Equijoin. In diesem Speziallfall bestimmt die Selektionsbedingung die Gleichheit eines Attributes A von R und eines Attributes B von S.

$$R \bowtie_{A=B} S := \{r \cup s : r \in R \wedge s \in S \wedge r_{[A]} = s_{[B]}\}$$

Das ist äquivalent zu

$$\sigma_{[A=B]}(R \times S)$$

Natural Join. Ein Natural Join setzt sich zusammen aus einem Equijoin und dem Ausblenden gleicher Spalten. Für zwei Relationen $R(A_1, \ldots, A_n, B_1, \ldots, B_n)$ und $S(B_1, \ldots, B_n, C_1, \ldots, C_n)$ ist

$$R \bowtie S := \{r \cup s_{[C_1, \ldots, C_n]} : r \in R \wedge s \in S \wedge r_{[B_1, \ldots, B_n]} = s_{[B_1, \ldots, B_n]}\}$$

2 Entity Relationship Model

2.1 Kardinalitäten

Teilnehmerkardinalitäten.

- $E1$ steht in Relation zu 0 oder 1 $E2$
- $E2$ steht in Relation zu 1 bis n $E1$

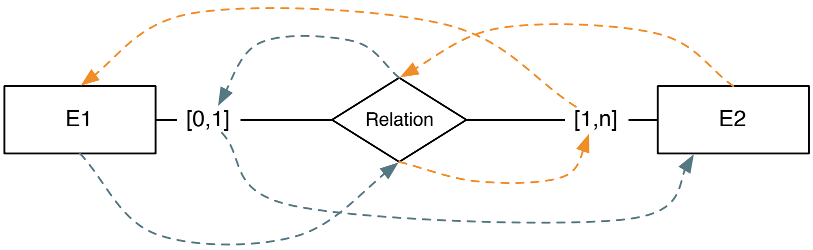

Figure 1: Leserichtung für Teilnehmerkardinalitäten

3 Relationaler Entwurf

Mehrwertige Abhängigkeit (Multi-Valued Dependency).

Universalrelation Die Universalrelation einer Menge von Relationen ist

$$R = R_1 \bowtie R_2 \bowtie \ldots R_n$$

3.1 Schlüssel

Superschlüssel. Die Attributmenge K ist ein Superschlüssel, falls sie die Tupel einer Relation eindeutig identifiziert, d.h es gilt die funktionale Abhängigkeit $K \to R$

Schlüsselkandidat. Die Attributmenge K ist ein Schlüsselkandidat, falls für das Relationenschema R die funktionale Abhängigkeit $K \to R$ gilt und K minimal ist.

Primärschlüssel. Aus der Menge aller Schlüsselkandidaten wird ein Primärschlüssel ausgewählt, um die Tupel der Relation eindeutig zu identifizieren.

Algorithm 1: Schlüssel finden

Input: Relation $R = (A_1, \dots, A_n)$, funktionale Abhängigkeiten F

$K \leftarrow \{\}$
for $X \to Y$ **in** F **do**
  $K \leftarrow K \cup X \backslash Y$
if $K^+ = R$ **then**
  if $\forall K' \subset K : K'^+ \neq R$ **then**
    return K

Hüllen. Die transitive Hülle F_R^+ einer Menge von funktionalen Abhängigkeiten F über der Relation R ist die Menge der funktionalen Abhängigkeiten, die von F impliziert werden:

$$F_R^+ := \{f : F \mid= f\}$$

Die Hülle einer Attributmenge X bezüglich einer Menge von funktionalen Abhängigkeiten F ist

$$X_F^* := \{A \ : \ X \to A \in F^+\}$$

Überdeckung

$$F \equiv G \Leftrightarrow F^+ = G^+$$

3.2 RAP-Algorithmus

Membership-Problem. Kann eine bestimmte funktionale Abhängigkeit $X \to Y$ aus einer Menge F abgeleitet werden? Gilt also

$$X \to Y \in F^+ \quad ?$$

Das modifizierte Membership-Problem

$$Y \subseteq X_F^*$$

kann durch den RAP-Algorithmus in Linearzeit (in der Anzahl der Attribute) gelöst werden.

RAP-Regeln

Reflexivität $\{\} \Rightarrow X \to X$

Akkumulation $\{X \to YZ, Z \to VW\} \Rightarrow X \to YZV, X \to YZW, \dots$

Projektivität $\{X \to YZ\} \Rightarrow X \to Y, X \to Z$

Algorithm 2: RAP-Algorithmus

Input: Attributmenge X, Attributmenge Y

$X^* \leftarrow X$
while X^* *nicht stabil* **do**
  if $\exists\ f_1 = X_1 \rightarrow Y_1 \in F,\ X_1 \subseteq X^*$ **then**
    $X^* \leftarrow X^* \cup Y_1$
if $Y \subseteq X^*$ **then**
  return wahr
else
  return falsch

Anomalien. Ein Relationenschema mit Redundanzen kann die Entstehung von Anomalien begünstigen, z.B.:

Einfügeanomalie Durch die Schlüsseldefinition muss zum Einfügen einer bestimmten Information mehr Information bzw. Null-Werte eingefügt werden.

Updateanomalie Ändert sich eine Information, so müssen mehrere Tupel aktualisiert werden, was aufwändig und fehleranfällig ist.

Löschanomalie Durch Löschen einer bestimmten Information geht mehr Information verloren als erwünscht.

Erwünschte Schemaeigenschaften

- **Redundanzen** vermeiden
- **Abhängigkeitstreue** besteht dann, wenn alle funktionalen Abhängigkeiten der Originalrelation auch in der zerlegten Relation noch gelten. Ein Relationenschema S ist abhängigkeitstreu bezüglich F wenn

$$F \equiv \{K \rightarrow R : (R, \mathcal{K}) \in S, K \in \mathcal{K}\}$$

- **Verbundtreue** bezeichnet die Möglichkeit, die Originalrelation aus der zerlegten Relation mittels Natural Joins wiederherstellen zu können.

Verbundtreue Die Dekomposition der Relation R in R_1 und R_2 ist verbundtreu, falls

$$R_1 \cap R_2 \rightarrow R_1 \quad \in F^+$$

oder

$$R_1 \cap R_2 \rightarrow R_2 \quad \in F^+$$

partielle Abhängigkeit liegt vor, wenn ein Nichtschlüsselattribut funktional schon von einem Teil des Schlüssels abhängt.

3.3 Normalisierung

1NF Jedes Attribut der Relation muss einen atomaren Wertebereich haben. Verbietet mengenwertige, geschachtelte oder zusammengesetzte Attribute.

2NF Jedes Nichtschlüsselattribut ist von jedem Schlüsselkandidaten voll funktional abhängig, d.h. abhängig vom ganzen Schlüssel, nicht nur von Teilen des Schlüssels.

3NF Kein Nichtschlüsselattribut hängt von einem Schlüsselkandidaten transitiv ab.

Boyce-Codd NF In allen Relationenschemata gehen die funktionalen Abhängigkeiten nur vom Primärschlüssel aus.

4NF Alle nicht-trivialen mehrwertigen Abhängigkeiten gehen vom Schlüsselkandidaten aus.

5NF

2NF: Eliminierung von partiellen Abhängigkeiten $(\underline{AB}CD)$ $A \to CD$ $(\underline{A}CD)$ $(\underline{AB})$

3.4 Syntheseverfahren

Ziel. Das Syntheseverfahren zerlegt eine Relation so, dass die 3NF erreicht wird bei gleichzeitiger Abhängigkeitstreue und Minimalität.

Algorithm 3: Syntheseverfahren

Input: Relation $R = (A_1, \ldots, A_n)$, funktionale Abhängigkeiten F

// führe weitere FD ein für Verbundtreue:
$F \leftarrow F \cup \{A_1 \ldots A_n \to \delta\}$
// zerlege FDs sodass rechte Seite atomar
for $X \to A_1 \ldots A_k$ **in** F_1 **do**
  $F \leftarrow F \backslash \{X \to A_1 \ldots A_k\} \cup \{X \to A_1, \ldots X \to A_k\}$
// eliminiere redundante FDs
for f **in** F **do**
  if $F \backslash \{f\} \equiv F$ **then**
    $F \leftarrow F \backslash \{f\}$
// entferne überflüssige Attribute auf der linken Seite
for $X \to Y$ **in** F **do**
  if $X' \to Y \in F, X' \subset X$ **then**
    $F \leftarrow F \backslash \{X \to Y\} \cup \{?\}$
// fasse FDs mit gleicher linker Seite zusammen
while $\exists X \to Y \wedge \exists X \to Z \in F$ **do**
  $F \leftarrow F \backslash \{X \to Y, X \to Z\} \cup \{X \to YZ\}$
//