Normalformen

Normalisierung

- Normalformen definieren Qualitätskriterien (Vermeidung der Inkonsistenzen)
- Redundanz ist oft die Ursache von Schemata Probleme (keine FDs → keine Redundanz)
- Normalisierung:
  - Jede Relation entspricht genau einer Objektmenge oder genau einer Relationship-Menge zwischen Objekten
  - Alle Informationen und Integritätsbedingungen (Schlüsselkandidaten und Fremdschlüssel) sind abgebildet
  - Redundanz ist weitgehend eliminiert
  - Es treten keine Änderungsanomalien auf

  ⇒ Beseitigung von Abhängigkeiten innerhalb der Relation

Normalisierung

- Integritätsregeln, insbesondere FDs, helfen ein schlechtes Schema zu identifizieren
- Die Lösung ist meistens Zerlegung der Relation
- **Achtung**: wann ist eine Zerlegung notwendig und wann ist sie möglich?

Zerlegung einer Relation

- Zerlegung notwendig
  - Ist die Relation mit Blick auf FDs redundanzfrei?
    - Ja: Keine Zerlegung notwendig
    - Nein: Starte Zerlegungsprozedur
- Zerlegungsprozedur
  - Zerlegung der Relation R in R_1 und R_2:
  - Ist dies informationsverlustfrei möglich?
    - Ja: OK
    - Nein: keine Zerlegung möglich!
  - Aufspaltung der zugehörigen FDs in FD1s und FD2s, die jeweils R_1 und R_2 zugeordnet werden könnten
  - Ist dies abhängigkeitserhaltend möglich?
    - Ja: OK
    - Nein: Ok, aber „unschön", da Überprüfung der FDs nur nach der Rekonstruktion von R möglich ist (Effizientverlust)

Normalformen

- Wenn eine Relation in einer Normalform ist, dann wissen wir dass bestimmte Probleme nicht vorkommen können.
- Es hilft um zu bestimmen ob eine Zerlegung weiter hilft oder nicht.
- Die Normalformen basierend auf FDs: 1NF, 2NF, 3NF, BCNF

$$\mathbf{BCNF} \subseteq \mathbf{3NF} \subseteq \mathbf{2NF} \subseteq \mathbf{1NF}$$

Normalformen

- 1. Normalform (1NF)
  - Keine mehrwertigen Attribute
- 2. Normalform (2NF)
  - Keine Vermischung von Sachverhalten in Relationen
- 3. Normalform (3NF)
  - Keine funktionalen Abhängigkeiten von Nichtschlüsselattributen innerhalb von Relationen
- Weitere Normalformen
  - Boyce-Codd- Normalform (BCNF)
  - 4. Normalform (4NF)
  - 5. Normalform (5NF)

1. Normalform

- **Definition.** Eine Relation ist in der ersten Normalform, wenn alle Attribute der Relation **atomar** sind. Zusammengesetzte und mehrwertige Attribute sind nicht erlaubt.
- Ist bei der benutzten Definition des relationalen Modells automatisch eingehalten
- Es gibt aber auch Datenbank Management Systeme welche Datenbanken, die nicht in 1NF sind, erlauben
- Atomar: String, Integer, etc.

Relation nicht in 1NF

Kunden (*<u>KundenID</u>, Name, Vorname, Produkte*)

Transformation in 1NF durch Zerlegung

Kunden (*<u>KundenID</u>, Name, Vorname*)
KundeProdukt *(<u>KundenID, Produkt</u>*)

Referenzielle Integritätsbedingung (Fremdschlüssel):
KundeProdukt.*KundenID* referenziert Kunden.*KundenID*

<u>KundenID</u>	**Name**	**Vorname**	**Produkte**
K1	Nuhr	Dieter	S II
K2	Pelzig	Erwin	iPhone 5, iPad, Vaio
K3	Gruber	Monika	Nexus, Galaxy Tab, Kindle

<u>KundenId</u>	**Name**	**Vorname**
K1	Nuhr	Dieter
K2	Pelzig	Erwin
K3	Gruber	Monika

<u>KundenID</u>	**<u>Produkt</u>**
K1	S II
K2	iPhone 5
K2	iPad
K2	Vaio
K3	Nexus
K3	Galaxy Tab
K3	Kindle

2. Normalform

- **Definition.** Eine Relation R mit zugehörigen FDs F ist in zweiter Normalform, genau dann wenn:
  - sie in 1NF ist **und**
  - jedes Nichtschlüssel-Attribut $A \in R$ voll funktional abhängig ist von jedem Kandidatenschlüssel der Relation.

- Eine funktionalle Abhängigkeit $X \rightarrow Y$ heißt **voll**, wenn es keine echte Teilmengen $Z \subset X$ gibt, s.d. gilt $Z \rightarrow Y$

2. Normalform

- **Bemerkung.**
  - Wenn eine Relation in 1NF, aber nicht in 2NF ist, dann enthält die Relation wenigstens ein zusammengesetztes Kandidatschlüssel.
  - Wenn eine Relation in 1NF, aber nicht in 2NF ist, dann gibt es eine funktionale Abhängigkeit $\alpha \rightarrow \beta$, sodas α in einem Kandidatschlüssel enthalten ist

- Intuitiv: Ein Relationenschema verletzt 2NF, wenn in der Relation Informationen über mehr als ein einziges Konzept modelliert werden.
- Selbst bei Erfüllung der 2NF können immer noch Redundanzen im Schema enthalten sein (durch transitive Abhängigkeiten)

2 NF - Beispiel

- StudentenBelegung (<u>MatrNr, VorlNr</u>, Name, Semester)
- Primärschlüssel: {MatrNr, VorlNr}
- FDs wodurch die 2NF verletzt wird:
  - {MatrNr} $\rightarrow$ {Name}
  - {MatrNr} $\rightarrow$ {Semester}
- Bemerkung. Hier können wir auch die Anomalien wiedererkennen.
- Zerlegung:

  Hören(<u>MatrNr</u>, <u>VorlNr</u>)

  Studenten(<u>MatrNr</u>, Name,Semester)

3. Normalform

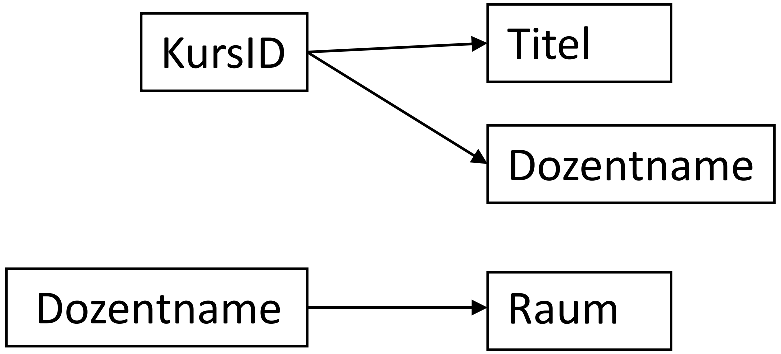
- **Definition 1.** Eine Relation mit zugehörigen FDs F ist in der 3. Normalform, wenn für alle Abhängigkeiten $A \rightarrow B$ aus F^+ gilt:
  - $B \subseteq A$ (FD ist trivial) **oder**
  - A enthält einen Schlüssel von R (A ist ein Superschlüssel) **oder**
  - B ist Teil eines Schlüsselkandidaten (B ist prim)
- **Definition 2.** Eine Relation R ist in 3NF, wenn sie:
  - in der 2NF ist **und**
  - kein Nichtschlüsselattribut von einem Schlüsselkandidaten transitiv abhängt.
- **Bemerkung.** 3NF beseitigt Abhängigkeiten von Nicht-Schlüsselattributen.
- Selbst bei Erfüllung der 3NF sind Redundanzen möglich.
- Eine Relation bleibt in 3NF, wenn BCNF nicht erreichbar ist (keine gute Zerlegung oder aus Leistungsgründen)

Relation nicht in 3NF

- Kurs (*KursID, Titel, DozentName, Raum*) - nicht in 3NF
- Ist die Relation in 2NF?
- Es gilt: {*DozentName*} → {*Raum*}
- Aber: Dozentname ist kein Schlüssel **und** Raum ist nicht Teil eines Schlüsselkandidaten
- Folgende Anomalien können auftreten:
  - Dozenten und Raum sind ohne Zuordnung eines Kurses nicht vefügbar
  - Falls einen Dozent keine Kurs gibt, werden alle Informationen über den Dozent und seinem Raum gelöscht
- Schema in 3NF:

  Kurs (*KursID, Titel, DozentName*)

  Dozent (*DozentName, Raum*)



Boyce-Codd Normalform

- **Definition.** Eine Relation R mit zugehörigen FDs F ist in der Boyce-Codd Normalform, wenn für alle Abhängigkeiten $A \rightarrow B$ aus F^+ gilt:
  - $B \subseteq A$ (FD ist trivial) **oder**
  - A enthält einen Schlüssel von R (A ist ein Superschlüssel)
- Die BCNF hat also außer den trivialen nur noch funktionalen Abhängigkeiten deren Determinante (linke Seite) ein Superschlüssel ist
- **Bemerkung.** Wenn R in BCNF ist, ist es automatisch auch in 3NF.

Relation in 3NF aber nicht BCNF

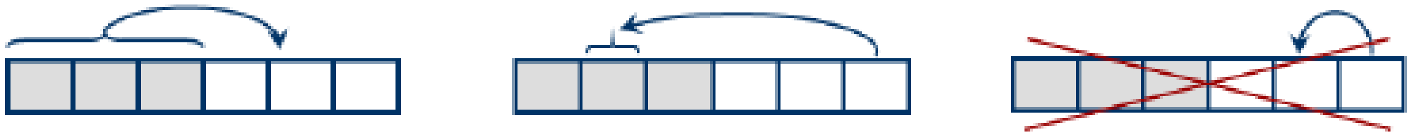
- Städte (*Ort, BLand, Ministerpräsident, Einw*)
- Schlüsselkandidaten:
  - (*Ort, BLand*)
  - (*Ort, Ministerpräsident*)
- FDs:
  - *BLand* → *Ministerpräsident*
  - {*Ort, BLand*} → *Einw*
  - *Ministerpräsident* → *BLand*
- Relation ist in **3NF**, aber **nicht** in **BCNF**
- Anomalien können auftreten, da die Information, wer welches Bundesland regiert, mehrfach abgespeichert wird

Normalformen - Zusammenfassung

1NF – alle Attribute sint atomar

2NF – alle Nichtschlüsselattribute sind voll funktional abhängig von jedem Kandidatenschlüssel (keine partielle Abhängigkeiten)

3NF – in 2NF und alle Nichtschlüsselattribute sind nur von Kandidatenschlüssel abhängig (keine transitive Abhängigkeiten)



BCNF – jede Determinante ist ein Superschlüssel (alle FDs werden von Kandidatenschlüssel bestimmt)

3NF vs. BCNF

- **Bemerkung.** Man kann jede Relation R so in $R_1,\ldots, R_n$ zerlegen, dass gilt:
  - Die Zerlegung ist **verlustlos** und **abhängigkeitsbewahrend**
  - R_i ist in **3NF**, $1 \leq i \leq n$
- **Bemerkung.** Man kann jede Relation R so in $R_1,\ldots, R_n$ zerlegen, dass gilt:
  - Die Zerlegung ist **verlustlos**
  - R_i ist in **BCNF**, $1 \leq i \leq n$
- Aber man kann nicht immer eine BCNF-Zerlegung finden, die auch abhängigkeitsbewahrend ist.

BCNF - Zerlegung

- Wenn die funktionale Abhängigkeit $\alpha \rightarrow \beta$ die BCNF verletzt, dann können wir die Relation in $R - \beta$ und $\alpha \cup \beta$ zerlegen. Wir können das weitermachen bis alle neuen Relation in BCNF sind (es geht immer zu Ende).
- Diese Zerlegung wird verlustlos, aber nicht unbedingt abhängigkeitsbewahrend sein
- **Bem**. Wenn es mehrere Abhängigkeiten gibt welche die BCNF verletzten, dann macht es einen Unterschied welche wir als erste für die Zerlegung auswählen.

BCNF – Zerlegung Beispiel

- R($\underline{C}$, S , J, D, P, Q, V)
- $F = \{ JP \rightarrow C, \quad SD \rightarrow P, \quad J \rightarrow S \}$
- $SD \rightarrow P$ verletzt BCNF $\Rightarrow$ Zerlegung ($\underline{S,D}$,P) und ($\underline{C}$,S,J,D,Q,V)
- $J \rightarrow S$ verletzt BCNF für die zweite Relation $\Rightarrow$ ($\underline{C}$,S,J,D,Q,V) wird in

  ($\underline{J}$, S) und ($\underline{C}$, J, D, Q, V) zerlegt
- D.h. R wird in ($\underline{S,D}$,P) , ($\underline{J}$, S) und ($\underline{C}$, J, D, Q, V) zerlegt – verlustlos, aber nicht abhängigkeitsbewahrend

3NF - Zerlegung

- Normalerweise: per Hand gemacht und überprüft

- Eine 3NF Zerlegung die verlustlos und abhängigkeitsbewahrend ist, ist möglich, aber wir müssen dafür nicht die ganze Menge der fkt. Abh. benutzen, sondern eine „minimale" Überdeckung von F
- Wichtigster Algorithmus: **3NF-Synthesealgorithmus**
- Braucht als Eingabe eine redundanzfreie Menge von FDs (**kanonische Überdeckung**)

Kanonische Überdeckung

- **Definition (Äquivalenz funktionaler Abhängigkeiten).**
  Zwei Mengen F und G von FDs eines Relationenschemas R sind äquivalent, falls $F^+ = G^+$ gilt.
- Wunsch: Berechne eine möglichst kleine Menge, die zu F äquivalent ist → geringer Aufwand beim Testen, ob ein neues Tupel eine FD verletzt

Kanonische Überdeckung

- Man kann **überflüssige Attribute** durch Links- und Rechtsreduktion entfernen:
- **Linksreduktion**
  - Führe für jede FD $A \rightarrow B$ aus F die Linksreduktion durch, indem für alle $X \in A$ überprüft wird ob das Attribut X überflüssig ist, d.h. ob gilt

    $B \subset$ Hülle(F, A - {X}) , also $(A - \{X\})^+$ in Beziehung zu F

    (anders gesagt (F - $\{A \rightarrow B\} \cup \{(A\text{-}\{X\}) \rightarrow B\}$ $)^+ = F^+$)

  Ist dies der Fall, ersetze $A \rightarrow B$ durch $A - \{X\} \rightarrow B$
- **Rechtsreduktion**
  - Führe für jede FD $A \rightarrow B$ aus F die Rechtsreduktion durch, indem für alle $Y \in B$ überprüft wird ob das Attribut Y überflüssig ist, d.h. ob gilt

    $Y \in$ Hülle(F – $(A \rightarrow B) \cup (A \rightarrow B - \{Y\})$, A)

    (anders gesagt (F - $\{A \rightarrow B\} \cup \{A \rightarrow (B - \{Y\})\}$ $)^+ = F^+$)

  Ist dies der Fall, ersetze $A \rightarrow B$ durch $A \rightarrow B - \{Y\}$

Kanonische Überdeckung

- **Definition (Kanonische Überdeckung).**

  Die Menge von FDs F_c wird als **kanonische Überdeckung** von F bezeichnet, falls folgende Bedingungen erfüllt sind:
  - $F_c^+ = F^+$ (Äquivalenz) (1)
  - Für alle FDs A → B in F_c gibt es keine überflüssigen Attribute in A und in B, d.h.
    - für alle Attribute C aus A gilt (F_c - {A → B} ∪ {(A-{C}) → B})$^+ \neq F^+$ (2)
    - für alle Attribute D aus B gilt (F_c - {A → B} ∪ {A → (B – {D})})$^+ \neq F^+$ (3)
    - jede linke Seite der FDs in F_c kommt nur einmal vor, d.h. Falls

      A → B und A → C, dann wird in F_c nur die FD A → B ∪ C vrewendet (4)
- Intuitiv: (2) – keine überflüssige Attribute auf der linken Seite von FDs

  (3) - keine überflüssige Attribute auf der rechten Seite von FDs

Berechnung der Kanonischen Überdeckung

- Schritt 1 : Linksreduktion
- Schritt 2 : Rechtsreduktion
- Schritt 3 : Entferne die FDs der Form $A \rightarrow \emptyset$
- Schritt 4 : Ersetze alle FDs der Form

  $A \rightarrow B_1, \ldots, A \rightarrow B_k$ durch $A \rightarrow B_1 \cup \cdots \cup B_k$

Kanonische Überdeckung - Beispiel

- F = {ABCD → E, E → D, A → B, AC → D}
- Schritt 1. Linksreduktion:
  - B und D sind überflüssig in ABCD → E ⇒ {AC → E, E → D, A → B, AC → D}
- Schritt 2. Rechtsreduktion:
  - D ist überflüssig in AC → D ⇒ {AC → E, E → D, A → B, AC → ∅}
- Schritt 3. ⇒ {AC → E, E → D, A → B}
- Schritt 4. –

$$\Rightarrow \quad F_c = \{AC \rightarrow E, E \rightarrow D, A \rightarrow B\}$$

Kanonische Überdeckung

- **Hausaufgabe:**
  - Berechne die kanonische Überdeckung von

    $F = \{A \rightarrow B, B \rightarrow C, AB \rightarrow C\}$
- **Bemerkung.** Die kanonische Überdeckung ist nicht eindeutig (es hängt von dem Auswahl der überflüssigen Attributen ab)

3NF - Synthesealgorithmus

- Ziel des Synthesealgorithmus:
  - Zerlegung einer Relation R mit funktionalen Abhängigkeiten F in Relationen $R_1,...,R_n$, so dass folgende Bedingungen erfüllt sind:
    - Kein Informationsverlust
    - Bewahrung der funktionalen Abhägigkeiten
    - $R_1,...,R_n$ in 3NF

3NF - Synthesealgorithmus

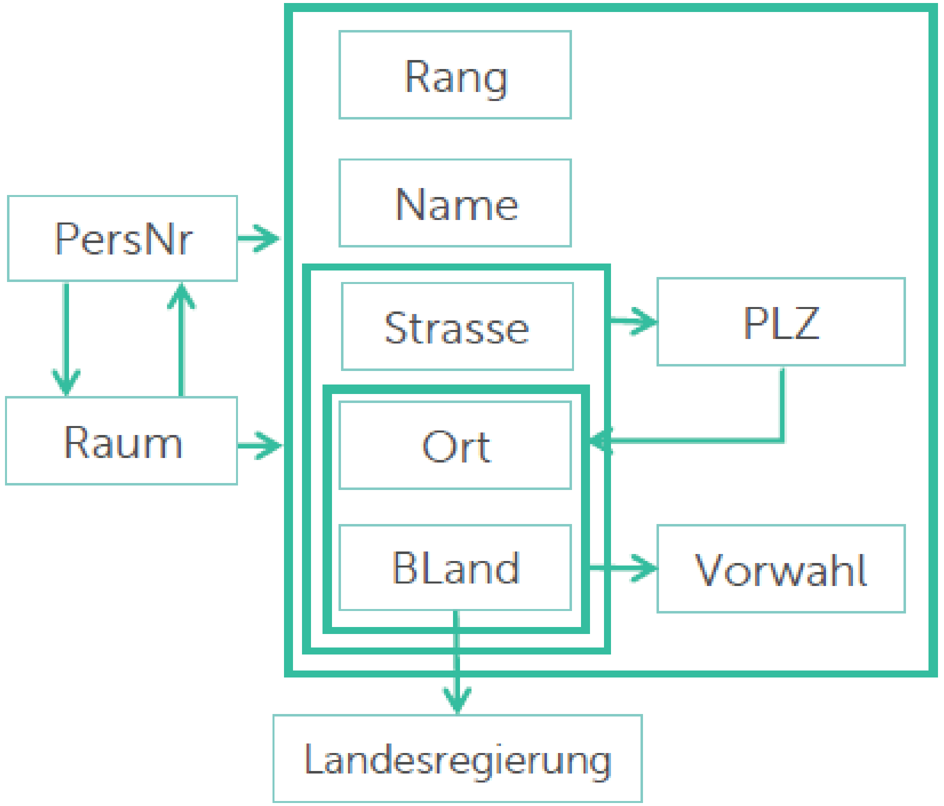
- Generierung einer derartigen Zerlgung
  1. Bestimme die kanonische Überdeckung F_c der Menge F
  2. Führe für jede FD $A \rightarrow B$ in F_c folgende Anweisungen:
     Erzeuge eine Relation $R_A = A \cup B$ und ordne R_A die FDs $F_A = \{C \rightarrow D \in F_c \mid C \cup D \subseteq R_A\}$ zu
  3. Falls alle Relationen erzeugt in Schritt 2 keinen Schlüsselkandidaten des ursprunglichen Relation R enthalten, so erzeuge zusätzlich eine neue Relation $R_K = K$ und $F_K = \emptyset$, wobei K ein Schlüsselkandidat von R ist
  4. Eliminiere die Relationen R_A, die in einem anderen Schema enthalten sind, d.h. $R_i \subseteq R_j$

Synthesealgorithmus - Beispiel

- R(A,B,C,D)
- F = {ABCD $\rightarrow$ E, E $\rightarrow$ D, A $\rightarrow$ B, AC $\rightarrow$ D}
- Kanonische Überdeckung $F_c = \{AC \rightarrow E, E \rightarrow D, A \rightarrow B\}$
- Einzige Kandidatenschlüssel : AC ($AC^+ = R$)
- R ist nicht 3NF wegen A $\rightarrow$ B
- 3NF Zerlegung von R:
  - R_1(A,C,E) mit F_1={AC $\rightarrow$ E}, R_2(E,D) mit F_2={E $\rightarrow$ D}, R_3(A,B) mit F_3={A $\rightarrow$ B}
  - R_1 enthält den Schlüssel AC

Synthesealgorithmus

- **Hausaufgabe**
  - Professoren(PersNr, Raum, Rang, Name, Strasse, Ort, BLand, Landesregierung, PLZ, Vorwahl)
  - FDs – in der Figur
  - Ist die Relation in 3NF? Warum ja/nicht?
  - Falls nicht, finde eine 3NF Zerlegung.



3NF Zerlegung

- **Bemerkung**. Eine 3NF Zerlegung ist nicht eindeutig. Es hängt vom Folgenden ab:
  - Auswahl der kanonischen Überdeckung
  - Auswahl der Relation, die in einer anderen enthalten ist, zum Eliminieren
- **Bemerkung.** Zerlegung ist eine Lösung für Redundanzen und Anomalien, **aber** zu viel Zerlegung kann schlecht sein aus Leistungsgründe.
- Beispiel.
  - R(Prof, Institut, Tel, Raum) mit F ={Prof $\rightarrow$ (Institut, Tel, Raum)}
  - Die Zerlegung R_1 = {Prof, Institut}, R_2= {Prof, Tel}, R_3= {Prof, Raum} ist korrekt, aber nicht notwendig

Mehrwertige Abhängigkeiten

- Verallgemeinerung funktionaler Anhängigkeiten
- Beispiel:
  - Fähigkeiten(PersNr, Sprache, Programmiersprache)
  - In diesem Schema gibt es keine FDs → BCNF, trotzdem haben wir Rendundanz
  - Eine Person kann mehrere Sprachen sprechen
  - Eine Person beherrscht mehrere Programmiersprachen
  - Sprache und Programmiersprache sind mehrwertig abhängig von PersNr

PersNr	**Sprache**	**ProgSprache**
3002	Deutsch	C
3002	Englisch	C
3002	Englisch	Java
3002	Deutsch	Java
3005	Englisch	C
3005	Deutsch	C

Mehrwertige Abhängigkeiten

- In der vorigen Schema werden voneinenader unabhängige Konzepte miteinander vermischt
- Kompaktere Speicherung:

PersNr	**Sprache**
3002	Deutsch
3002	Englisch
3005	Englisch
3005	Deutsch

PersNr	**ProgSprache**
3002	C
3002	Java
3005	C

Mehrwertige Abhängigkeiten (MVD)

- **Definition.** Sei R eine Relation und $\alpha, \beta \subseteq$ R, γ = R - $\alpha\beta$ Dann ist β **mehrwertig abhängig** von α, $\boldsymbol{\alpha} \rightarrow\rightarrow \boldsymbol{\beta}$, wenn in jeder Ausprägung r der Relation R gilt:

$$t_1, t_2 \in r \text{ und } \pi_{\alpha}(t_1) = \pi_{\alpha}(t_2) \Rightarrow$$

$$\exists\ t_3 \in r \text{ so dass } \pi_{\alpha\beta}(t_1) = \pi_{\alpha\beta}(t_3) \text{ und } \pi_{\gamma}(t_2) = \pi_{\gamma}(t_3)$$

Als Konsequenz daraus, wenn wir t_2 und t_1 betrachten, folgt dass

$$\exists\ t_4 \in r \text{ so dass } \pi_{\alpha\beta}(t_2) = \pi_{\alpha\beta}(t_4) \text{ und } \pi_{\gamma}(t_1) = \pi_{\gamma}(t_4)$$

- Intuition: Für alle Tupel mit gleichem Wert für A kommen alle B-C Kombinationen vor

Inferenzregeln

- Komplement: $X \rightarrow\rightarrow Y \Rightarrow X \rightarrow\rightarrow R\text{-}XY$
- Mehrwertige Verstärkung: $X \rightarrow\rightarrow Y$, $Z \subseteq W \Rightarrow WX \rightarrow\rightarrow YZ$
- Mehrwertige Transitivität: $X \rightarrow\rightarrow Y$, $Y \rightarrow\rightarrow Z \Rightarrow X \rightarrow\rightarrow Z\text{-}Y$
- Verallgemeinerung: $X \rightarrow Y \Rightarrow X \rightarrow\rightarrow Y$
- Koaleszenz: $X \rightarrow\rightarrow Y$, $W \cap Y = \emptyset$, $W \rightarrow Z$, $Z \subseteq Y \Rightarrow X \rightarrow Y$ (Coalescence)

Vierte Normalform

- **Definition.** Sei R eine Relation, $X,Y \subseteq R$. R ist in $\mathbf{4NF}$ wenn für jede $\mathrm{MVD}\ \ X \to \to$ Y eine der folgenden Bedingungen gilt:
  - $Y \subseteq X \quad (1)$ oder
  - $XY = R \ \ (2)$ oder
  - X ist ein Superschlüssel (X enthält einen Schlüssel von R) (3)
- Bem. Wenn (1) oder (2) gilt, dann heißt die Abhängigkeit $X \to \to$ Y **trivial.**
- **Äquivalente Def.** Eine Relation R ist in 4NF, wenn sie in der BCNF ist und nur noch triviale MVDs hat.
- 4NF ist eine Verstärkung der BCNF

Vierte Normalform - Beispiel

- Fähigkeiten(PersNr, Sprache, Programmiersprache)
  - ist in der BCNF
  - Ist nicht in der 4NF
    - $\text{PersNr} \rightarrow \rightarrow \text{Sprache} \;\; \text{verletzt 4NF}$
- Zerlegung:
  - Sprachen(PersNr, Sprache) – 4NF

    MVD: PersNr → → Sprache
  - ProgrSprachen(PersNr, Programmiersprache) – 4NF

    MVD: PersNr → → Programmiersprache

Fünfte Normalform

- **Definition(Verbund-Abhängigkeit/ Join Dependency).** Eine Relation R genügt der Verbund-Abhängigkeit(JD) $\otimes$ $\{R_1, ..., R_n\}$ genau dann, wenn $\{R_1, ..., R_n\}$ eine verlustlose Zerlegung von R ist.
- Eine MVD X $\rightarrow\rightarrow$ Y der Relation R kann als Join-Abhängigkeit representiert werden wie folgt: $\otimes$ $\{XY, X(R\text{-}Y)\}$

Fünfte Normalform

- **Definition.** Eine Relation R ist in 5NF falls für jede Verbund-Abhängigkeit JD in R eine der folgenden Bedingungen gilt:
  - R_i = R für eine i (JD ist trivial)
  - JD wird durch Schlüssel von R verursacht
- $\text{Eine JD} \otimes$ {R_1, ..., R_n} wird durch Schlüssel von R verursacht, wenn jeder R_i ein Superschlüssel ist (ein Kandidatschlüssel enthält)

Fünfte Normalform

- Relation Angestellte *(<u>PersNr</u>, Abteilung, Name, Vorname, Age)*
- Die Relation könnte durch folgende Verbunde entstanden sein:
    - ((*PersNr, Abteilung*), (*PersNr, Name, Vorname, Age*))
    - ((*PersNr, Abteilung*), (*PersNr, Name*), (*PersNr, Vorname*), (*PersNr, Age*))
- Ein Verbund mit Hilfe von Schlüsselattributen (*PersNr*) ist problemlos, da jedem Schlüsselwert der einen, ein Schlüsselwert der anderen entspricht. In der Ergebnisrelation kommen auf diese Weise auch nur die Tupel der verschiedenen Relationen, die einen gemeinsamen Schlüssel haben.

Normalisierung – wie weit?

- In der Praxis spielt 5NF so gut wie keine Rolle. Es ist nicht immer sinnvoll alle Normalisierungsschritte durchzuführen.
- Die Realisierung von BCNF ist zu empfehlen und die Verhinderung einer Mehrwertigen Abhängigkeit auch, so dass die 4NF immer sichergestellt ist.