klucza.
2. $R = ABC$ oraz $F = \{AB \to C\}$ jest w BCNF

# Rozkład relacji

# Rozkład relacji

- **Rozkładem relacji $R$** nazywamy zbiór relacji $\{R_1, \ldots, R_k\}$ taki, że $R = R_1 \cup \ldots \cup R_k$.

# Rozkład relacji

- Rozkładem relacji $R$ nazywamy zbiór relacji $\{R_1, \ldots, R_k\}$ taki, że $R = R_1 \cup \ldots \cup R_k$.
- Dla $F$ — zbioru zależności $R$, rzutem $F$ na $R_i$ jest $F_i = \{\alpha \to \beta \in F^+ \mid \alpha, \beta \in R_i\}$.

# Rozkład relacji

- Rozkładem relacji $R$ nazywamy zbiór relacji $\{R_1, \ldots, R_k\}$ taki, że $R = R_1 \cup \ldots \cup R_k$.
- Dla $F$ — zbioru zależności $R$, rzutem $F$ na $R_i$ jest $F_i = \{\alpha \to \beta \in F^+ \mid \alpha, \beta \in R_i\}$.
- Dla $r$ — stanu relacji $R$, stanem $R_i$ jest $r_i = \pi_{R_i}(r)$.

# Rozkład relacji

- Rozkładem relacji $R$ nazywamy zbiór relacji $\{R_1, \ldots, R_k\}$ taki, że $R = R_1 \cup \ldots \cup R_k$.
- Dla $F$ — zbioru zależności $R$, rzutem $F$ na $R_i$ jest $F_i = \{\alpha \to \beta \in F^+ \mid \alpha, \beta \in R_i\}$.
- Dla $r$ — stanu relacji $R$, stanem $R_i$ jest $r_i = \pi_{R_i}(r)$.
- Złączenie naturalne jest operacją przeciwną do rokładu.

# Rozkład relacji

- Rozkładem relacji $R$ nazywamy zbiór relacji $\{R_1, \ldots, R_k\}$ taki, że $R = R_1 \cup \ldots \cup R_k$.
- Dla $F$ — zbioru zależności $R$, rzutem $F$ na $R_i$ jest $F_i = \{\alpha \to \beta \in F^+ \mid \alpha, \beta \in R_i\}$.
- Dla $r$ — stanu relacji $R$, stanem $R_i$ jest $r_i = \pi_{R_i}(r)$.
- Złączenie naturalne jest operacją przeciwną do rokładu.
- Rozkład $R$ na $R_1, \ldots, R_k$ jest odwracalny, jeśli dla każdego poprawnego stanu $r$ (spełniającego zależności $F$) zachodzi:

$$r \;=\; r_1 \bowtie r_2 \bowtie \cdots \bowtie r_k$$

# Rozkład relacji

- Rozkładem relacji $R$ nazywamy zbiór relacji $\{R_1, \ldots, R_k\}$ taki, że $R = R_1 \cup \ldots \cup R_k$.
- Dla $F$ — zbioru zależności $R$, rzutem $F$ na $R_i$ jest $F_i = \{\alpha \to \beta \in F^+ \mid \alpha, \beta \in R_i\}$.
- Dla $r$ — stanu relacji $R$, stanem $R_i$ jest $r_i = \pi_{R_i}(r)$.
- Złączenie naturalne jest operacją przeciwną do rokładu.
- Rozkład $R$ na $R_1, \ldots, R_k$ jest odwracalny, jeśli dla każdego poprawnego stanu $r$ (spełniającego zależności $F$) zachodzi:

$$r = r_1 \bowtie r_2 \bowtie \cdots \bowtie r_k$$

- Rozkład $R$ na $R_1, \ldots, R_k$ zachowuje zależności, jeśli:

$$F^+ = (F_1 \cup F_2 \cup \ldots \cup F_k)^+$$

# Rozkład relacji

- Rozkładem relacji $R$ nazywamy zbiór relacji $\{R_1, \ldots, R_k\}$ taki, że $R = R_1 \cup \ldots \cup R_k$.
- Dla $F$ — zbioru zależności $R$, rzutem $F$ na $R_i$ jest $F_i = \{\alpha \to \beta \in F^+ \mid \alpha, \beta \in R_i\}$.
- Dla $r$ — stanu relacji $R$, stanem $R_i$ jest $r_i = \pi_{R_i}(r)$.
- Złączenie naturalne jest operacją przeciwną do rokładu.
- Rozkład $R$ na $R_1, \ldots, R_k$ jest odwracalny, jeśli dla każdego poprawnego stanu $r$ (spełniającego zależności $F$) zachodzi:

$$r \; = \; r_1 \bowtie r_2 \bowtie \cdots \bowtie r_k$$

- Rozkład $R$ na $R_1, \ldots, R_k$ zachowuje zależności, jeśli:

$$F^+ \; = \; (F_1 \cup F_2 \cup \ldots \cup F_k)^+$$

- Rozkład relacji na składowe MUSI być odwracalny i POWINIEN zachowywać zależności.

# Rozkład relacji do BCNF

# Rozkład relacji do BCNF

## Lemma

*Niech R będzie relacją i F jej zbiorem zależności funkcyjnych. Jeżeli* $\alpha \to \beta \in F^+$ *jest nietrywialna* ($\alpha \cap \beta = \emptyset$)*, to rozkład R na* $R_1 = \alpha\beta$ *i* $R_2 = R \setminus \beta$ *jest odwracalny.*

# Rozkład relacji do BCNF

## Lemma

*Niech R będzie relacją i F jej zbiorem zależności funkcyjnych. Jeżeli $\alpha \to \beta \in F^+$ jest nietrywialna ($\alpha \cap \beta = \emptyset$), to rozkład R na $R_1 = \alpha\beta$ i $R_2 = R \setminus \beta$ jest odwracalny.*

## Lemma

*Każda relacja ma **odwracalny** rozkład na składowe w BCNF.*

# Rozkład relacji do BCNF

## Lemma

*Niech R będzie relacją i F jej zbiorem zależności funkcyjnych. Jeżeli $\alpha \to \beta \in F^+$ jest nietrywialna ($\alpha \cap \beta = \emptyset$), to rozkład R na $R_1 = \alpha\beta$ i $R_2 = R \setminus \beta$ jest odwracalny.*

## Lemma

*Każda relacja ma **odwracalny** rozkład na składowe w BCNF.*

## Lemma

*Istnieją relacje, które nie mają **odwracalnego i zachowującego zależności** rozkładu na składowe w BCNF.*

# Przykład rozkładu do BCNF

## Dane:

- $R = KNGSUO$
- $F = \{K \to N, KU \to O, GS \to K, GU \to S, NG \to S\}$
- klucz $R$ to $GU$

## Rozkład:

.

# Przykład rozkładu do BCNF

## Dane:

- $R = KNGSUO$
- $F = \{K \to N, KU \to O, GS \to K, GU \to S, NG \to S\}$
- klucz $R$ to $GU$

## Rozkład:

$R$: $R$ nie jest w BCNF. Rozkładamy wg $KU \to O$ na $R_1 = KUO$ i $R_2 = KNGSU$;

.

# Przykład rozkładu do BCNF

## Dane:

- $R = KNGSUO$
- $F = \{K \rightarrow N, KU \rightarrow O, GS \rightarrow K, GU \rightarrow S, NG \rightarrow S\}$
- klucz $R$ to $GU$

## Rozkład:

$R$: $R$ nie jest w BCNF. Rozkładamy wg $KU \rightarrow O$ na $R_1 = KUO$ i $R_2 = KNGSU$;

$R_1$ : $R_1 = \underline{KU}O$, $F_1 = \{KU \rightarrow O\}$

$R_2$: $R_2 = KN\underline{G}S\underline{U}$, $F_2 = \{\underline{K \rightarrow N}, NG \rightarrow S, GS \rightarrow K, GU \rightarrow S\}$

.

# Przykład rozkładu do BCNF

## Dane:

- $R = KNGSUO$
- $F = \{K \to N, KU \to O, GS \to K, GU \to S, NG \to S\}$
- klucz $R$ to $GU$

## Rozkład:

$R$: $R$ nie jest w BCNF. Rozkładamy wg $KU \to O$ na $R_1 = KUO$ i $R_2 = KNGSU$;

$R_1$ : $R_1 = \underline{KU}O$, $F_1 = \{KU \to O\}$ (BCNF)

$R_2$: $R_2 = KN\underline{G}S\underline{U}$, $F_2 = \{\underline{K \to N}, NG \to S, GS \to K, GU \to S\}$ (Nie-BCNF)

.

# Przykład rozkładu do BCNF

## Dane:

- $R = KNGSUO$
- $F = \{K \to N, KU \to O, GS \to K, GU \to S, NG \to S\}$
- klucz $R$ to $GU$

## Rozkład:

$R$: $R$ nie jest w BCNF. Rozkładamy wg $KU \to O$ na $R_1 = KUO$ i $R_2 = KNGSU$;

$R_1$ : $R_1 = \underline{KU}O$, $F_1 = \{KU \to O\}$ (BCNF)

$R_2$: $R_2 = KN\underline{G}S\underline{U}$, $F_2 = \{\underline{K \to N}, NG \to S, GS \to K, GU \to S\}$ (Nie-BCNF)

$R_{21}$: $R_{21} = K\underline{G}S\underline{U}$, $F_{21} = \{\underline{KG \to S}, GS \to K, GU \to S\}$.

$R_{22}$: $R_{22} = \underline{K}N$, $F_{22} = \{K \to N\}$ .

# Przykład rozkładu do BCNF

## Dane:

- $R = KNGSUO$
- $F = \{K \to N, KU \to O, GS \to K, GU \to S, NG \to S\}$
- klucz $R$ to $GU$

## Rozkład:

$R$: $R$ nie jest w BCNF. Rozkładamy wg $KU \to O$ na $R_1 = KUO$ i $R_2 = KNGSU$;

$R_1$ : $R_1 = \underline{KU}O$, $F_1 = \{KU \to O\}$ (BCNF)

$R_2$: $R_2 = KN\underline{G}S\underline{U}$, $F_2 = \{\underline{K \to N}, NG \to S, GS \to K, GU \to S\}$ (Nie-BCNF)

$R_{21}$: $R_{21} = K\underline{G}S\underline{U}$, $F_{21} = \{\underline{KG \to S}, GS \to K, GU \to S\}$. (Nie-BCNF)

$R_{22}$: $R_{22} = \underline{K}N$, $F_{22} = \{K \to N\}$ (BCNF).

# Przykład rozkładu do BCNF

## Dane:

- $R = KNGSUO$
- $F = \{K \to N, KU \to O, GS \to K, GU \to S, NG \to S\}$
- klucz $R$ to $GU$

## Rozkład:

$R$: $R$ nie jest w BCNF. Rozkładamy wg $KU \to O$ na $R_1 = KUO$ i $R_2 = KNGSU$;

$R_1$ : $R_1 = \underline{KU}O$, $F_1 = \{KU \to O\}$ (BCNF)

$R_2$: $R_2 = KN\underline{G}S\underline{U}$, $F_2 = \{\underline{K \to N}, NG \to S, GS \to K, GU \to S\}$ (Nie-BCNF)

$R_{21}$: $R_{21} = K\underline{G}S\underline{U}$, $F_{21} = \{\underline{KG \to S}, GS \to K, GU \to S\}$. (Nie-BCNF)

$R_{211}$: $R_{211} = KGS$, $F_{211} = \{KG \to S, GS \to K\}$, klucze: $KG$ i $GS$.

$R_{212}$: $R_{212} = K\underline{GU}$, $F_{212} = \{GU \to K\}$.

$R_{22}$: $R_{22} = \underline{K}N$, $F_{22} = \{K \to N\}$ (BCNF).

# Przykład rozkładu do BCNF

## Dane:

- $R = KNGSUO$
- $F = \{K \to N, KU \to O, GS \to K, GU \to S, NG \to S\}$
- klucz $R$ to $GU$

## Rozkład:

$R$: $R$ nie jest w BCNF. Rozkładamy wg $KU \to O$ na $R_1 = KUO$ i $R_2 = KNGSU$;

$R_1$ : $R_1 = \underline{KU}O$, $F_1 = \{KU \to O\}$ (BCNF)

$R_2$: $R_2 = KN\underline{G}S\underline{U}$, $F_2 = \{\underline{K \to N}, NG \to S, GS \to K, GU \to S\}$ (Nie-BCNF)

$R_{21}$: $R_{21} = K\underline{G}S\underline{U}$, $F_{21} = \{\underline{KG \to S}, GS \to K, GU \to S\}$. (Nie-BCNF)

$R_{211}$: $R_{211} = KGS$, $F_{211} = \{KG \to S, GS \to K\}$, klucze: $KG$ i $GS$. (BCNF)

$R_{212}$: $R_{212} = K\underline{GU}$, $F_{212} = \{GU \to K\}$. (BCNF)

$R_{22}$: $R_{22} = \underline{K}N$, $F_{22} = \{K \to N\}$ (BCNF).

# Przykład rozkładu do BCNF

## Dane:

- $R = KNGSUO$
- $F = \{K \rightarrow N, KU \rightarrow O, GS \rightarrow K, GU \rightarrow S, NG \rightarrow S\}$
- klucz $R$ to $GU$

## Rozkład:

$R$: $R$ nie jest w BCNF. Rozkładamy wg $KU \rightarrow O$ na $R_1 = KUO$ i $R_2 = KNGSU$;

$R_1$ : $R_1 = \underline{KU}O$, $F_1 = \{KU \rightarrow O\}$ (BCNF)

$R_2$: $R_2 = KN\underline{G}S\underline{U}$, $F_2 = \{\underline{K \rightarrow N}, NG \rightarrow S, GS \rightarrow K, GU \rightarrow S\}$ (Nie-BCNF)

$R_{21}$: $R_{21} = K\underline{G}S\underline{U}$, $F_{21} = \{\underline{KG \rightarrow S}, GS \rightarrow K, GU \rightarrow S\}$. (Nie-BCNF)

$R_{211}$: $R_{211} = KGS$, $F_{211} = \{KG \rightarrow S, GS \rightarrow K\}$, klucze: $KG$ i $GS$. (BCNF)

$R_{212}$: $R_{212} = K\underline{GU}$, $F_{212} = \{GU \rightarrow K\}$. (BCNF)

$R_{22}$: $R_{22} = \underline{K}N$, $F_{22} = \{K \rightarrow N\}$ (BCNF).

**Wynik rozkładu:** $R = KUO \cup KGS \cup KGU \cup KN$ i
$\{KU \rightarrow O, KG \rightarrow S, GS \rightarrow K, GU \rightarrow K, K \rightarrow N\}$

# Trzecia postać normalna

## Definition (Trzecia postać normalna, 3NF)

Relacja *R* z zależnościami funkcyjnymi *F* jest w trzeciej postaci normalnej, jeśli każda zależność $\alpha \rightarrow B \in F$

# Trzecia postać normalna

## Definition (Trzecia postać normalna, 3NF)

Relacja $R$ z zależnościami funkcyjnymi $F$ jest w trzeciej postaci normalnej, jeśli każda zależność $\alpha \to B \in F$

- jest trywialna ($B \in \alpha$) albo

# Trzecia postać normalna

## Definition (Trzecia postać normalna, 3NF)

Relacja $R$ z zależnościami funkcyjnymi $F$ jest w trzeciej postaci normalnej, jeśli każda zależność $\alpha \to B \in F$

- jest trywialna ($B \in \alpha$) albo
- wynika z nadklucza ($(\alpha)_F^+ = R$) albo

# Trzecia postać normalna

## Definition (Trzecia postać normalna, 3NF)

Relacja $R$ z zależnościami funkcyjnymi $F$ jest w trzeciej postaci normalnej, jeśli każda zależność $\alpha \to B \in F$

- jest trywialna ($B \in \alpha$) albo
- wynika z nadklucza ($(\alpha)_F^+ = R$) albo
- ma po prawej stronie atrybut główny ($B$ należy do jakiegoś klucza).

# Trzecia postać normalna

## Definition (Trzecia postać normalna, 3NF)

Relacja *R* z zależnościami funkcyjnymi *F* jest w trzeciej postaci normalnej, jeśli każda zależność $\alpha \to B \in F$

- jest trywialna ($B \in \alpha$) albo
- wynika z nadklucza ($(\alpha)^+_F = R$) albo
- ma po prawej stronie atrybut główny (*B* należy do jakiegoś klucza).

## Lemma

*Każda relacja ma odwracalny i zachowujący zależności rozkład na składowe w postaci 3NF.*

# Algorytm rozkładu do 3NF

# Algorytm rozkładu do 3NF

## Definition ($F_{min}$)

Minimalnym pokryciem zbioru zależności funkcyjnych $F$ nazwiemy równoważny $F$ zbiór $F_{min}$, w którym:

# Algorytm rozkładu do 3NF

## Definition ($F_{min}$)

Minimalnym pokryciem zbioru zależności funkcyjnych $F$ nazwiemy równoważny $F$ zbiór $F_{min}$, w którym:

- nie ma zależności trywialnych, np. $AB \to AC$

# Algorytm rozkładu do 3NF

## Definition ($F_{min}$)

Minimalnym pokryciem zbioru zależności funkcyjnych $F$ nazwiemy równoważny $F$ zbiór $F_{min}$, w którym:

- nie ma zależności trywialnych, np. $AB \to C$

# Algorytm rozkładu do 3NF

## Definition ($F_{min}$)

Minimalnym pokryciem zbioru zależności funkcyjnych $F$ nazwiemy równoważny $F$ zbiór $F_{min}$, w którym:

- nie ma zależności trywialnych, np. $AB \rightarrow C$
- nie ma zależności nadmiarowych, czyli wynikających z pozostałych zależności $F_{min}$, np. $A \rightarrow B, B \rightarrow C, A \rightarrow C$

# Algorytm rozkładu do 3NF

## Definition ($F_{min}$)

Minimalnym pokryciem zbioru zależności funkcyjnych $F$ nazwiemy równoważny $F$ zbiór $F_{min}$, w którym:

- nie ma zależności trywialnych, np. $AB \to C$
- nie ma zależności nadmiarowych, czyli wynikających z pozostałych zależności $F_{min}$, np. $A \to B, B \to C$,

# Algorytm rozkładu do 3NF

## Definition ($F_{min}$)

Minimalnym pokryciem zbioru zależności funkcyjnych $F$ nazwiemy równoważny $F$ zbiór $F_{min}$, w którym:

- nie ma zależności trywialnych, np. $AB \rightarrow C$
- nie ma zależności nadmiarowych, czyli wynikających z pozostałych zależności $F_{min}$, np. $A \rightarrow B, B \rightarrow C$,
- nie ma atrybutów lewostronnie nadmiarowych $AB \rightarrow C, A \rightarrow B$.

# Algorytm rozkładu do 3NF

## Definition ($F_{min}$)

Minimalnym pokryciem zbioru zależności funkcyjnych $F$ nazwiemy równoważny $F$ zbiór $F_{min}$, w którym:

- nie ma zależności trywialnych, np. $AB \to C$
- nie ma zależności nadmiarowych, czyli wynikających z pozostałych zależności $F_{min}$, np. $A \to B, B \to C,$
- nie ma atrybutów lewostronnie nadmiarowych $A \to C, A \to B$.

# Algorytm rozkładu do 3NF

## Definition ($F_{min}$)

Minimalnym pokryciem zbioru zależności funkcyjnych $F$ nazwiemy równoważny $F$ zbiór $F_{min}$, w którym:

- nie ma zależności trywialnych, np. $AB \rightarrow C$
- nie ma zależności nadmiarowych, czyli wynikających z pozostałych zależności $F_{min}$, np. $A \rightarrow B, B \rightarrow C,$
- nie ma atrybutów lewostronnie nadmiarowych $A \rightarrow C, A \rightarrow B$.

## Algorytm rozkładu do 3NF

# Algorytm rozkładu do 3NF

## Definition ($F_{min}$)

Minimalnym pokryciem zbioru zależności funkcyjnych $F$ nazwiemy równoważny $F$ zbiór $F_{min}$, w którym:

- nie ma zależności trywialnych, np. $AB \to C$
- nie ma zależności nadmiarowych, czyli wynikających z pozostałych zależności $F_{min}$, np. $A \to B, B \to C$,
- nie ma atrybutów lewostronnie nadmiarowych $A \to C, A \to B$.

## Algorytm rozkładu do 3NF

1. Wyznacz $F_{min}$.

# Algorytm rozkładu do 3NF

## Definition ($F_{min}$)

Minimalnym pokryciem zbioru zależności funkcyjnych $F$ nazwiemy równoważny $F$ zbiór $F_{min}$, w którym:

- nie ma zależności trywialnych, np. $AB \to C$
- nie ma zależności nadmiarowych, czyli wynikających z pozostałych zależności $F_{min}$, np. $A \to B, B \to C$,
- nie ma atrybutów lewostronnie nadmiarowych $A \to C, A \to B$.

## Algorytm rozkładu do 3NF

1. Wyznacz $F_{min}$.
2. Dla każdej zależności $\alpha \to \beta \in F_{min}$ utwórz składową $R_i = \alpha\beta$. Usuń składowe zawierające się w innych.

# Algorytm rozkładu do 3NF

## Definition ($F_{min}$)

Minimalnym pokryciem zbioru zależności funkcyjnych $F$ nazwiemy równoważny $F$ zbiór $F_{min}$, w którym:

- nie ma zależności trywialnych, np. $AB \to C$
- nie ma zależności nadmiarowych, czyli wynikających z pozostałych zależności $F_{min}$, np. $A \to B, B \to C,$
- nie ma atrybutów lewostronnie nadmiarowych $A \to C, A \to B$.

## Algorytm rozkładu do 3NF

1. Wyznacz $F_{min}$.
2. Dla każdej zależności $\alpha \to \beta \in F_{min}$ utwórz składową $R_i = \alpha\beta$. Usuń składowe zawierające się w innych.
3. Jeśli żadna z utworzonych składowych nie zawiera klucza $R$, to dodaj do rozkładu składową $K$ dla pewnego klucza $K$ relacji $R$.

# Postaci normalne 1-5 i in.

# Postaci normalne 1-5 i in.

1NF — relacja jest w pierwszej postaci normalnej, jeśli ma atrybuty bez wewnątrznej struktury i bez powtórzeń.

# Postaci normalne 1-5 i in.

1NF — relacja jest w pierwszej postaci normalnej, jeśli ma atrybuty bez wewnątrznej struktury i bez powtórzeń.

2NF — relacja jest w drugiej postaci normalnej, jeśli jest w 1NF i nie ma zależności częściowych (unikamy redundancji).

# Postaci normalne 1-5 i in.

1NF — relacja jest w pierwszej postaci normalnej, jeśli ma atrybuty bez wewnątrznej struktury i bez powtórzeń.

2NF — relacja jest w drugiej postaci normalnej, jeśli jest w 1NF i nie ma zależności częściowych (unikamy redundancji).

3NF — relacja jest w trzeciej postaci normalnej, jeśli jest w 2NF i nie ma zależności przechodnich (unikamy redundancji).

# Postaci normalne 1-5 i in.

- 1NF — relacja jest w pierwszej postaci normalnej, jeśli ma atrybuty bez wewnątrznej struktury i bez powtórzeń.
- 2NF — relacja jest w drugiej postaci normalnej, jeśli jest w 1NF i nie ma zależności częściowych (unikamy redundancji).
- 3NF — relacja jest w trzeciej postaci normalnej, jeśli jest w 2NF i nie ma zależności przechodnich (unikamy redundancji).
- 3.5NF — BCNF (unikamy redundancji i upraszaczmy sprawdzanie zależności).

# Postaci normalne 1-5 i in.

- 1NF — relacja jest w pierwszej postaci normalnej, jeśli ma atrybuty bez wewnątrznej struktury i bez powtórzeń.
- 2NF — relacja jest w drugiej postaci normalnej, jeśli jest w 1NF i nie ma zależności częściowych (unikamy redundancji).
- 3NF — relacja jest w trzeciej postaci normalnej, jeśli jest w 2NF i nie ma zależności przechodnich (unikamy redundancji).
- 3.5NF — BCNF (unikamy redundancji i upraszaczmy sprawdzanie zależności).
- 4NF — GR(idg,idk,idp,limit), TS(idg,termin,sala), Z(ids,idg), PLAN(ids,idg,termin,sala) — w planie studenta muszą znaleźć się wszystkie terminy grupy, do której student się zapisał; relacja jest w czwartej postaci normalnej, jeśli jest w BCNF i nie ma nietrywialnych **zależności wielowartościowych**.

# Postaci normalne 1-5 i in.

1NF — relacja jest w pierwszej postaci normalnej, jeśli ma atrybuty bez wewnątrznej struktury i bez powtórzeń.

2NF — relacja jest w drugiej postaci normalnej, jeśli jest w 1NF i nie ma zależności częściowych (unikamy redundancji).

3NF — relacja jest w trzeciej postaci normalnej, jeśli jest w 2NF i nie ma zależności przechodnich (unikamy redundancji).

3.5NF — BCNF (unikamy redundancji i upraszaczmy sprawdzanie zależności).

4NF — GR(idg,idk,idp,limit), TS(idg,termin,sala), Z(ids,idg), PLAN(ids,idg,termin,sala) — w planie studenta muszą znaleźć się wszystkie terminy grupy, do której student się zapisał; relacja jest w czwartej postaci normalnej, jeśli jest w BCNF i nie ma nietrywialnych **zależności wielowartościowych**.

5NF — Pr(firma,sok), Pd(sok,bar), D(firma,bar), MożnaZamówić(sok,bar,firma) — sok można zamówić, jeśli bar go podaje, firma — produkuje i jest dostawcą baru; relacja jest w piątej postaci normalnej, jeśli jest w 4NF i nie ma nietrywialnych **zależności złączeniowych**.

# Postaci normalne 1-5 i in.

- 1NF — relacja jest w pierwszej postaci normalnej, jeśli ma atrybuty bez wewnątrznej struktury i bez powtórzeń.
- 2NF — relacja jest w drugiej postaci normalnej, jeśli jest w 1NF i nie ma zależności częściowych (unikamy redundancji).
- 3NF — relacja jest w trzeciej postaci normalnej, jeśli jest w 2NF i nie ma zależności przechodnich (unikamy redundancji).
- 3.5NF — BCNF (unikamy redundancji i upraszaczmy sprawdzanie zależności).
- 4NF — GR(idg,idk,idp,limit), TS(idg,termin,sala), Z(ids,idg), PLAN(ids,idg,termin,sala) — w planie studenta muszą znaleźć się wszystkie terminy grupy, do której student się zapisał; relacja jest w czwartej postaci normalnej, jeśli jest w BCNF i nie ma nietrywialnych **zależności wielowartościowych**.
- 5NF — Pr(firma,sok), Pd(sok,bar), D(firma,bar), MożnaZamówić(sok,bar,firma) — sok można zamówić, jeśli bar go podaje, firma — produkuje i jest dostawcą baru; relacja jest w piątej postaci normalnej, jeśli jest w 4NF i nie ma nietrywialnych **zależności złączeniowych**.
- inne — unikalność `ido` w PR i ST, przestrzeganie limitu w grupie, brak kolizji w salach, podawanie świeżych soków tylko w wyznaczonych barach.

# Zależności wielowartościowe i 4NF

# Zależności wielowartościowe i 4NF

## Definition (Zależność wielowartościowa)

Mamy:

- relację $R$ ze zbiorem zależności funkcyjnych $F$
- podzbiory atrybutów $\alpha, \beta \subseteq R$ i $\gamma = R \setminus (\alpha \cup \beta)$,
- $r$ — poprawny stan $R$.

Wtedy zachodzi zależność wielowartościowa $\alpha \rightarrow\rightarrow \beta$, o ile

- jeśli mamy $t_1, t_2 \in r$ oraz $t_1.\alpha = t_2.\alpha = a$, to
- do $r$ należą także $s_1, s_2$, takie że $s_1 = (a, t_1.\beta, t_2.\gamma)$ oraz $s_2 = (a, t_2.\beta, t_1.\gamma)$

Zależność $\alpha \rightarrow\rightarrow \beta$ jest **trywialna**, jeśli $\beta \subseteq \alpha$ lub $\gamma \subseteq \alpha$.

# Zależności wielowartościowe i 4NF

## Definition (Zależność wielowartościowa)

Mamy:

- relację $R$ ze zbiorem zależności funkcyjnych $F$
- podzbiory atrybutów $\alpha, \beta \subseteq R$ i $\gamma = R \setminus (\alpha \cup \beta)$,
- $r$ — poprawny stan $R$.

Wtedy zachodzi zależność wielowartościowa $\alpha \to\to \beta$, o ile

- jeśli mamy $t_1, t_2 \in r$ oraz $t_1.\alpha = t_2.\alpha = a$, to
- do $r$ należą także $s_1, s_2$, takie że $s_1 = (a, t_1.\beta, t_2.\gamma)$ oraz $s_2 = (a, t_2.\beta, t_1.\gamma)$

Zależność $\alpha \to\to \beta$ jest **trywialna**, jeśli $\beta \subseteq \alpha$ lub $\gamma \subseteq \alpha$.

## Definition (Czwarta postać normalna, 4NF)

Relacja $R$ z zależnościami funkcyjnymi $F$ i wielowartościowymi $M$ jest w czwartej postaci normalnej, jeśli każda nietrywialna zależność $\alpha \to\to \beta \in M$ wynika z nadklucza, tzn. $(\alpha)^+_F = R$.

# Zależności wielowartościowe i 4NF

## Definition (Zależność wielowartościowa)

Mamy:

- relację $R$ ze zbiorem zależności funkcyjnych $F$
- podzbiory atrybutów $\alpha, \beta \subseteq R$ i $\gamma = R \setminus (\alpha \cup \beta)$,
- $r$ — poprawny stan $R$.

Wtedy zachodzi zależność wielowartościowa $\alpha \rightarrow\rightarrow \beta$, o ile

- jeśli mamy $t_1, t_2 \in r$ oraz $t_1.\alpha = t_2.\alpha = a$, to
- do $r$ należą także $s_1, s_2$, takie że $s_1 = (a, t_1.\beta, t_2.\gamma)$ oraz $s_2 = (a, t_2.\beta, t_1.\gamma)$

Zależność $\alpha \rightarrow\rightarrow \beta$ jest **trywialna**, jeśli $\beta \subseteq \alpha$ lub $\gamma \subseteq \alpha$.

## Definition (Czwarta postać normalna, 4NF)

Relacja $R$ z zależnościami funkcyjnymi $F$ i wielowartościowymi $M$ jest w czwartej postaci normalnej, jeśli każda nietrywialna zależność $\alpha \rightarrow\rightarrow \beta \in M$ wynika z nadklucza, tzn. $(\alpha)^+_F = R$.

**Uwaga:** Relację $R$ z nietrywialną zależnością $\alpha \rightarrow\rightarrow \beta$, gdzie $\alpha \cap \beta = \emptyset$ można bezpiecznie (odwracalnie i z zachowaniem zależności) rozłożyć na składowe $R_1 = \alpha\beta$ oraz $R_2 = \alpha \cup (R \setminus \beta)$.

# Zależności generujące krotki (tgds)

Kolejnym rodzajem zależności są *zależności generujące krotki (tuple-generating dependencies)*:

$$\forall \vec{x}\vec{y}\varphi(\vec{x},\vec{y}) \rightarrow \exists \vec{z}\psi(\vec{x},\vec{z})$$

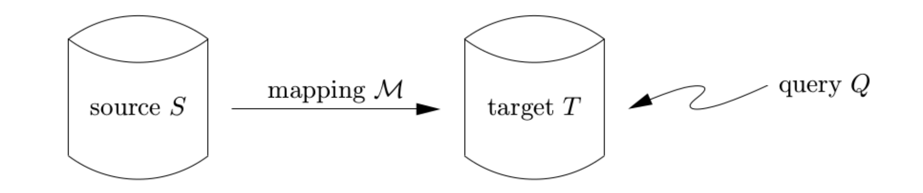
Dalsza część wykładu wg.: Arenas et. al. *Relational and XML Data Exchange.* Morgan and Claypool Publishers.

# Zależności generujące krotki (tgds)

Kolejnym rodzajem zależności są *zależności generujące krotki (tuple-generating dependencies)*:

$$\forall \vec{x}\vec{y}\varphi(\vec{x},\vec{y}) \to \exists \vec{z}\psi(\vec{x},\vec{z})$$

- Zależności tgds są świetnym narzędziem do modelowania migracji schematów (wymiana danych).




Dalsza część wykładu wg.: Arenas et. al. *Relational and XML Data Exchange.* Morgan and Claypool Publishers.

# Wymiana danych (data exchange).

- Chcemy stworzyć bazę o schemacie:

```
ROUTES(flight#,source,destination)
INFO_FLIGHT(flight#,departure_time,arrival_time,airline)
SERVES(airline,city,country,phone)
```

# Wymiana danych (data exchange).

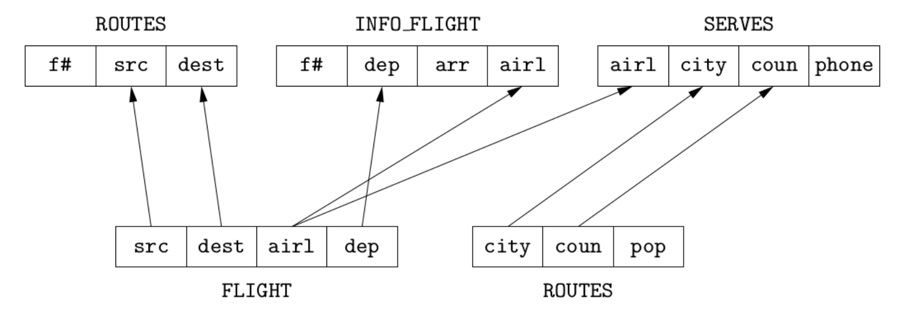
- Chcemy stworzyć bazę o schemacie:

  ```
  ROUTES(flight#,source,destination)
  INFO_FLIGHT(flight#,departure_time,arrival_time,airline)
  SERVES(airline,city,country,phone)
  ```

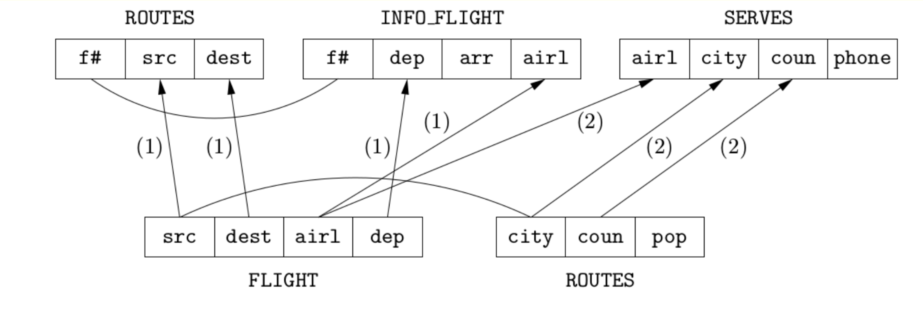
- Obecnie dane mamy w bazie o schemacie:

  ```
  FLIGHT(source,destination,airline,departure)
  GEO(city,country,population)
  ```

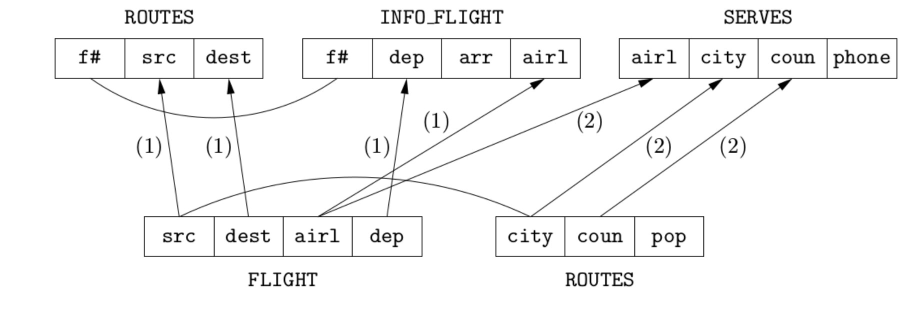
# Mapowanie schematów za pomocą tgds.



# Mapowanie schematów za pomocą tgds.

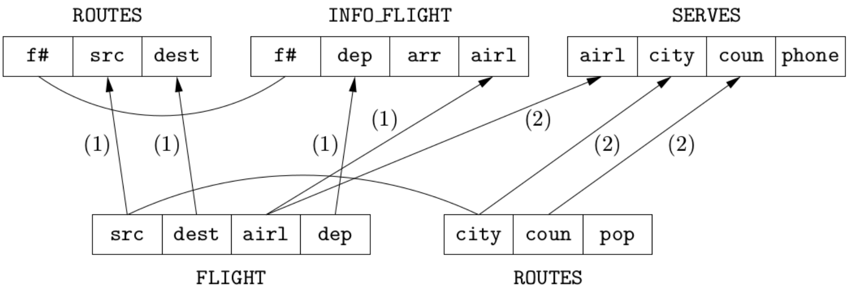


# Mapowanie schematów za pomocą tgds.




```
FLIGHT(src,dest,airl,dep) ⟶
      ROUTES(_,src,dest),INFO_FLIGHT(_,dep,_,airl)
```

# Mapowanie schematów za pomocą tgds.




$$\texttt{FLIGHT(src,dest,airl,dep)} \longrightarrow$$
$$\texttt{ROUTES(\_,src,dest),INFO\_FLIGHT(\_,dep,\_,airl)}$$

$$\forall\texttt{src}\ \forall\texttt{dest}\ \forall\texttt{airl}\ \forall\texttt{dep}\ \Big(\texttt{FLIGHT(src,dest,airl,dep)} \longrightarrow$$
$$\exists\texttt{f\#}\ \exists\texttt{arr}\ \big(\texttt{ROUTES(f\#,src,dest)} \wedge \texttt{INFO\_FLIGHT(f\#,dep,arr,airl)}\big)\Big)$$

# Mapowanie schematów za pomocą tgds.

$$(1)\quad \mathtt{FLIGHT(src,dest,airl,dep)} \longrightarrow \exists \mathtt{f\#}\, \exists \mathtt{arr} \left( \mathtt{ROUTES(f\#,src,dest)} \wedge \mathtt{INFO\_FLIGHT(f\#,dep,arr,airl)} \right)$$

$$(2)\quad \mathtt{FLIGHT(city,dest,airl,dep)} \wedge \mathtt{GEO(city,country,popul)} \longrightarrow \exists \mathtt{phone}\ \mathtt{SERVES(airl,city,country,phone)}$$

$$(3)\quad \mathtt{FLIGHT(src,city,airl,dep)} \wedge \mathtt{GEO(city,country,popul)} \longrightarrow \exists \mathtt{phone}\ \mathtt{SERVES(airl,city,country,phone)}$$

# Mapowanie schematów za pomocą tgds.

- Dla danej źródłowej bazy danych $S$ i każdej zależności $\varphi$, docelowa baza danych $T$ musi spełniać
$$(S, T) \models \varphi$$

# Mapowanie schematów za pomocą tgds.

- Dla danej źródłowej bazy danych $S$ i każdej zależności $\varphi$, docelowa baza danych $T$ musi spełniać
$$(S, T) \models \varphi$$
- Jednak takich $T$ może istnieć wiele...

# Mapowanie schematów za pomocą tgds.

- Dla danej źródłowej bazy danych $S$ i każdej zależności $\varphi$, docelowa baza danych $T$ musi spełniać
$$(S, T) \models \varphi$$
- Jednak takich $T$ może istnieć wiele...
- Rozwiązanie uniwersalne $U$ - istnieje homomorfizm z $U$ w dowolną bazę docelową $T$ (każde $T$ jest *uszczegółowieniem* $U$).

# Mapowanie schematów za pomocą tgds.

- Dla danej źródłowej bazy danych $S$ i każdej zależności $\varphi$, docelowa baza danych $T$ musi spełniać
$$(S, T) \models \varphi$$
- Jednak takich $T$ może istnieć wiele...
- Rozwiązanie uniwersalne $U$ - istnieje homomorfizm z $U$ w dowolną bazę docelową $T$ (każde $T$ jest *uszczegółowieniem* $U$).
- Jak znaleźć rozwiązanie uniwersalne?

# Mapowanie schematów za pomocą tgds.

- Dla danej źródłowej bazy danych $S$ i każdej zależności $\varphi$, docelowa baza danych $T$ musi spełniać
$$(S, T) \models \varphi$$
- Jednak takich $T$ może istnieć wiele...
- Rozwiązanie uniwersalne $U$ - istnieje homomorfizm z $U$ w dowolną bazę docelową $T$ (każde $T$ jest *uszczegółowieniem* $U$).
- Jak znaleźć rozwiązanie uniwersalne? Pogoń (*chase*).

# Mapowanie schematów za pomocą tgds.

- Dla danej źródłowej bazy danych $S$ i każdej zależności $\varphi$, docelowa baza danych $T$ musi spełniać
$$(S, T) \models \varphi$$
- Jednak takich $T$ może istnieć wiele...
- Rozwiązanie uniwersalne $U$ - istnieje homomorfizm z $U$ w dowolną bazę docelową $T$ (każde $T$ jest *uszczegółowieniem* $U$).
- Jak znaleźć rozwiązanie uniwersalne? Pogoń (*chase*).
- Jak odpowiadać na zapytania skoro jest wiele rozwiązań?

# Mapowanie schematów za pomocą tgds.

- Dla danej źródłowej bazy danych $S$ i każdej zależności $\varphi$, docelowa baza danych $T$ musi spełniać
$$(S, T) \models \varphi$$
- Jednak takich $T$ może istnieć wiele. . .
- Rozwiązanie uniwersalne $U$ - istnieje homomorfizm z $U$ w dowolną bazę docelową $T$ (każde $T$ jest *uszczegółowieniem* $U$).
- Jak znaleźć rozwiązanie uniwersalne? Pogoń (*chase*).
- Jak odpowiadać na zapytania skoro jest wiele rozwiązań?
Pewne odpowiedzi (*certain answers*) to odpowiedzi prawdziwe w dowolnej bazie docelowej.

# Mapowanie schematów za pomocą tgds.

- Dla danej źródłowej bazy danych $S$ i każdej zależności $\varphi$, docelowa baza danych $T$ musi spełniać
$$(S, T) \models \varphi$$
- Jednak takich $T$ może istnieć wiele...
- Rozwiązanie uniwersalne $U$ - istnieje homomorfizm z $U$ w dowolną bazę docelową $T$ (każde $T$ jest *uszczegółowieniem* $U$).
- Jak znaleźć rozwiązanie uniwersalne? Pogoń (*chase*).
- Jak odpowiadać na zapytania skoro jest wiele rozwiązań?
  Pewne odpowiedzi (*certain answers*) to odpowiedzi prawdziwe w dowolnej bazie docelowej.
- Skoro nasze zapytania (CQs) są monotoniczne to wystarczy zmaterializować (jakieś) rozwiązanie uniwersalne i na nim wyliczyć odpowiedzi.