

# Kapitel DB:VI

## VI. Entwurfstheorie relationaler Datenbanken

- ❑ Informelle Entwurfskriterien für Relationenschemata
- ❑ Funktionale Abhängigkeiten
- ❑ Normalformen
- ❑ Dekompositionseigenschaften von Relationen
- ❑ Relationale Dekomposition
- ❑ Relationale Synthese
- ❑ Mehrwertige Abhängigkeiten

# Informelle Entwurfskriterien für Relationenschemata

(1) Semantik von Relationenschemata klar halten

Datenbankschema mit einleuchtender Semantik:

## Mitarbeiter

Name PersNr Wohnort AbtNr

## Abteilung

AbtName AbtNr ChefPersNr

## Standort

AbtNr AbtOrt

## Projekt

ProjektName ProjektNr ProjektOrt AbtNr

## ArbeitetInProjekt

PersNr ProjektNr Stunden

# Informelle Entwurfskriterien für Relationenschemata

(1) Semantik von Relationenschemata klar halten

Datenbankschema mit einleuchtender Semantik:

| Mitarbeiter |               |         |       |
|-------------|---------------|---------|-------|
| Name        | <u>PersNr</u> | Wohnort | AbtNr |

| Abteilung |              |            |
|-----------|--------------|------------|
| AbtName   | <u>AbtNr</u> | ChefPersNr |

| Standort     |               |
|--------------|---------------|
| <u>AbtNr</u> | <u>AbtOrt</u> |

| Projekt     |                  |            |       |
|-------------|------------------|------------|-------|
| ProjektName | <u>ProjektNr</u> | ProjektOrt | AbtNr |

| ArbeitetInProjekt |                  |         |
|-------------------|------------------|---------|
| <u>PersNr</u>     | <u>ProjektNr</u> | Stunden |

Problematisches Datenbankschema:

| MitarbeiterAbteilung |               |         |       |         |            |
|----------------------|---------------|---------|-------|---------|------------|
| Name                 | <u>PersNr</u> | Wohnort | AbtNr | AbtName | ChefPersNr |

| MitarbeiterProjekt |                  |         |      |             |            |
|--------------------|------------------|---------|------|-------------|------------|
| <u>PersNr</u>      | <u>ProjektNr</u> | Stunden | Name | ProjektName | ProjektOrt |

# Informelle Entwurfskriterien für Relationenschemata

## (2) Vermeidung von (Update-)Anomalien

| Mitarbeiter |               |         |       |
|-------------|---------------|---------|-------|
| Name        | <u>PersNr</u> | Wohnort | AbtNr |
| Smith       | 1234          | Weimar  | 5     |
| Wong        | 3334          | Köln    | 5     |
| Zelaya      | 9998          | Erfurt  | 4     |
| Wallace     | 9876          | Berlin  | 4     |

⊗

| Abteilung  |              |            |
|------------|--------------|------------|
| AbtName    | <u>AbtNr</u> | ChefPersNr |
| Forschung  | 5            | 3334       |
| Verwaltung | 4            | 9876       |
| Stab       | 1            | 8886       |

=

| MitarbeiterAbteilung |               |         |       |            |            |
|----------------------|---------------|---------|-------|------------|------------|
| Name                 | <u>PersNr</u> | Wohnort | AbtNr | AbtName    | ChefPersNr |
| Smith                | 1234          | Weimar  | 5     | Forschung  | 3334       |
| Wong                 | 3334          | Köln    | 5     | Forschung  | 3334       |
| Zelaya               | 9998          | Erfurt  | 4     | Verwaltung | 9876       |
| Wallace              | 9876          | Berlin  | 4     | Verwaltung | 9876       |

Redundanz

# Informelle Entwurfskriterien für Relationenschemata

## (2) Vermeidung von (Update-)Anomalien (Fortsetzung)

| MitarbeiterAbteilung |               |         |       |            |            |
|----------------------|---------------|---------|-------|------------|------------|
| Name                 | <u>PersNr</u> | Wohnort | AbtNr | AbtName    | ChefPersNr |
| Smith                | 1234          | Weimar  | 5     | Forschung  | 3334       |
| Wong                 | 3334          | Köln    | 5     | Forschung  | 3334       |
| Zelaya               | 9998          | Erfurt  | 4     | Verwaltung | 9876       |
| Wallace              | 9876          | Berlin  | 4     | Verwaltung | 9876       |

Einfüge- bzw. Insert-Anomalie:

1. Beim Einfügen eines neuen Tupels ist sicherzustellen, dass für redundante Attribute die richtigen Werte eingetragen werden.

Im Beispiel: AbtNr, AbtName, ChefPersNr

2. Das Einfügen von Tupeln, für die nicht alle Attribute bekannt sind, ist problematisch.

Im Beispiel: Wie soll eine neue Abteilung ohne Mitarbeiter eingetragen werden? Beachte, dass das Attribut „PersNr“ Schlüssel dieser Relation ist und hier keine Nullwerte eingetragen werden dürfen.

# Informelle Entwurfskriterien für Relationenschemata

## (2) Vermeidung von (Update-)Anomalien (Fortsetzung)

| MitarbeiterAbteilung |               |         |       |            |            |
|----------------------|---------------|---------|-------|------------|------------|
| Name                 | <u>PersNr</u> | Wohnort | AbtNr | AbtName    | ChefPersNr |
| Smith                | 1234          | Weimar  | 5     | Forschung  | 3334       |
| Wong                 | 3334          | Köln    | 5     | Forschung  | 3334       |
| Zelaya               | 9998          | Erfurt  | 4     | Verwaltung | 9876       |
| Wallace              | 9876          | Berlin  | 4     | Verwaltung | 9876       |

Lösch- bzw. Delete-Anomalie:

Das Löschen von Tupeln ist problematisch, weil hierbei Wissen über andere Konzepte verloren gehen kann.

Im Beispiel: Wird der „letzte“ Mitarbeiter einer Abteilung gelöscht, so verschwindet auch das Wissen über die Abteilung.

# Informelle Entwurfskriterien für Relationenschemata

## (2) Vermeidung von (Update-)Anomalien (Fortsetzung)

| MitarbeiterAbteilung |               |         |       |            |            |
|----------------------|---------------|---------|-------|------------|------------|
| Name                 | <u>PersNr</u> | Wohnort | AbtNr | AbtName    | ChefPersNr |
| Smith                | 1234          | Weimar  | 5     | Forschung  | 3334       |
| Wong                 | 3334          | Köln    | 5     | Forschung  | 3334       |
| Zelaya               | 9998          | Erfurt  | 4     | Verwaltung | 9876       |
| Wallace              | 9876          | Berlin  | 4     | Verwaltung | 9876       |

Modifizierungs- bzw. Modify-Anomalie:

Die Änderung eines Attributes muss in allen Tupeln, die den gleichen Attributwert haben, nachvollzogen werden. Ansonsten wird die Datenbank inkonsistent.

Im Beispiel: Änderung des Abteilungsnamens oder der Chefpersonalnummer.

## Bemerkungen:

- ❑ Bzgl. der Konsistenzerhaltung müssen Modifizierungsanomalien nicht kritisch sein, weil durch die Verwendung von Triggern, Stored-Procedures oder durch ein einziges SQL-Statement alle Korrekturen erfassbar sind. Performanzeinbußen und das Problem des erhöhten Speicherbedarfs bleiben jedoch.
- ❑ Umgekehrt kann es aus Performanzgründen in bestimmten Situationen sinnvoll sein, Anomalien in Kauf zu nehmen und *mehrere* Entity-Typen innerhalb *einer* Relation zu modellieren.

Beispiel: Eine Anwendung hat extrem viele Anfragen, in denen immer wieder die gleichen Entity-Typen vorkommen. Stichwort: Denormalisierung

Leistungsfähige Datenbanksysteme erkennen solche Situationen und richten automatisch – ausgehend von Views – verbundene Relationen ein, um die Anzahl von Join-Operationen klein zu halten.

- ❑ In [Kemper/Eickler 2015] werden Modifizierungsanomalien als Update-Anomalien bezeichnet.



# Informelle Entwurfskriterien für Relationenschemata

## (3) Möglichst wenig Nullwerte [Nullwerte in SQL]

Nullwerte sind aus folgenden Gründen kritisch:

### 1. Speicherplatz

Attribute, die nur für sehr wenige Tupel einen Wert besitzen, sollten in einem eigenen Schema untergebracht werden.

### 2. Interpretation

Die Semantik von Nullwerten ist unklar. Mögliche Interpretationen:

- ❑ Wert unbekannt – aber könnte irgendwann bekannt sein.
- ❑ Wert bekannt – soll aber nicht gespeichert werden.
- ❑ Wert existiert nicht für das Attribut – und wird auch nie existieren.

### 3. Verrechnung

Nullwerte verhindern die sinnvolle Verrechnung von Attributen bei der Verwendung von Aggregatfunktionen wie `count`, `max`, `etc.`

# Informelle Entwurfskriterien für Relationenschemata

## (4) Verhinderung der Erzeugung falscher Tupel

| MitarbeiterProjekt |                  |         |       |                 |            |
|--------------------|------------------|---------|-------|-----------------|------------|
| <u>PersNr</u>      | <u>ProjektNr</u> | Stunden | Name  | ProjektName     | ProjektOrt |
| 1234               | 1                | 32.5    | Smith | Produkt-X       | Bellaire   |
| 1234               | 2                | 7.5     | Smith | Produkt-Y       | Sugarland  |
| 3334               | 2                | 10.0    | Wong  | Produkt-Y       | Sugarland  |
| 3334               | 3                | 10.0    | Wong  | Produkt-Z       | Houston    |
| 3334               | 10               | 10.0    | Wong  | Vernetzung      | Stafford   |
| 3334               | 20               | 10.0    | Wong  | Neuorganisation | Houston    |

Nach entsprechender Projektion:

| MitarbeiterEinsatzOrte |            |
|------------------------|------------|
| Name                   | ProjektOrt |
| Smith                  | Bellaire   |
| Smith                  | Sugarland  |
| Wong                   | Sugarland  |
| Wong                   | Houston    |
| Wong                   | Stafford   |

| MitarbeiterProjekt2 |                  |         |                 |            |
|---------------------|------------------|---------|-----------------|------------|
| <u>PersNr</u>       | <u>ProjektNr</u> | Stunden | ProjektName     | ProjektOrt |
| 1234                | 1                | 32.5    | Produkt-X       | Bellaire   |
| 1234                | 2                | 7.5     | Produkt-Y       | Sugarland  |
| 3334                | 2                | 10.0    | Produkt-Y       | Sugarland  |
| 3334                | 3                | 10.0    | Produkt-Z       | Houston    |
| 3334                | 10               | 10.0    | Vernetzung      | Stafford   |
| 3334                | 20               | 10.0    | Neuorganisation | Houston    |

# Informelle Entwurfskriterien für Relationenschemata

## (4) Verhinderung der Erzeugung falscher Tupel (Fortsetzung)

... und anschließendem Join:

| MitarbeiterEinsatzOrte |            | ⋈ | MitarbeiterProjekt2 |           |         |                 |            |
|------------------------|------------|---|---------------------|-----------|---------|-----------------|------------|
| Name                   | ProjektOrt |   | PersNr              | ProjektNr | Stunden | ProjektName     | ProjektOrt |
| Smith                  | Bellaire   |   | 1234                | 1         | 32.5    | Produkt-X       | Bellaire   |
| Smith                  | Sugarland  |   | 1234                | 2         | 7.5     | Produkt-Y       | Sugarland  |
| Wong                   | Sugarland  |   | 3334                | 2         | 10.0    | Produkt-Y       | Sugarland  |
| Wong                   | Houston    |   | 3334                | 3         | 10.0    | Produkt-Z       | Houston    |
| Wong                   | Stafford   |   | 3334                | 10        | 10.0    | Vernetzung      | Stafford   |
|                        |            |   | 3334                | 20        | 10.0    | Neuorganisation | Houston    |

|   | PersNr | ProjektNr | Stunden | Name  | ProjektName     | ProjektOrt |   |
|---|--------|-----------|---------|-------|-----------------|------------|---|
|   | 1234   | 1         | 32.5    | Smith | Produkt-X       | Bellaire   |   |
|   | 1234   | 2         | 7.5     | Smith | Produkt-Y       | Sugarland  |   |
|   | 1234   | 2         | 7.5     | Wong  | Produkt-Y       | Sugarland  | * |
| = | 3334   | 2         | 10.0    | Smith | Produkt-Y       | Sugarland  | * |
|   | 3334   | 2         | 10.0    | Wong  | Produkt-Y       | Sugarland  |   |
|   | 3334   | 3         | 10.0    | Wong  | Produkt-Z       | Houston    |   |
|   | 3334   | 10        | 10.0    | Wong  | Vernetzung      | Stafford   |   |
|   | 3334   | 20        | 10.0    | Wong  | Neuorganisation | Houston    |   |

## Bemerkungen:

- ❑ Die Zerlegung der Relation „MitarbeiterProjekt“ in „MitarbeiterEinsatzOrte“ und „MitarbeiterProjekt2“ ist nicht sinnvoll, da mittels des natürlichen Verbundes die Originalrelation nicht mehr hergestellt werden kann: es entstehen falsche (*spurious*) Tupel, hier mit  $\gg\star\ll$  gekennzeichnet.
- ❑ Im Beispiel ist das Join-Attribut „ProjektOrt“. Wäre das Join-Attribut in mindestens einer der beiden Relationen ein Schlüssel, entstünden keine falschen Tupel. Warum nicht?

## VI. Entwurfstheorie relationaler Datenbanken

- ❑ Informelle Entwurfskriterien für Relationenschemata
- ❑ Funktionale Abhängigkeiten
- ❑ Normalformen
- ❑ Dekompositionseigenschaften von Relationen
- ❑ Relationale Dekomposition
- ❑ Relationale Synthese
- ❑ Mehrwertige Abhängigkeiten

# Funktionale Abhängigkeiten

## Definition 1 (funktionale Abhängigkeit, FD)

Gegeben sei ein Relationenschema  $\mathcal{R}$ . Weiterhin seien  $\alpha, \beta$  Mengen von Attributen mit  $\alpha \subseteq \mathcal{R}$  und  $\beta \subseteq \mathcal{R}$ . Eine funktionale Abhängigkeit (*Functional Dependency*, FD)

$$\alpha \rightarrow \beta$$

liegt vor, wenn die möglichen gültigen Ausprägungen  $r(\mathcal{R})$  von  $\mathcal{R}$  folgende Bedingung erfüllen:

$$\forall t_1, t_2 \in r : t_1(\alpha) = t_2(\alpha) \text{ impliziert } t_1(\beta) = t_2(\beta)$$

# Funktionale Abhängigkeiten

## Definition 1 (funktionale Abhängigkeit, FD)

Gegeben sei ein Relationenschema  $\mathcal{R}$ . Weiterhin seien  $\alpha, \beta$  Mengen von Attributen mit  $\alpha \subseteq \mathcal{R}$  und  $\beta \subseteq \mathcal{R}$ . Eine funktionale Abhängigkeit (*Functional Dependency*, FD)

$$\alpha \rightarrow \beta$$

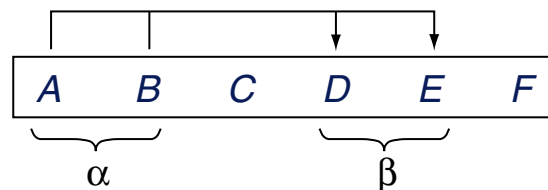
liegt vor, wenn die möglichen gültigen Ausprägungen  $r(\mathcal{R})$  von  $\mathcal{R}$  folgende Bedingung erfüllen:

$$\forall t_1, t_2 \in r : t_1(\alpha) = t_2(\alpha) \text{ impliziert } t_1(\beta) = t_2(\beta)$$

Sprechweisen:

- die  $\alpha$ -Werte bestimmen die  $\beta$ -Werte funktional
- die  $\beta$ -Werte sind funktional abhängig von den  $\alpha$ -Werten
- Relation  $r$  erfüllt die funktionale Abhängigkeit  $\alpha \rightarrow \beta$
- Relation  $r$  genügt der funktionalen Abhängigkeit  $\alpha \rightarrow \beta$

Grafische Darstellung:



## Bemerkungen:

- Funktionale Abhängigkeiten stellen semantische Konsistenzbedingungen dar, die sich aus einem *sachlogischen Verständnis* der Anwendung ergeben – aber nicht unbedingt aus einer aktuellen Relationenausprägung.

Somit sind funktionale Abhängigkeiten Eigenschaften von Relationenschemata und können nicht automatisch von der Ausprägung eines Schemas (= Relation) abgeleitet werden: Die Konsistenz von Ausprägung und Schema ist notwendig, aber nicht hinreichend für eine funktionale Abhängigkeit.

- Syntaxkonventionen in der Datenbankliteratur:
  - große lateinische Buchstaben,  $A, B, C, \dots$ , bezeichnen Attribute
  - kleine griechische Buchstaben,  $\alpha, \beta, \gamma, \dots$ , bezeichnen Attributmengen
  - $AB \rightarrow C$  steht für  $\{A, B\} \rightarrow \{C\}$
  - $\alpha\beta$  steht für  $\alpha \cup \beta$
- Wiederholung. Für eine Teilmenge der Attribute  $\alpha \subseteq \mathcal{R}$  eines Relationenschemas bezeichne  $t(\alpha)$  (bzw.  $t|_{\alpha}$ ) die Einschränkung eines Tupels  $t \in r(\mathcal{R})$  auf die Menge  $\alpha$ , also ein Teiltupel über  $\mathcal{R}$ .



# Funktionale Abhängigkeiten

Beispiel:

|       | $\mathcal{R}$ |       |       |       |
|-------|---------------|-------|-------|-------|
|       | $A$           | $B$   | $C$   | $D$   |
| $t_1$ | $a_4$         | $b_2$ | $c_4$ | $d_3$ |
| $t_2$ | $a_1$         | $b_1$ | $c_1$ | $d_1$ |
| $t_3$ | $a_1$         | $b_1$ | $c_1$ | $d_2$ |
| $t_4$ | $a_2$         | $b_2$ | $c_3$ | $d_2$ |
| $t_5$ | $a_3$         | $b_2$ | $c_4$ | $d_3$ |

Beobachtung:

Die funktionale Abhängigkeit  $\alpha \rightarrow \beta$  ist in einer Relation  $r$  erfüllt, falls für jede mögliche Ausprägung  $w$  von  $t(\alpha)$  gilt, dass die Projektion der zugehörigen Tupel auf  $\beta$  nur *ein* Element enthält. Formal:  $|\pi_\beta(\sigma_{\alpha=w}(r))| = 1$

# Funktionale Abhängigkeiten

Beispiel:

|       | $\mathcal{R}$ |       |       |       |
|-------|---------------|-------|-------|-------|
|       | $A$           | $B$   | $C$   | $D$   |
| $t_1$ | $a_4$         | $b_2$ | $c_4$ | $d_3$ |
| $t_2$ | $a_1$         | $b_1$ | $c_1$ | $d_1$ |
| $t_3$ | $a_1$         | $b_1$ | $c_1$ | $d_2$ |
| $t_4$ | $a_2$         | $b_2$ | $c_3$ | $d_2$ |
| $t_5$ | $a_3$         | $b_2$ | $c_4$ | $d_3$ |

Beobachtung:

Die funktionale Abhängigkeit  $\alpha \rightarrow \beta$  ist in einer Relation  $r$  erfüllt, falls für jede mögliche Ausprägung  $w$  von  $t(\alpha)$  gilt, dass die Projektion der zugehörigen Tupel auf  $\beta$  nur *ein* Element enthält. Formal:  $|\pi_\beta(\sigma_{\alpha=w}(r))| = 1$

Im Beispiel:

$$A \rightarrow C$$

# Funktionale Abhängigkeiten

Beispiel:

|       | $\mathcal{R}$ |       |       |       |
|-------|---------------|-------|-------|-------|
|       | $A$           | $B$   | $C$   | $D$   |
| $t_1$ | $a_4$         | $b_2$ | $c_4$ | $d_3$ |
| $t_2$ | $a_1$         | $b_1$ | $c_1$ | $d_1$ |
| $t_3$ | $a_1$         | $b_1$ | $c_1$ | $d_2$ |
| $t_4$ | $a_2$         | $b_2$ | $c_3$ | $d_2$ |
| $t_5$ | $a_3$         | $b_2$ | $c_4$ | $d_3$ |

Beobachtung:

Die funktionale Abhängigkeit  $\alpha \rightarrow \beta$  ist in einer Relation  $r$  erfüllt, falls für jede mögliche Ausprägung  $w$  von  $t(\alpha)$  gilt, dass die Projektion der zugehörigen Tupel auf  $\beta$  nur *ein* Element enthält. Formal:  $|\pi_\beta(\sigma_{\alpha=w}(r))| = 1$

Im Beispiel:

$$A \rightarrow C$$

$$t(A) = a_1 \quad (\text{analog für } a_2, a_3, a_4)$$

# Funktionale Abhängigkeiten

Beispiel:

|       | $\mathcal{R}$ |       |       |       |
|-------|---------------|-------|-------|-------|
|       | $A$           | $B$   | $C$   | $D$   |
| $t_1$ | $a_4$         | $b_2$ | $c_4$ | $d_3$ |
| $t_2$ | $a_1$         | $b_1$ | $c_1$ | $d_1$ |
| $t_3$ | $a_1$         | $b_1$ | $c_1$ | $d_2$ |
| $t_4$ | $a_2$         | $b_2$ | $c_3$ | $d_2$ |
| $t_5$ | $a_3$         | $b_2$ | $c_4$ | $d_3$ |

Beobachtung:

Die funktionale Abhängigkeit  $\alpha \rightarrow \beta$  ist in einer Relation  $r$  erfüllt, falls für jede mögliche Ausprägung  $w$  von  $t(\alpha)$  gilt, dass die Projektion der zugehörigen Tupel auf  $\beta$  nur *ein* Element enthält. Formal:  $|\pi_\beta(\sigma_{\alpha=w}(r))| = 1$

Im Beispiel:

$$A \rightarrow C$$

$$t(A) = a_1 \quad (\text{analog für } a_2, a_3, a_4)$$

$$|\pi_C(\sigma_{A=a_1}(r))| = |\{c_1\}| = 1$$

# Funktionale Abhängigkeiten

## Definition 2 (triviale funktionale Abhängigkeiten)

Funktionale Abhängigkeiten, die von jeder Relationenausprägung erfüllt sind, heißen triviale funktionale Abhängigkeiten.

# Funktionale Abhängigkeiten

## Definition 2 (triviale funktionale Abhängigkeiten)

Funktionale Abhängigkeiten, die von jeder Relationenausprägung erfüllt sind, heißen triviale funktionale Abhängigkeiten.

## Definition 3 (voll funktional abhängig)

Gegeben sei ein Relationenschema  $\mathcal{R}$  mit  $\alpha \subseteq \mathcal{R}$  und  $\beta \subseteq \mathcal{R}$ .  $\beta$  ist voll funktional abhängig von  $\alpha$ , falls gilt:

1.  $\alpha \rightarrow \beta$
2.  $\forall A \in \alpha : (\alpha - \{A\}) \not\rightarrow \beta$ ,  $\alpha$  ist also nicht verkleinerbar.

# Funktionale Abhängigkeiten

## Definition 2 (triviale funktionale Abhängigkeiten)

Funktionale Abhängigkeiten, die von jeder Relationenausprägung erfüllt sind, heißen triviale funktionale Abhängigkeiten.

## Definition 3 (voll funktional abhängig)

Gegeben sei ein Relationenschema  $\mathcal{R}$  mit  $\alpha \subseteq \mathcal{R}$  und  $\beta \subseteq \mathcal{R}$ .  $\beta$  ist voll funktional abhängig von  $\alpha$ , falls gilt:

1.  $\alpha \rightarrow \beta$
2.  $\forall A \in \alpha : (\alpha - \{A\}) \not\rightarrow \beta$ ,  $\alpha$  ist also nicht verkleinerbar.

## Definition 4 (Superschlüssel, Schlüsselkandidat, Primattribute, Primärschlüssel)

Gegeben sei ein Relationenschema  $\mathcal{R}$  mit  $\alpha \subseteq \mathcal{R}$ .

- $\alpha$  ist ein Superschlüssel, falls  $\alpha \rightarrow \mathcal{R}$  gilt.
- $\alpha$  ist ein Schlüssel bzw. Schlüsselkandidat von  $\mathcal{R}$ , falls  $\mathcal{R}$  voll funktional abhängig von  $\alpha$  ist. Die Attribute eines Schlüssels heißen Primattribute.
- Ein Primärschlüssel ist ein ausgezeichneteter Schlüsselkandidat.

## Bemerkungen:

- ❑ Man kann zeigen, dass nur funktionale Abhängigkeiten der Art  $\alpha \rightarrow \beta$  mit  $\beta \subseteq \alpha$  trivial sind.
- ❑ Gilt  $\alpha \rightarrow \beta$  und  $\exists A \in \alpha : (\alpha - \{A\}) \rightarrow \beta$ , so nennt man  $\beta$  auch *partiell abhängig* von  $\alpha$ .
- ❑ Das Konzept der vollen funktionalen Abhängigkeit dient dazu, Schlüssel von Superschlüsseln abzugrenzen.
- ❑ Die Menge aller Attribute einer Relation  $\mathcal{R}$  bildet einen Superschlüssel:  $\mathcal{R} \rightarrow \mathcal{R}$ .
- ❑ Die Auszeichnung eines Primärschlüssels ist hilfreich, um bei mehreren Fremdschlüsselverweisen immer denselben Schlüssel in der referenzierten Relation zu verwenden.



# Funktionale Abhängigkeiten

## Ableitung funktionaler Abhängigkeiten

Beispiel:

| $\mathcal{R}$ |       |       |
|---------------|-------|-------|
| $A$           | $B$   | $C$   |
| $a_1$         | $b_1$ | $c_1$ |
| $a_2$         | $b_1$ | $c_1$ |
| $a_3$         | $b_2$ | $c_1$ |
| $a_4$         | $b_1$ | $c_1$ |

Beobachtung:

Genügt  $\mathcal{R}$  den funktionalen Abhängigkeiten  $A \rightarrow B$  und  $B \rightarrow C$ , so genügt  $\mathcal{R}$  auch der funktionalen Abhängigkeit  $A \rightarrow C$ . Von  $\{A \rightarrow B, B \rightarrow C\}$  ist  $A \rightarrow C$  ableitbar.

# Funktionale Abhängigkeiten

## Ableitung funktionaler Abhängigkeiten

Beispiel:

| $\mathcal{R}$ |       |       |
|---------------|-------|-------|
| $A$           | $B$   | $C$   |
| $a_1$         | $b_1$ | $c_1$ |
| $a_2$         | $b_1$ | $c_1$ |
| $a_3$         | $b_2$ | $c_1$ |
| $a_4$         | $b_1$ | $c_1$ |

Beobachtung:

Genügt  $\mathcal{R}$  den funktionalen Abhängigkeiten  $A \rightarrow B$  und  $B \rightarrow C$ , so genügt  $\mathcal{R}$  auch der funktionalen Abhängigkeit  $A \rightarrow C$ . Von  $\{A \rightarrow B, B \rightarrow C\}$  ist  $A \rightarrow C$  ableitbar.

### Definition 5 (Hülle einer Menge von FDs)

Sei  $F$  eine Menge funktionaler Abhängigkeiten. Dann bezeichnet  $F^+$  die Hülle von  $F$ . Die Hülle  $F^+$  enthält die Menge  $F$  sowie alle funktionalen Abhängigkeiten, die auf der Basis von  $F$  ableitbar sind.

# Funktionale Abhängigkeiten

## Ableitung funktionaler Abhängigkeiten (Fortsetzung)

Sei  $\mathcal{R}$  ein Relationenschema und seien  $\alpha, \beta, \gamma$  und  $\delta$  Teilmengen von  $\mathcal{R}$ .

Armstrong-Inferenzregeln [Armstrong 1974]:

1. Reflexivität. Falls  $\beta \subseteq \alpha$ , dann gilt  $\alpha \rightarrow \beta$ . Insbesondere gilt  $\alpha \rightarrow \alpha$ .
2. *Verstärkung*. Falls  $\alpha \rightarrow \beta$ , dann gilt  $\alpha\gamma \rightarrow \beta\gamma$ .
3. *Transitivität*. Falls  $\alpha \rightarrow \beta$  und  $\beta \rightarrow \gamma$ , dann gilt  $\alpha \rightarrow \gamma$ .

# Funktionale Abhängigkeiten

## Ableitung funktionaler Abhängigkeiten (Fortsetzung)

Sei  $\mathcal{R}$  ein Relationenschema und seien  $\alpha, \beta, \gamma$  und  $\delta$  Teilmengen von  $\mathcal{R}$ .

Armstrong-Inferenzregeln [Armstrong 1974]:

1. Reflexivität. Falls  $\beta \subseteq \alpha$ , dann gilt  $\alpha \rightarrow \beta$ . Insbesondere gilt  $\alpha \rightarrow \alpha$ .
2. *Verstärkung*. Falls  $\alpha \rightarrow \beta$ , dann gilt  $\alpha\gamma \rightarrow \beta\gamma$ .
3. *Transitivität*. Falls  $\alpha \rightarrow \beta$  und  $\beta \rightarrow \gamma$ , dann gilt  $\alpha \rightarrow \gamma$ .

Zusätzliche Ableitungsregeln:

4. *Vereinigung*. Falls  $\alpha \rightarrow \beta$  und  $\alpha \rightarrow \gamma$ , dann gilt  $\alpha \rightarrow \beta\gamma$ .
5. *Dekomposition*. Falls  $\alpha \rightarrow \beta\gamma$ , dann gilt  $\alpha \rightarrow \beta$  und  $\alpha \rightarrow \gamma$ .
6. *Pseudotransitivität*. Falls  $\alpha \rightarrow \beta$  und  $\beta\gamma \rightarrow \delta$ , dann gilt  $\alpha\gamma \rightarrow \delta$ .

## Bemerkungen:

- ❑ Konvention zur Syntax (Wiederholung):  $\alpha\beta$  steht für  $\alpha \cup \beta$ . Weil es sich um Mengen handelt, ist keine Reihenfolge zwischen den Attributen festgelegt:  $\alpha\beta = \beta\alpha$ .
- ❑ In der Literatur werden die drei Armstrong-Inferenzregeln auch als Armstrong-Axiome bezeichnet. Der Begriff „Axiom“ ist unglücklich verwendet: Axiome sind voraussetzungslose Forderungen, die als wahr angenommen werden (und die zusammen eine Theorie bilden). Tatsächlich entspricht jede Menge von FDs einer Theorie: unter der Annahme ihrer Wahrheit lassen sich hieraus weitere wahre FDs ableiten.
- ❑ Die drei Armstrong-Inferenzregeln sind korrekt (*sound*) und vollständig (*complete*).
  1. Korrektheit. Mit Hilfe der drei Armstrong-Inferenzregeln lassen sich aus einer Menge  $F$  von FDs nur solche FDs ableiten, die von jeder Relationenausprägung erfüllt sind, für die  $F$  erfüllt ist.
  2. Vollständigkeit. Mit Hilfe der drei Armstrong-Inferenzregeln lassen sich alle FDs ableiten, die durch  $F$  impliziert sind (= die auf der Basis von  $F$  ableitbar sind).

Auf den Beweis dieser Eigenschaften wird hier verzichtet.

- ❑ Aufgrund der Vollständigkeit der Armstrong-Inferenzregeln sind die zusätzlichen Ableitungsregeln nicht nötig; mit ihnen lassen sich Ableitungsprozesse jedoch einfacher gestalten.

# Funktionale Abhängigkeiten

## Ableitung funktionaler Abhängigkeiten (Fortsetzung)

Algorithm: AttributeClosure

Input:  $F$ . Menge funktionaler Abhängigkeiten.

$\alpha$ . Menge von Attributen.

Output:  $\alpha_F^+$ . Hülle der Attributmenge  $\alpha$  unter der Menge  $F$ .

AttributeClosure( $F, \alpha$ )

1.  $\alpha_F^+ = \alpha$
2. **REPEAT**
3.      $\alpha_{\text{tmp}}^+ = \alpha_F^+$
4.     **FOREACH**  $(\beta \rightarrow \gamma) \in F$  **DO**
5.         **IF**  $\beta \subseteq \alpha_F^+$  **THEN**  $\alpha_F^+ = \alpha_F^+ \cup \gamma$
6.     **ENDDO**
7. **UNTIL**  $(\alpha_F^+ = \alpha_{\text{tmp}}^+)$
8. **return**( $\alpha_F^+$ )

Beispiel:  $F = \{A \rightarrow B, B \rightarrow C\}$ ,  $\alpha = \{A\}$

## Bemerkungen:

- Die Bestimmung von  $F^+$ , also der Menge *aller* FDs, die aus  $F$  ableitbar sind, erfordert im Allgemeinen exponentiellen Aufwand in der Anzahl  $k$  der verschiedenen Attribute in  $F$ .
- Oft ist man jedoch nicht an  $F^+$  interessiert, sondern nur an der Menge derjenigen Attribute, die von einer gegebenen Attributmenge  $\alpha$  durch eine Menge  $F$  von FDs funktional bestimmt werden, hier als  $\alpha_F^+$  bezeichnet. Der Algorithmus *AttributeClosure* führt genau diese Berechnung durch.
- Mit Hilfe des Algorithmus *AttributeClosure*( $F, \kappa$ ) lässt sich überprüfen, ob  $\kappa$  einen Superschlüssel eines Schemas  $\mathcal{R}$  hinsichtlich der FDs  $F$  darstellt. Wie?

# Funktionale Abhängigkeiten

## Äquivalenz von Mengen funktionaler Abhängigkeiten

### Definition 6 (Überdeckung einer Menge von FDs)

Eine Menge funktionaler Abhängigkeiten  $F_1$  überdeckt eine Menge funktionaler Abhängigkeiten  $F_2$ , falls jede FD aus  $F_2$  Element in  $F_1^+$  ist.

### Definition 7 (Äquivalenz zweier Mengen von FDs)

Zwei Mengen  $F_1$  und  $F_2$  funktionaler Abhängigkeiten heißen äquivalent, in Zeichen  $F_1 \equiv F_2$ , genau dann wenn sie die gleichen Hüllen besitzen, also falls  $F_1^+ = F_2^+$  gilt.



## Bemerkungen:

- Falls  $F_1$  die Menge  $F_2$  überdeckt, so kann jede FD in  $F_2$  auch mittels  $F_1$  abgeleitet werden.
- $F_1$  ist äquivalent zu  $F_2$  genau dann, falls  $F_1$  die Menge  $F_2$  überdeckt und falls  $F_2$  die Menge  $F_1$  überdeckt.

# Funktionale Abhängigkeiten

## Minimale und kanonische Überdeckungen von FDs

Konsistenzprüfungen bei Datenbankmodifikationen oder beim Datenbankentwurf erfordern den Vergleich von zwei Mengen funktionaler Abhängigkeiten  $F_1$  und  $F_2$ .

→ Hierfür interessiert die kleinstmögliche, zu  $F$  äquivalente Menge an FDs.

# Funktionale Abhängigkeiten

## Minimale und kanonische Überdeckungen von FDs

Konsistenzprüfungen bei Datenbankmodifikationen oder beim Datenbankentwurf erfordern den Vergleich von zwei Mengen funktionaler Abhängigkeiten  $F_1$  und  $F_2$ .

→ Hierfür interessiert die kleinstmögliche, zu  $F$  äquivalente Menge an FDs.

### Definition 8 (kanonische Überdeckung [Kemper/Eickler 2015])

Zu einer gegebenen Menge  $F$  von FDs nennt man  $F_c$  eine kanonische Überdeckung, falls folgende Eigenschaften erfüllt sind:

1.  $F_c \equiv F$  bzw.  $F_c^+ = F^+$  ( $F_c$  und  $F$  sind äquivalent)

2. Alle FDs  $\alpha \rightarrow \beta$  in  $F_c$  sind linksreduziert

(a)  $\forall A \in \alpha : (F_c - \{\alpha \rightarrow \beta\}) \cup \{(\alpha - \{A\}) \rightarrow \beta\} \neq F_c$

# Funktionale Abhängigkeiten

## Minimale und kanonische Überdeckungen von FDs

Konsistenzprüfungen bei Datenbankmodifikationen oder beim Datenbankentwurf erfordern den Vergleich von zwei Mengen funktionaler Abhängigkeiten  $F_1$  und  $F_2$ .

→ Hierfür interessiert die kleinstmögliche, zu  $F$  äquivalente Menge an FDs.

### Definition 8 (kanonische Überdeckung [Kemper/Eickler 2015])

Zu einer gegebenen Menge  $F$  von FDs nennt man  $F_c$  eine kanonische Überdeckung, falls folgende Eigenschaften erfüllt sind:

1.  $F_c \equiv F$  bzw.  $F_c^+ = F^+$  ( $F_c$  und  $F$  sind äquivalent)
2. Alle FDs  $\alpha \rightarrow \beta$  in  $F_c$  sind linksreduziert und rechtsreduziert.
  - (a)  $\forall A \in \alpha : (F_c - \{\alpha \rightarrow \beta\}) \cup \{(\alpha - \{A\}) \rightarrow \beta\} \neq F_c$
  - (b)  $\forall B \in \beta : (F_c - \{\alpha \rightarrow \beta\}) \cup \{ \alpha \rightarrow (\beta - \{B\}) \} \neq F_c$

# Funktionale Abhängigkeiten

## Minimale und kanonische Überdeckungen von FDs

Konsistenzprüfungen bei Datenbankmodifikationen oder beim Datenbankentwurf erfordern den Vergleich von zwei Mengen funktionaler Abhängigkeiten  $F_1$  und  $F_2$ .

→ Hierfür interessiert die kleinstmögliche, zu  $F$  äquivalente Menge an FDs.

### Definition 8 (kanonische Überdeckung [Kemper/Eickler 2015])

Zu einer gegebenen Menge  $F$  von FDs nennt man  $F_c$  eine kanonische Überdeckung, falls folgende Eigenschaften erfüllt sind:

1.  $F_c \equiv F$  bzw.  $F_c^+ = F^+$  ( $F_c$  und  $F$  sind äquivalent)
2. Alle FDs  $\alpha \rightarrow \beta$  in  $F_c$  sind linksreduziert und rechtsreduziert.
  - (a)  $\forall A \in \alpha : (F_c - \{\alpha \rightarrow \beta\}) \cup \{(\alpha - \{A\}) \rightarrow \beta\} \neq F_c$
  - (b)  $\forall B \in \beta : (F_c - \{\alpha \rightarrow \beta\}) \cup \{ \alpha \rightarrow (\beta - \{B\}) \} \neq F_c$
3. In  $F_c$  gibt es keine zwei FDs, die die gleiche linke Seite besitzen.

## Bemerkungen:

- Das Konzept der kanonischen Überdeckung macht zwei Mengen von FDs,  $F_1, F_2$ , mit  $F_1 \neq F_2$ , leicht vergleichbar: Haben  $F_1$  und  $F_2$  die gleiche kanonische Überdeckung  $F_c$ , so sind sie äquivalent, in Zeichen  $F_1 \equiv F_2$ .

# Funktionale Abhängigkeiten

## Minimale und kanonische Überdeckungen von FDs (Fortsetzung)

Aus einer gegebenen Menge  $F$  von FDs lässt sich eine kanonische Überdeckung wie folgt bestimmen.

1. **Linksreduktion** für jede FD  $(\alpha \rightarrow \beta) \in F$ .  
Falls  $\beta \subseteq \text{AttributeClosure}(F, \alpha - \{A\})$ ,  
dann ersetze  $\alpha \rightarrow \beta$  durch  $(\alpha - \{A\}) \rightarrow \beta$ .

# Funktionale Abhängigkeiten

## Minimale und kanonische Überdeckungen von FDs (Fortsetzung)

Aus einer gegebenen Menge  $F$  von FDs lässt sich eine kanonische Überdeckung wie folgt bestimmen.

1. Linksreduktion für jede FD  $(\alpha \rightarrow \beta) \in F$ .  
Falls  $\beta \subseteq \text{AttributeClosure}(F, \alpha - \{A\})$ ,  
dann ersetze  $\alpha \rightarrow \beta$  durch  $(\alpha - \{A\}) \rightarrow \beta$ .
2. Rechtsreduktion für jede FD  $(\alpha \rightarrow \beta) \in F$ .  
Falls  $B \in \text{AttributeClosure}((F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - \{B\})\}, \alpha)$ ,  
dann ersetze  $\alpha \rightarrow \beta$  durch  $\alpha \rightarrow (\beta - \{B\})$ .



# Funktionale Abhängigkeiten

## Minimale und kanonische Überdeckungen von FDs (Fortsetzung)

Aus einer gegebenen Menge  $F$  von FDs lässt sich eine kanonische Überdeckung wie folgt bestimmen.

1. Linksreduktion für jede FD  $(\alpha \rightarrow \beta) \in F$ .  
Falls  $\beta \subseteq \text{AttributeClosure}(F, \alpha - \{A\})$ ,  
dann ersetze  $\alpha \rightarrow \beta$  durch  $(\alpha - \{A\}) \rightarrow \beta$ .
2. Rechtsreduktion für jede FD  $(\alpha \rightarrow \beta) \in F$ .  
Falls  $B \in \text{AttributeClosure}((F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - \{B\})\}, \alpha)$ ,  
dann ersetze  $\alpha \rightarrow \beta$  durch  $\alpha \rightarrow (\beta - \{B\})$ .
3. Löschen aller FDs der Form  $(\alpha \rightarrow \emptyset) \in F$ .

# Funktionale Abhängigkeiten

## Minimale und kanonische Überdeckungen von FDs (Fortsetzung)

Aus einer gegebenen Menge  $F$  von FDs lässt sich eine kanonische Überdeckung wie folgt bestimmen.

1. Linksreduktion für jede FD  $(\alpha \rightarrow \beta) \in F$ .  
Falls  $\beta \subseteq \text{AttributeClosure}(F, \alpha - \{A\})$ ,  
dann ersetze  $\alpha \rightarrow \beta$  durch  $(\alpha - \{A\}) \rightarrow \beta$ .
2. Rechtsreduktion für jede FD  $(\alpha \rightarrow \beta) \in F$ .  
Falls  $B \in \text{AttributeClosure}((F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - \{B\})\}, \alpha)$ ,  
dann ersetze  $\alpha \rightarrow \beta$  durch  $\alpha \rightarrow (\beta - \{B\})$ .
3. Löschen aller FDs der Form  $(\alpha \rightarrow \emptyset) \in F$ .
4. Anwenden der Vereinigungsregel.  
Ersetze alle Regeln der Form  $\alpha \rightarrow \beta_1, \dots, \alpha \rightarrow \beta_n$  durch  $\alpha \rightarrow \beta_1, \dots, \beta_n$ .

## Bemerkungen:

- Eine Linksreduktion entspricht einer Verstärkung einer funktionalen Abhängigkeit; eine Rechtsreduktion entspricht einer Abschwächung einer funktionalen Abhängigkeit.  
Für alle funktionalen Abhängigkeiten werden beide Reduktionen testweise angewandt, um diejenigen Reduktionen zu erkennen (und anzuwenden), die die Menge der ableitbaren funktionalen Abhängigkeiten nicht verändern.
- In einer kanonischen Überdeckung  $F_c$  einer Menge  $F$  von funktionalen Abhängigkeiten existieren keine funktionalen Abhängigkeiten mehr, die reduziert werden können, ohne dass die Äquivalenz  $F_c \equiv F$  zerstört wird.
- In bestimmten Fällen (z.B. bei zirkulären oder redundanten Abhängigkeiten) kann es für eine Menge  $F$  von funktionalen Abhängigkeiten verschiedene kanonische Überdeckungen  $F_c$  geben. Die Eindeutigkeit kann jedoch durch eine Festlegung der Bearbeitungsreihenfolge der funktionalen Abhängigkeiten im Algorithmus einfach hergestellt werden.
- In [Elmasri/Navathe 2010] wird das Konzept der *minimalen Überdeckung* definiert und zu dessen Bestimmung ein vergleichbarer Algorithmus vorgestellt. Wesentlicher Unterschied zu der Definition von [Kemper/Eickler 2015] ist, dass statt eindeutiger linker Seiten aller funktionalen Abhängigkeiten gefordert wird, dass die rechten Seiten aller funktionalen Abhängigkeiten aus nur einem Attribut bestehen.

## VI. Entwurfstheorie relationaler Datenbanken

- ❑ Informelle Entwurfskriterien für Relationenschemata
- ❑ Funktionale Abhängigkeiten
- ❑ Normalformen
- ❑ Dekompositionseigenschaften von Relationen
- ❑ Relationale Dekomposition
- ❑ Relationale Synthese
- ❑ Mehrwertige Abhängigkeiten

# Normalformen

Die Normalisierung ist ein formales Werkzeug zur Analyse von Relationenschemata hinsichtlich ihrer funktionalen Abhängigkeiten und Primärschlüssel. Ziele sind:

1. die Minimierung von Redundanzen
2. die Minimierung von Einfüge-, Lösch- und Modifizierungsanomalien

Folgende Normalformen werden zunächst betrachtet:

- ❑ erste Normalform, 1NF
- ❑ zweite Normalform, 2NF
- ❑ dritte Normalform, 3NF
- ❑ Boyce-Codd-Normalform, BCNF

## Bemerkungen:

- ❑ Die Normalform eines Schemas  $\mathcal{R}$  bezieht sich immer auf die höchste Normalform, die  $\mathcal{R}$  einhält.
- ❑ Die Herstellung von einer (hohen) Normalform ist kein hinreichendes Kriterium für ein gutes Datenbankdesign.

# Normalformen

## Erste Normalform

Ein Relationenschema ist in der ersten Normalform (1NF), wenn der Wertebereich (die Domäne) eines Attributes nur atomare Werte enthält. Mengenwertige oder relationenwertige Attributdomänen sind nicht zulässig.

Beispielrelation, die nicht in erster Normalform ist:

| Abteilung  |              |            |                                |
|------------|--------------|------------|--------------------------------|
| AbtName    | <u>AbtNr</u> | ChefPersNr | AbtOrte                        |
| Forschung  | 5            | 3334       | {Bellaire, Sugarland, Houston} |
| Verwaltung | 4            | 9876       | {Stafford}                     |
| Stab       | 1            | 8886       | {Houston}                      |

# Normalformen

## Erste Normalform

Ein Relationenschema ist in der ersten Normalform (1NF), wenn der Wertebereich (die Domäne) eines Attributes nur atomare Werte enthält. Mengenwertige oder relationenwertige Attributdomänen sind nicht zulässig.

Beispielrelation, die nicht in erster Normalform ist:

| Abteilung  |              |            |                                |
|------------|--------------|------------|--------------------------------|
| AbtName    | <u>AbtNr</u> | ChefPersNr | AbtOrte                        |
| Forschung  | 5            | 3334       | {Bellaire, Sugarland, Houston} |
| Verwaltung | 4            | 9876       | {Stafford}                     |
| Stab       | 1            | 8886       | {Houston}                      |

~>

| Abteilung  |              |            |               |
|------------|--------------|------------|---------------|
| AbtName    | <u>AbtNr</u> | ChefPersNr | <u>AbtOrt</u> |
| Forschung  | 5            | 3334       | Bellaire      |
| Forschung  | 5            | 3334       | Sugarland     |
| Forschung  | 5            | 3334       | Houston       |
| Verwaltung | 4            | 9876       | Stafford      |
| Stab       | 1            | 8886       | Houston       |

[entspricht Möglichkeit 1]



# Normalformen

## Erste Normalform (Fortsetzung)

Möglichkeiten zur Herstellung der ersten Normalform:

1. Einführung zusätzlicher Tupel für jeden Wert eines mehrwertigen Attributes.  
Nachteil: es wird Redundanz in die ursprüngliche Relation eingebracht
2. Entfernung des mehrwertigen Attributes aus der ursprünglichen Relation und Erzeugung eines weiteren Schemas zusammen mit dem Primärschlüssel der ursprünglichen Relation.

Im Beispiel:

| Abteilung |              |            |
|-----------|--------------|------------|
| AbtName   | <u>AbtNr</u> | ChefPersNr |

| Standort     |               |
|--------------|---------------|
| <u>AbtNr</u> | <u>AbtOrt</u> |

# Normalformen

## Erste Normalform (Fortsetzung)

Möglichkeiten zur Herstellung der ersten Normalform:

1. Einführung zusätzlicher Tupel für jeden Wert eines mehrwertigen Attributes.  
Nachteil: es wird Redundanz in die ursprüngliche Relation eingebracht
2. Entfernung des mehrwertigen Attributes aus der ursprünglichen Relation und Erzeugung eines weiteren Schemas zusammen mit dem Primärschlüssel der ursprünglichen Relation.

Im Beispiel:

| Abteilung |              |            |
|-----------|--------------|------------|
| AbtName   | <u>AbtNr</u> | ChefPersNr |

| Standort     |               |
|--------------|---------------|
| <u>AbtNr</u> | <u>AbtOrt</u> |

3. Falls die Maximalzahl der Werte des mehrwertigen Attributes bekannt sind, Einführung entsprechend vieler atomarer Attribute. Nachteile:
  - ❑ es können viele Nullwerte eingeführt werden
  - ❑ fragwürdige Semantik einer Rangordnung unter diesen Attributen
  - ❑ die Formulierung von Anfragen gestaltet sich schwieriger

## Bemerkungen:

- Das NF<sup>2</sup>-Datenmodell erlaubt Mengen und geschachtelte Relationen. Dabei steht NF<sup>2</sup> für *Non-First (NF) Normal Form (NF)*.

# Normalformen

## Zweite Normalform

Sei  $\mathcal{R}$  ein Relationenschema in 1NF mit den Schlüsselkandidaten  $\kappa_1, \dots, \kappa_k$ .

Formulierung 1:

$\mathcal{R}$  ist in zweiter Normalform (2NF), falls jedes Nicht-Schlüsselattribut voll funktional abhängig von jedem Schlüssel  $\kappa_1, \dots, \kappa_k$  ist.

$A$  ist ein Nicht-Schlüsselattribut, falls  $A \in (\mathcal{R} - (\kappa_1 \cup \dots \cup \kappa_k))$ .

# Normalformen

## Zweite Normalform

Sei  $\mathcal{R}$  ein Relationenschema in 1NF mit den Schlüsselkandidaten  $\kappa_1, \dots, \kappa_k$ .

Formulierung 1:

$\mathcal{R}$  ist in zweiter Normalform (2NF), falls jedes Nicht-Schlüsselattribut voll funktional abhängig von jedem Schlüssel  $\kappa_1, \dots, \kappa_k$  ist.

$A$  ist ein Nicht-Schlüsselattribut, falls  $A \in (\mathcal{R} - (\kappa_1 \cup \dots \cup \kappa_k))$ .

Formulierung 2:

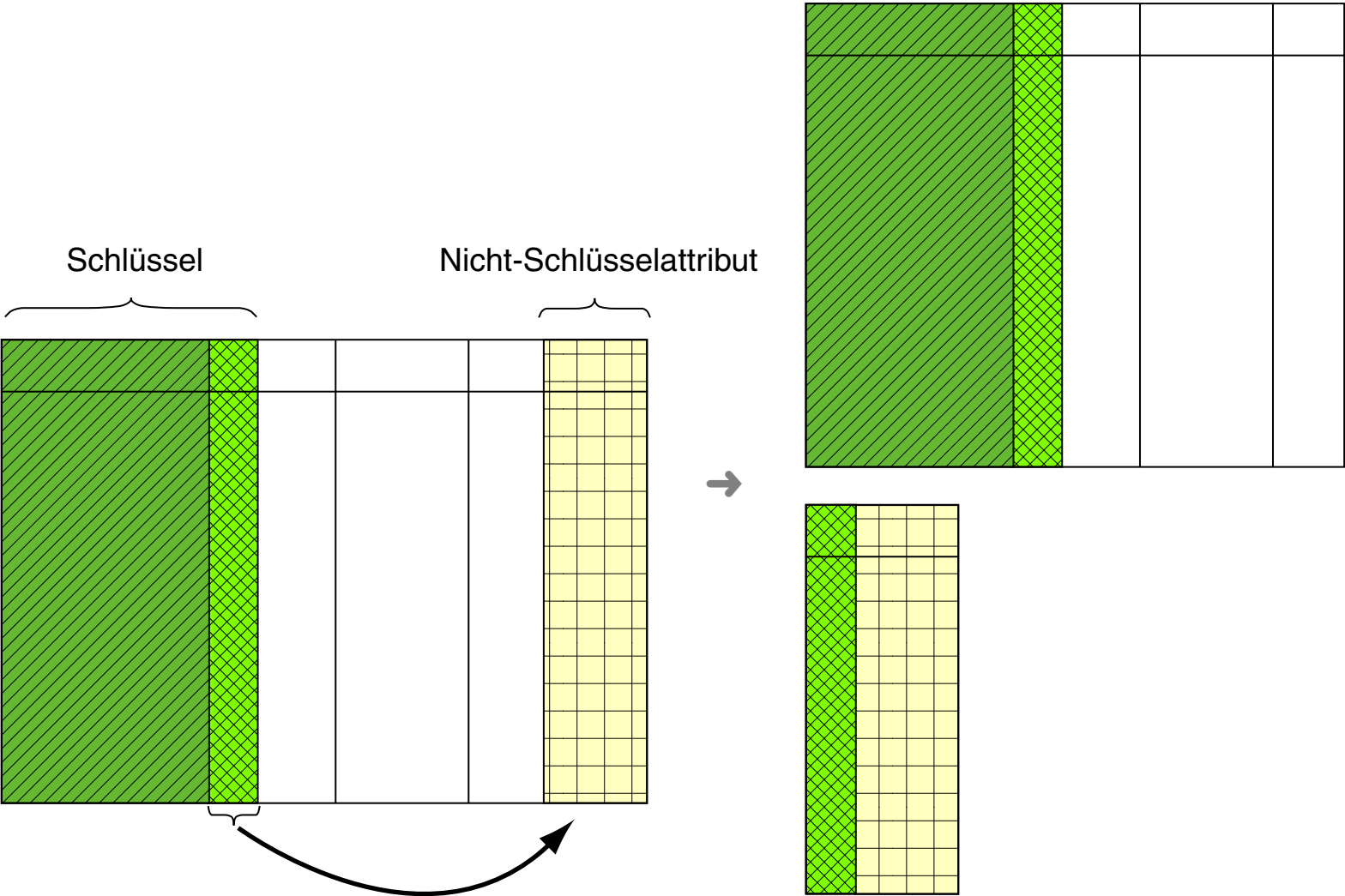
$\mathcal{R}$  ist in zweiter Normalform, falls keine partiellen Abhängigkeiten zwischen einem Schlüssel und Nicht-Schlüsselattributen existieren.

Formal:

$$\neg \underbrace{(\exists \kappa \exists \alpha \exists A : \kappa \in \{\kappa_1, \dots, \kappa_k\} \wedge \alpha \subset \kappa \wedge A \in (\mathcal{R} - (\kappa_1 \cup \dots \cup \kappa_k)) : \alpha \rightarrow A)}_{\text{partielle Abhängigkeit}}$$

# Normalformen

## Zweite Normalform (Fortsetzung)



# Normalformen

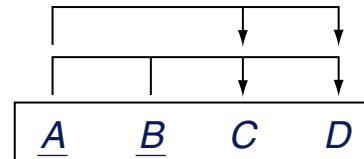
## Zweite Normalform (Fortsetzung)

Für ein Relationenschema  $\mathcal{R}$  kann die zweite Normalform durch Zerlegung hergestellt werden: Elimination der rechten Seite der partiellen Abhängigkeit in  $\mathcal{R}$ . Erstellung eines weiteren Schemas bestehend aus den Attributen der partiellen Abhängigkeit.

Beispielrelation, die nicht in zweiter Normalform ist:

Studentenbelegung

| <u>MatrNr</u> | <u>VorlNr</u> | Name         | Semester |
|---------------|---------------|--------------|----------|
| 26120         | 5001          | Fichte       | 10       |
| 27550         | 5001          | Schopenhauer | 6        |
| 27550         | 4052          | Schopenhauer | 6        |
| 28106         | 5041          | Carnap       | 3        |
| 28106         | 5052          | Carnap       | 3        |
| 28106         | 5216          | Carnap       | 3        |
| 28106         | 5259          | Carnap       | 3        |



# Normalformen

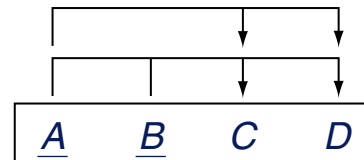
## Zweite Normalform (Fortsetzung)

Für ein Relationenschema  $\mathcal{R}$  kann die zweite Normalform durch Zerlegung hergestellt werden: Elimination der rechten Seite der partiellen Abhängigkeit in  $\mathcal{R}$ . Erstellung eines weiteren Schemas bestehend aus den Attributen der partiellen Abhängigkeit.

Beispielrelation, die nicht in zweiter Normalform ist:

Studentenbelegung

| <u>MatrNr</u> | <u>VorlNr</u> | Name         | Semester |
|---------------|---------------|--------------|----------|
| 26120         | 5001          | Fichte       | 10       |
| 27550         | 5001          | Schopenhauer | 6        |
| 27550         | 4052          | Schopenhauer | 6        |
| 28106         | 5041          | Carnap       | 3        |
| 28106         | 5052          | Carnap       | 3        |
| 28106         | 5216          | Carnap       | 3        |
| 28106         | 5259          | Carnap       | 3        |



hoeren

MatrNr VorlNr

Studenten

MatrNr Name Semester



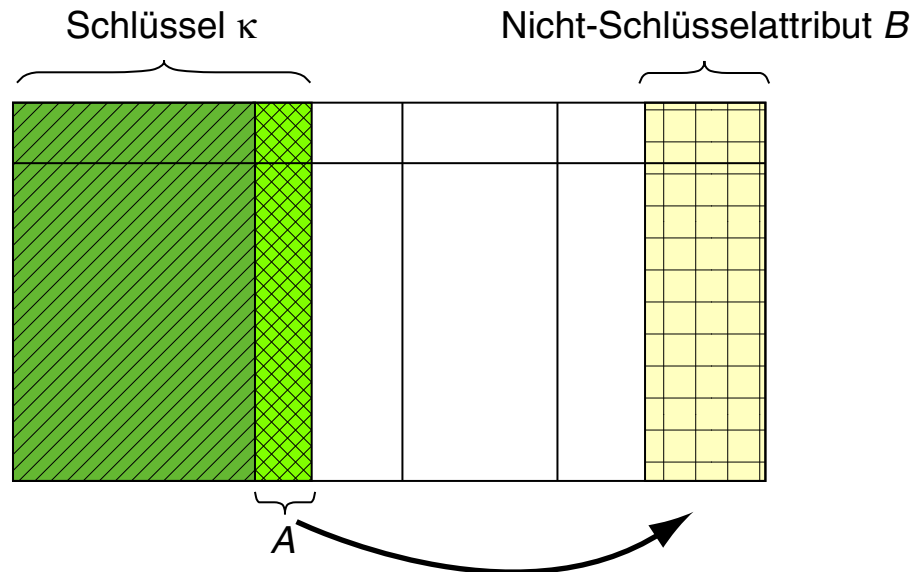
## Bemerkungen:

- Im Beispiel gibt es neben den Schlüsselabhängigkeiten die FD  $\text{MatrNr} \rightarrow \{\text{Name}, \text{Semester}\}$ . Folgende Anomalien resultieren:
  1. Einfügeanomalie: Was macht man mit Studierenden, die noch keine Vorlesung hören?
  2. Modifizierungsanomalie: Wenn Carnap ins vierte Semester kommt, müssen vier Tupel geändert werden.
  3. Löschanomalie: Was passiert, wenn Fichte ihre einzige Vorlesung absagt?  
Beachte, dass bei „VorNr“ kein Nullwert eingetragen werden kann, weil es sich hierbei um ein Schlüsselattribut handelt. Also muss das Tupel – und mit ihm die Information über Fichte – aus der Relation gelöscht werden.
- 2NF nicht mit NF<sup>2</sup> verwechseln!

# Normalformen

## Zweite Normalform (Fortsetzung)

Hängt ein Attribut  $B$  von einem Attribut  $A$  ab, wobei  $A$  Element des Schlüssels  $\kappa$  ist, so ist Redundanz unvermeidbar. Argumentation:

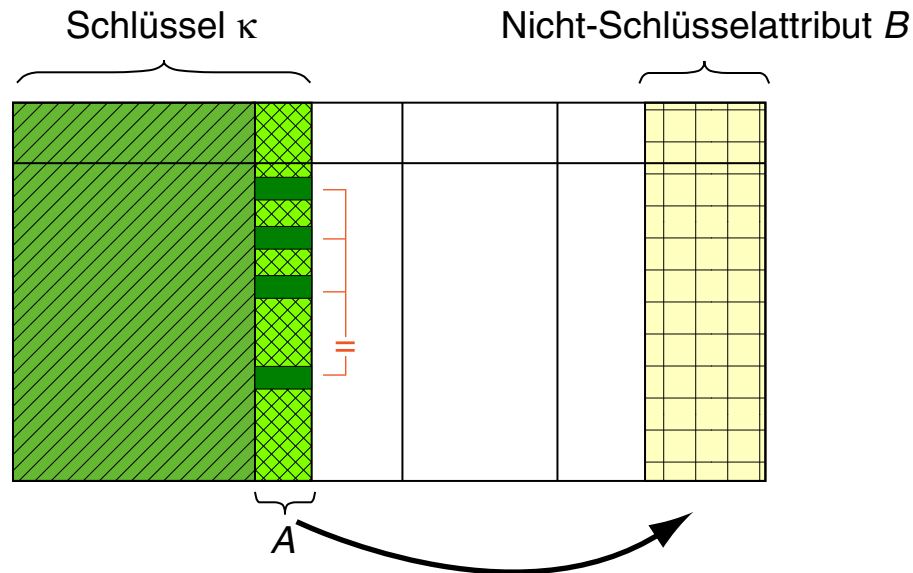


# Normalformen

## Zweite Normalform (Fortsetzung)

Hängt ein Attribut  $B$  von einem Attribut  $A$  ab, wobei  $A$  Element des Schlüssels  $\kappa$  ist, so ist Redundanz unvermeidbar. Argumentation:

- $A$  kann in mehreren Tupeln dieselbe Ausprägung besitzen, da  $A$  sonst Schlüssel wäre; laut Voraussetzung ist aber  $\kappa, \kappa \supset \{A\}$ , Schlüssel.

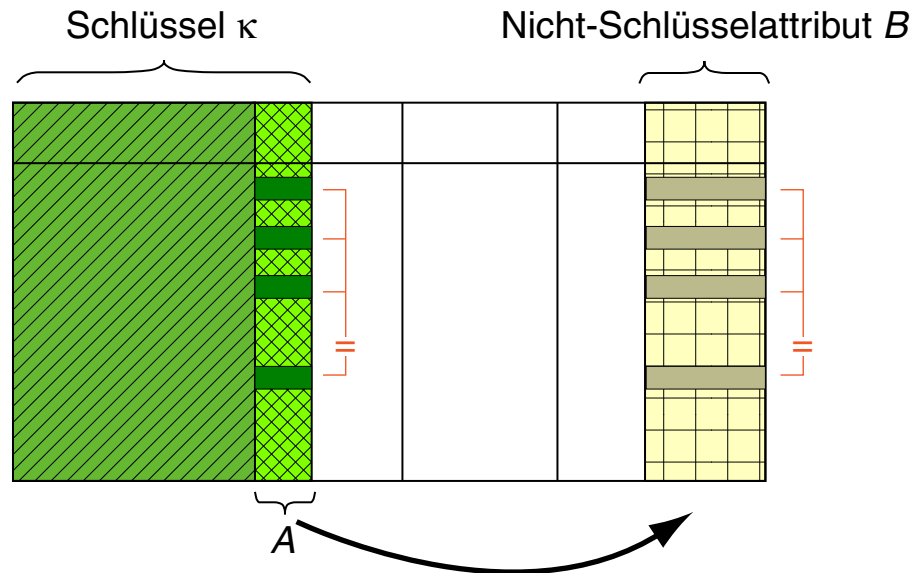


# Normalformen

## Zweite Normalform (Fortsetzung)

Hängt ein Attribut  $B$  von einem Attribut  $A$  ab, wobei  $A$  Element des Schlüssels  $\kappa$  ist, so ist Redundanz unvermeidbar. Argumentation:

- $A$  kann in mehreren Tupeln dieselbe Ausprägung besitzen, da  $A$  sonst Schlüssel wäre; laut Voraussetzung ist aber  $\kappa, \kappa \supset \{A\}$ , Schlüssel.
- Weiterhin gilt  $A \rightarrow B$ , und somit können in der Relation  $r$  Tupel  $t_1, t_2$  mit  $t_1(A, B) = t_2(A, B)$  existieren.



# Normalformen

## Dritte Normalform [\[Boyce-Codd-Normalform\]](#)

### Formulierung 1:

Ein Relationenschema  $\mathcal{R}$  in 1NF ist in dritter Normalform (3NF), wenn für jede [nicht-triviale](#) funktionale Abhängigkeit  $\alpha \rightarrow A$ , die für  $\mathcal{R}$  gilt, mindestens eine der folgenden Bedingungen erfüllt ist:

1.  $\alpha$  ist [Superschlüssel](#) von  $\mathcal{R}$
2.  $A$  ist ein Primattribut von  $\mathcal{R}$

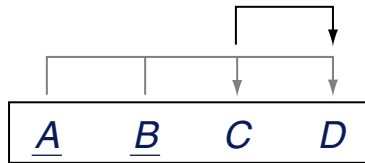
# Normalformen

## Dritte Normalform (Fortsetzung)

Formulierung 2:

$\mathcal{R}$  ist in dritter Normalform, falls  $\mathcal{R}$  in zweiter Normalform ist und keine transitiven Abhängigkeiten zwischen einem Schlüssel und Nicht-Schlüsselattributen existieren.

Eine funktionale Abhängigkeit  $\alpha \rightarrow \beta$  ist eine transitive Abhängigkeit, falls es eine Menge  $\gamma$  von Attributen gibt, die **weder Schlüssel noch Teilmenge eines Schlüssels** in  $\mathcal{R}$  ist, und für die  $\alpha \rightarrow \gamma$  und  $\gamma \rightarrow \beta$  gilt:



$$\alpha = \{A, B\}, \gamma = \{C\}, \beta = \{D\}$$

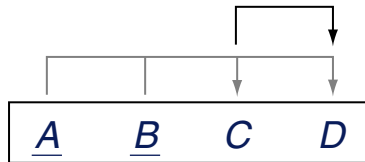
# Normalformen

## Dritte Normalform (Fortsetzung)

### Formulierung 2:

$\mathcal{R}$  ist in dritter Normalform, falls  $\mathcal{R}$  in zweiter Normalform ist und keine transitiven Abhängigkeiten zwischen einem Schlüssel und Nicht-Schlüsselattributen existieren.

Eine funktionale Abhängigkeit  $\alpha \rightarrow \beta$  ist eine transitive Abhängigkeit, falls es eine Menge  $\gamma$  von Attributen gibt, die **weder Schlüssel noch Teilmenge eines Schlüssels** in  $\mathcal{R}$  ist, und für die  $\alpha \rightarrow \gamma$  und  $\gamma \rightarrow \beta$  gilt:



$$\alpha = \{A, B\}, \gamma = \{C\}, \beta = \{D\}$$

### Formulierung 3:

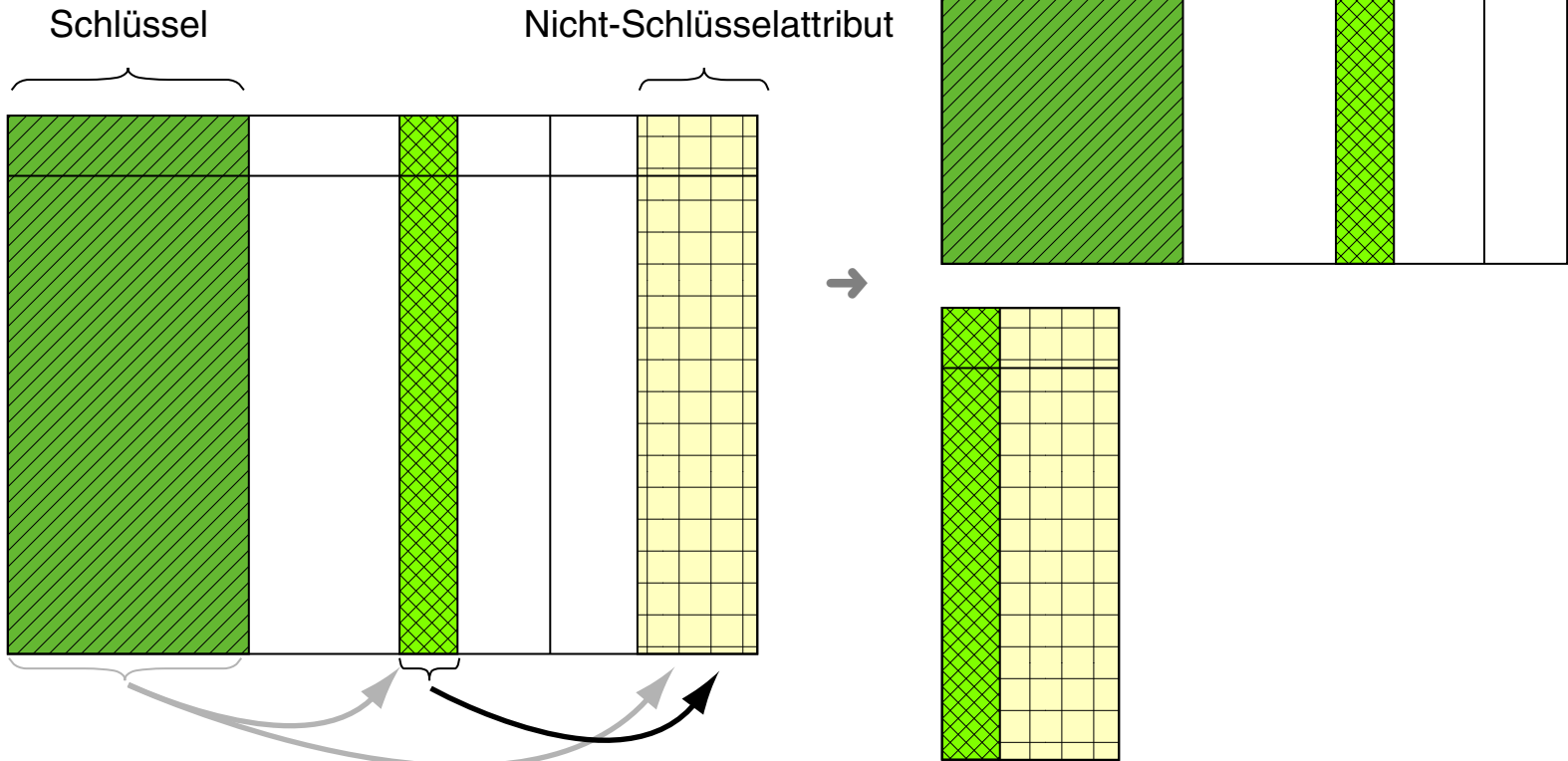
$\mathcal{R}$  ist in dritter Normalform, falls  $\mathcal{R}$  in zweiter Normalform ist und kein Nicht-Schlüsselattribut funktional von einer Menge anderer Nicht-Schlüsselattribute abhängt.

Vergleiche Formulierung 2:  $\beta$  hängt funktional von  $\gamma$  ab.

# Normalformen

Dritte Normalform (Fortsetzung)

[Boyce-Codd-Normalform]





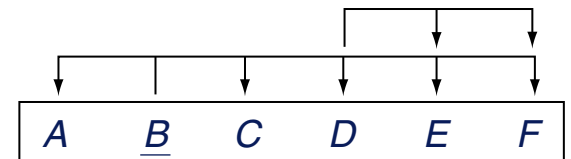
# Normalformen

## Dritte Normalform (Fortsetzung)

Für ein Relationenschema  $\mathcal{R}$  kann die dritte Normalform durch Zerlegung hergestellt werden: Elimination der transitiv abhängigen Attributmengende in  $\mathcal{R}$ . Erstellung eines weiteren Schemas bestehend aus der zwischengeschalteten Attributmengende und der transitiv abhängigen Attributmengende.

Beispielrelation, die nicht in dritter Normalform ist:

| MitarbeiterAbteilung |               |         |       |            |            |
|----------------------|---------------|---------|-------|------------|------------|
| Name                 | <u>PersNr</u> | Wohnort | AbtNr | AbtName    | ChefPersNr |
| Smith                | 1234          | Weimar  | 5     | Forschung  | 3334       |
| Wong                 | 3334          | Köln    | 5     | Forschung  | 3334       |
| Zelaya               | 9998          | Erfurt  | 4     | Verwaltung | 9876       |
| Wallace              | 9876          | Berlin  | 4     | Verwaltung | 9876       |



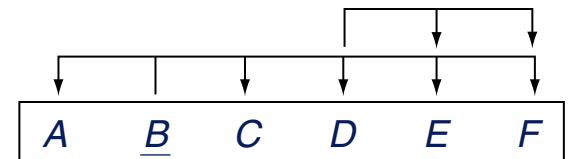
# Normalformen

## Dritte Normalform (Fortsetzung)

Für ein Relationenschema  $\mathcal{R}$  kann die dritte Normalform durch Zerlegung hergestellt werden: Elimination der transitiv abhängigen Attributmengung in  $\mathcal{R}$ . Erstellung eines weiteren Schemas bestehend aus der zwischengeschalteten Attributmengung und der transitiv abhängigen Attributmengung.

Beispielrelation, die nicht in dritter Normalform ist:

| MitarbeiterAbteilung |               |         |       |            |            |
|----------------------|---------------|---------|-------|------------|------------|
| Name                 | <u>PersNr</u> | Wohnort | AbtNr | AbtName    | ChefPersNr |
| Smith                | 1234          | Weimar  | 5     | Forschung  | 3334       |
| Wong                 | 3334          | Köln    | 5     | Forschung  | 3334       |
| Zelaya               | 9998          | Erfurt  | 4     | Verwaltung | 9876       |
| Wallace              | 9876          | Berlin  | 4     | Verwaltung | 9876       |



↪

| Mitarbeiter |               |         |       |
|-------------|---------------|---------|-------|
| Name        | <u>PersNr</u> | Wohnort | AbtNr |

| Abteilung |              |            |
|-----------|--------------|------------|
| AbtName   | <u>AbtNr</u> | ChefPersNr |

## Bemerkungen:

- Im Beispiel gibt es neben den Schlüsselabhängigkeiten  $\text{PersNr} \rightarrow \mathcal{R}$  die FD  $\text{AbtNr} \rightarrow \{\text{AbtName}, \text{ChefPersNr}\}$ . „AbtName“ und „ChefPersNr“ hängen deshalb transitiv von „PersNr“ ab – mit der Folge, dass der Abteilungsname und die Chefpersonalnummer redundant gespeichert sind.
- Beachte, dass die Formulierungen 2 und 3 der dritten Normalform die Überprüfung erfordern, ob das Relationenschema  $\mathcal{R}$  die Bedingungen der zweiten Normalform erfüllt.
- Die dritte Normalform wird verletzt, wenn ein Nicht-Schlüsselattribut einen Fakt zu einer Attributmenge darstellt, die keinen Schlüssel bildet. [Kent 1983]
  - In der Formulierung 2 der dritten Normalform stellt  $\beta$  einen Fakt zu  $\gamma$  dar.
  - Im Beispiel stellen „Abteilungsname“ und „Chefpersonalnummer“ Fakten zur „Abteilungsnummer“ dar.

# Normalformen

## Boyce-Codd-Normalform [dritte Normalform: 1, 2]

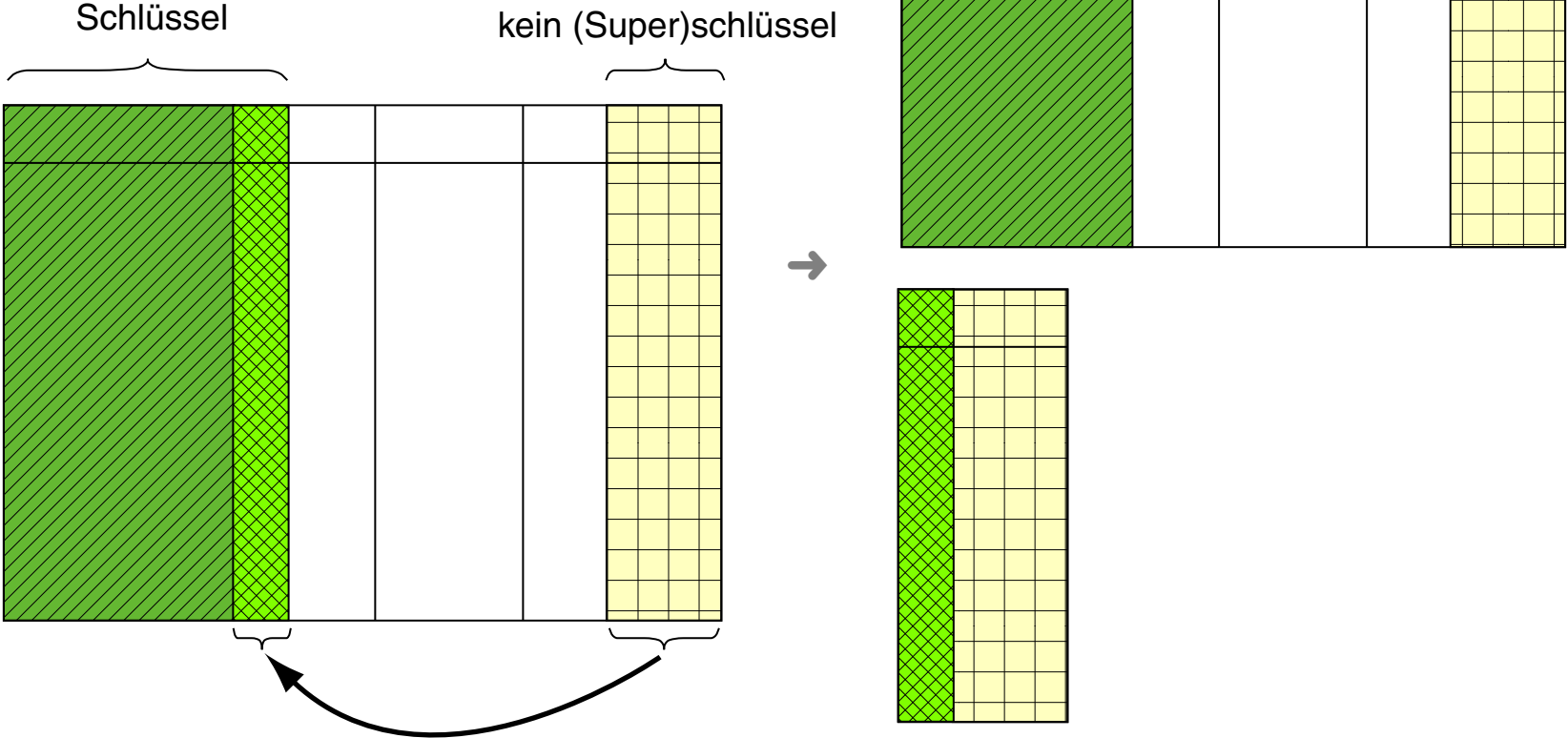
Ein Relationenschema  $\mathcal{R}$  ist in Boyce-Codd-Normalform (BCNF), wenn für jede nicht-triviale funktionale Abhängigkeit  $\alpha \rightarrow A$ , die für  $\mathcal{R}$  gilt, folgende Bedingung erfüllt ist:

- $\alpha$  ist Superschlüssel von  $\mathcal{R}$ .

# Normalformen

Boyce-Codd-Normalform (Fortsetzung)

[dritte Normalform]



# Normalformen

## Boyce-Codd-Normalform (Fortsetzung)

Für ein Relationenschema  $\mathcal{R}$  kann die Boyce-Codd-Normalform durch Zerlegung hergestellt werden: Elimination der abhängigen Attributmenge in  $\mathcal{R}$ . Erstellung eines weiteren Schemas bestehend aus den Attributen der funktionalen Abhängigkeit.

Beispielrelation, die nicht in BCNF ist:

| Grundstuecke        |               |               |                    |
|---------------------|---------------|---------------|--------------------|
| <u>SteuerNr</u> $A$ | Landkreis $B$ | GrundstNr $C$ | GrundstGroesse $D$ |

FDs:  $A \rightarrow BCD$        $A$  ist (Super)schlüssel  
 $BC \rightarrow AD$        $BC$  ist (Super)schlüssel  
 $D \rightarrow B$        $D$  ist kein Superschlüssel  $\leadsto D \rightarrow B$  unerwünschte FD

# Normalformen

## Boyce-Codd-Normalform (Fortsetzung)

Für ein Relationenschema  $\mathcal{R}$  kann die Boyce-Codd-Normalform durch Zerlegung hergestellt werden: Elimination der abhängigen Attributmenge in  $\mathcal{R}$ . Erstellung eines weiteren Schemas bestehend aus den Attributen der funktionalen Abhängigkeit.

Beispielrelation, die nicht in BCNF ist:

| Grundstuecke        |               |               |                    |
|---------------------|---------------|---------------|--------------------|
| <u>SteuerNr</u> $A$ | Landkreis $B$ | GrundstNr $C$ | GrundstGroesse $D$ |

FDs:  $A \rightarrow BCD$        $A$  ist (Super)schlüssel  
 $BC \rightarrow AD$        $BC$  ist (Super)schlüssel  
 $D \rightarrow B$        $D$  ist kein Superschlüssel  $\leadsto D \rightarrow B$  unerwünschte FD

$\rightsquigarrow$

| Grundstuecke1       |               |                    |
|---------------------|---------------|--------------------|
| <u>SteuerNr</u> $A$ | GrundstNr $C$ | GrundstGroesse $D$ |

| Grundstuecke2 |                           |
|---------------|---------------------------|
| Landkreis $B$ | <u>GrundstGroesse</u> $D$ |

FDs:  $A \rightarrow \cancel{BCD}$        $A$  ist (Super)schlüssel  
 $\cancel{B} \cancel{C} \not\rightarrow A \cancel{D}$       verlorene FD  
 $D \rightarrow B$        $D$  ist (Super)schlüssel

## Bemerkungen:

- Im Beispiel gibt es die beiden Schlüssel  $\{\text{SteuerNr}\}$  und  $\{\text{Landkreis, GrundstNr}\}$  und die FD  $\text{GrundstGroesse} \rightarrow \text{Landkreis}$ .

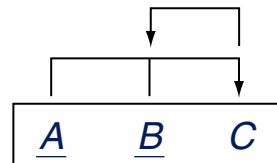
Die Relation „Grundstuecke“ ist nicht in BCNF: „GrundstGroesse“ ist kein Superschlüssel bzw. „Landkreis“ ist transitiv abhängig von „SteuerNr“ ( $A \rightarrow BCD \wedge D \rightarrow B$ ).

Bei der Zerlegung in die Schemata „Grundstuecke1“ und „Grundstuecke2“ geht die FD  $\{\text{Landkreis, GrundstNr}\} \rightarrow \{\text{SteuerNr, GrundstGroesse}\}$  ( $BC \rightarrow AD$ ) verloren.

- Die Definition der Boyce-Codd-Normalform ist eine Vereinfachung der Definition der dritten Normalform, stellt aber eine strengere Forderung dar. D.h., ein Relationenschema in BCNF ist gleichzeitig in 3NF, aber nicht jedes Relationenschema in 3NF ist auch in BCNF:

$$\text{BCNF} \Rightarrow \text{3NF}$$

- Stellt man die BCNF für ein Relationenschema her, das in der dritten Normalform aber nicht in BCNF ist, so gehen bei der Elimination der abhängigen Attributmenge zwangsläufig Abhängigkeiten verloren, weil die linke Seite von einer FD zerschnitten wird. Kleinstes Relationenschema, das in der dritten Normalform aber nicht in BCNF ist:



- BCNF in a nutshell: “Each field must represent a fact about the key, the whole key and nothing but the key.” [Kent 1983]



# Normalformen

## Minimalität

Normalformen vermeiden Redundanz *innerhalb* einer Relation  $\mathcal{R}$ .

Redundanz kann aber auch global in einem Datenbankschema  $\mathcal{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_p\}$  entstehen, obwohl alle Relationenschemata  $\mathcal{R}_i$  eine hohe Normalform erfüllen.

Beispiel:

$$(a) \mathcal{R}_1 = \{\{\underline{A}, B\}, \{\underline{B}, C\}\}$$

$$(b) \mathcal{R}_2 = \{\{\underline{A}, B\}, \{\underline{B}, C\}, \{\underline{A}, C\}\}$$

Die Relationen in  $\mathcal{R}_1$  und  $\mathcal{R}_2$  sind jeweils in BCNF. Die Redundanz in  $\mathcal{R}_2$  kann jedoch zu Inkonsistenzen durch Update-Anomalien führen.

# Normalformen

## Minimalität (Fortsetzung)

Die Forderung nach Minimalität bedeutet, dass von einer Menge von alternativen Datenbankschemata  $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_p$ , deren Relationen in der gewünschten Normalform sind, dasjenige Schema  $\mathcal{R}^*$  gewählt wird, für das gilt:

$$|\mathcal{R}^*| \leq |\mathcal{R}_i|, \quad i = 1, \dots, p$$