

Verfeinerung des relationalen Schemas

Ein schlechtes Schema

Filmliste

| Titel | Regisseur | Kino | Telefonnummer | Zeit |
|------------------------|------------------|-------------|----------------------|-------------|
| The Hobbit | Jackson | Cinema City | 441111 | 11:30 |
| The Lord of the Rings3 | Jackson | Cinema City | 441111 | 14:30 |
| Adventures of Tintin | Spielberg | Odeon | 442222 | 11:00 |
| The Lord of the Rings3 | Jackson | Odeon | 442222 | 14:00 |
| War Horse | Spielberg | Odeon | 442222 | 16:30 |

- Was könnte Primärschlüssel sein?

Einfüge-Anomalie

| Titel | Regisseur | Kino | Telefonnummer | Zeit |
|------------------------|-----------|-------------|---------------|-------|
| The Hobbit | Jackson | Cinema City | 441111 | 11:30 |
| The Lord of the Rings3 | Jackson | Cinema City | 441111 | 14:30 |
| Adventures of Tintin | Spielberg | Odeon | 442222 | 11:00 |
| The Lord of the Rings3 | Jackson | Odeon | 442222 | 14:00 |
| War Horse | Spielberg | Odeon | 442222 | 16:30 |

- Was ist das Problem mit dieser SQL-Anweisung?

```
INSERT INTO Filmliste(Titel,Regisseur)  
VALUES (`I.T.` , `Moore`)
```

Lösch-Anomalie

| Titel | Regisseur | Kino | Telefonnummer | Zeit |
|------------------------|-----------|-------------|---------------|-------|
| The Hobbit | Jackson | Cinema City | 441111 | 11:30 |
| The Lord of the Rings3 | Jackson | Cinema City | 441111 | 14:30 |
| Adventures of Tintin | Spielberg | Odeon | 442222 | 11:00 |
| The Lord of the Rings3 | Jackson | Odeon | 442222 | 14:00 |
| War Horse | Spielberg | Odeon | 442222 | 16:30 |

- Was ist das Problem mit dieser SQL-Anweisung?

```
DELETE FROM Filmliste  
WHERE Regisseur = `Jackson`
```

Update-Anomalie

| Titel | Regisseur | Kino | Telefonnummer | Zeit |
|------------------------|-----------|-------------|---------------|-------|
| The Hobbit | Jackson | Cinema City | 441111 | 11:30 |
| The Lord of the Rings3 | Jackson | Cinema City | 441111 | 14:30 |
| Adventures of Tintin | Spielberg | Odeon | 442222 | 11:00 |
| The Lord of the Rings3 | Jackson | Odeon | 442222 | 14:00 |
| War Horse | Spielberg | Odeon | 442222 | 16:30 |

- Was ist das Problem mit dieser SQL-Anweisung?

```
UPDATE Filmliste
```

```
SET Telefonnummer = 441133
```

```
WHERE Titel = `The Hobbit` and Kino = `Cinema City`
```

Probleme (Anomalien)

- Wegen schlechtes Schemadesign können folgende Probleme vorkommen:
 - **Einfüge-Anomalie:** Wir können kein neues Tupel einfügen, wenn das Kino und die Zeit für den Film nicht bekannt sind
 - Ursache: Vermischung zweier Entitätstypen
 - Wirkung: Es können nicht Daten nur einer neuen Entitätstypen angelegt werden
 - **Lösch-Anomalie:** Wenn wir alle Filme von Peter Jackson löschen, verlieren wir alle Informationen über das Kino „Cinema City“
 - Ursache: Vermischung zweier Entitätstypen
 - Wirkung: Beim Löschen der Informationen zu einer Entität können Informationen zu einer andere Entität ungewollt verloren gehen
 - **Update-Anomalie:** Die Änderung des Telefonnummers eines Kinos kann zu Inkonsistenzen führen
 - Ursache: Redundanz innerhalb der Relation
 - Wirkung: Eine Aktualisierung kann zu Inkonsistenzen führen, wenn die Änderung nicht in allen betroffenen Datensätze durchgeführt wird

Ziele des Datenbankentwurfs

- Vermeidung von Redundanzen und Anomalien:
 - Änderungen können bei Beachtung der Primär- und Fremdschlüssel keine Inkonsistenzen hervorrufen
 - Alle Informationen lassen sich unter Wahrung der Primärschlüssel- und Fremdschlüsselbedingungen einfügen
 - Informationen können einzeln wieder gelöscht werden, ohne die Primärschlüssel- oder Fremdschlüsselbedingungen zu verletzen
- Vermeidung von Informationsverlust
- Evtl. Einbeziehung von Effizienzüberlegungen
- **Vorgehen:** Prozess der Zerlegung eines gegebenen Datenbank-Schemas in ein äquivalentes Schema ohne Redundanz und Anomalien
(ein „schlechtes“ Schema in mehrere „gute“ Schemas zerlegen)

Zerlegung der Relation *Filmliste*

Filme

| Titel | Regisseur |
|------------------------|-----------|
| The Hobbit | Jackson |
| The Lord of the Rings3 | Jackson |
| Adventures of Tintin | Spielberg |
| War Horse | Spielberg |

Kinos

| Kino | Telefonnummer |
|-------------|---------------|
| Cinema City | 441111 |
| Odeon | 442222 |

Screens

| Kino | Zeit | Titel |
|-------------|-------|------------------------|
| Cinema City | 11:30 | The Hobbit |
| Cinema City | 14:30 | The Lord of the Rings3 |
| Odeon | 11:00 | Adventures of Tintin |
| Odeon | 14:00 | The Lord of the Rings3 |
| Odeon | 16:30 | War Horse |

Ziele des Datenbankentwurfs

- Fragestellung bei der Entwurfstheorie:
 - Wie kann die Güte eines Datenbankschemas beurteilt werden? (ist das Schema gut oder nicht?)
 - Wie sieht ein gutes konzeptuelles Schema der Datenbank aus? / Wie kann man ein schlechtes Schema in ein gutes Schema umwandeln?
- Erstellung „schöner“ Relationenschemata:
 - Normalisierung = Überführung in eine redundanzarme Standard-Form
 - Die Theorie der funktionalen Abhängigkeiten
 - Vorgehen: Prozess der Zerlegung eines gegebenen Datenbank-Schemas in ein äquivalentes Schema ohne Redundanz und Anomalien (ein „schlechtes“ Schema in mehrere „gute“ Schemas zerlegen)

Funktionale Abhängigkeiten

- Informelle Definition:
 - Ein Attribut oder eine Kombination von Attributen bestimmt die Werte eines anderen Attributs oder Attributkombination
- Notation: $A \rightarrow B$
- Sprechweisen:
 - A bestimmt B (funktional)
 - A identifiziert oder impliziert B
- Definition

Seien A und B Attributmengen der Relations R, t_1, t_2 Tupeln der Ausprägung $r(R)$ und $\pi_A(t_1)$ die Projektion des Tupels t_1 auf die Attributenmenge A.

B ist von A **funktional abhängig**, $A \rightarrow B$, wenn für alle möglichen Ausprägungen $r(R)$ (alle Instanzen der Relation) zu jedem Wert in A genau ein Wert in B gehört

$$A \rightarrow B \Leftrightarrow (\forall t_1, t_2 \in r(RS): \pi_A(t_1) = \pi_A(t_2) \Rightarrow \pi_B(t_1) = \pi_B(t_2))$$

Funktionale Abhängigkeiten - Beispiel

Filmliste

| Titel | Regisseur | Kino | Telefonnummer | Zeit |
|------------------------|-----------|-------------|---------------|-------|
| The Hobbit | Jackson | Cinema City | 441111 | 11:30 |
| The Lord of the Rings3 | Jackson | Cinema City | 441111 | 14:30 |
| Adventures of Tintin | Spielberg | Odeon | 442222 | 11:00 |
| The Lord of the Rings3 | Jackson | Odeon | 442222 | 14:00 |
| War Horse | Spielberg | Odeon | 442222 | 16:30 |

- Welche funktionalen Abhängigkeiten könnt ihr identifizieren?
 - Titel → Regisseur
 - Kino → Telefonnummer
 - Vielleicht Kino, Zeit → Titel (wenn das Kino nur ein Saal hat)

Funktionale Abhängigkeiten

- **Achtung.** Funktionale Abhängigkeiten sind abhängig von der Semantik des Schemas, nicht von der aktuellen Ausprägung der Relation!

Ausprägung der Relation Filme

| Titel | Regisseur | Komponist |
|---------------------|-----------|-----------|
| Schindler's List | Spielberg | Williams |
| Saving Private Ryan | Spielberg | Williams |
| North by Northwest | Hitchcock | Herrmann |
| Angela's Ashes | Parker | Williams |
| Vertigo | Hitchcock | Herrmann |

- Komponist → Regisseur?
- Regisseur → Komponist ?
- Obwohl diese funktionale Abhängigkeit auf diese Ausprägung gilt, können wir nicht sagen dass es für die ganze Relation gilt

Funktionale Abhängigkeiten

- Eine Abhängigkeit $A \rightarrow B$ ist **trivial**, wenn gilt $B \subseteq A$
- Eine funktionale Abhängigkeit $X \rightarrow Y$ heißt **voll**, wenn es keine echte Teilmengen $Z \subset X$ gibt, s.d. gilt $Z \rightarrow Y$
 - Alle Attribute in X sind für die funktionale Abhängigkeit notwendig, es darf keines weggelassen werden
 - Die Determinate X ist also irreduzibel
- Gibt es eine solche Teilmenge, dann heißt $X \rightarrow Y$ **partielle** Abhängigkeit

Funktionale Abhängigkeiten - Beispiel

- Lieferant (LiefName, LiefAdr, Ware, Preis)
- Welche funktionalen Abhängigkeiten könnt ihr identifizieren?
 - LiefName \rightarrow LiefAdr (ein Lieferantennamenname bestimmt eindeutig seine Adresse)
 - LiefName, Ware \rightarrow Preis (der Schlüssel Name, Ware bestimmt eindeutig den Preis)
 - LiefName, Ware \rightarrow Ware (trivial)
 - LiefName, Ware \rightarrow LiefAdr (partiell)
 - LiefName \rightarrow LiefName (trivial)

Berechnung funktionalen Abhängigkeiten

- Aus einer Menge F von FDs sind weitere FDs herleitbar
 - Eine **Hülle (closure)** F^+ von F ist die Menge aller funktionalen Abhängigkeiten, die aus den funktioanlen Abhängigkeiten in F ableitbar sind.
 - Es gibt Inferenzregeln, die Armstrong Axiome, zum Herleiten weiterer FDs
- Notation: F eine Menge von FDs, $A, B, C \subseteq R$
- **Armstrong Axiome:**
 - Reflexivität: Sei $B \subseteq A$. Dann gilt stets $A \rightarrow B$ (Sonderfall $A \rightarrow A$)
 - Verstärkung: Falls $A \rightarrow B$ gilt, dann gilt auch $A \cup C \rightarrow B \cup C$
 - Transitivität: Falls $A \rightarrow B$ und $B \rightarrow C$, dann gilt auch $A \rightarrow C$

Berechnung funktionalen Abhängigkeiten

- Armstrong Axiome sind:
 - Korrekt: abgeleitete Regeln sind für alle Relationen des Schemas gültig
 - Vollständig: Alle gültigen FDs in der Hülle F^+ sind mit Hilfe dieser Regeln ableitbar
- Erweiterung der Armstrong Axiome
 - Vereinigungsregel: Falls $A \rightarrow B$ und $A \rightarrow C$ gilt, dann gilt auch $A \rightarrow B \cup C$
 - Dekompositionsregel: Falls $A \rightarrow B \cup C$ gilt, dann gilt auch $A \rightarrow B$ und $A \rightarrow C$
 - Pseudotransitivität: Falls $A \rightarrow B$ und $B \cup C \rightarrow D$ gilt, dann gilt auch $A \cup C \rightarrow D$

Berechnung funktionalen Abhängigkeiten

- Beispiel

- Sei $R(A,B,C,D,E)$ und die folgenden FDs:

- $A \rightarrow C$

- $B \rightarrow C$

- $C \cup D \rightarrow E$

- Zu zeigen: $A \cup D \rightarrow E$

- Lösung

- $A \rightarrow C$ (gegeben)

- Verstärkung: $A \cup D \rightarrow C \cup D$

- $C \cup D \rightarrow E$ (gegeben)

- Transitivität: $A \cup D \rightarrow E$

Superschlüssel / Oberschlüssel

- Was machen wir mit FDs?
 - Man kann einen Schlüssel für R berechnen
- Superschlüssel
 - In der Relation R ist $A \subseteq R$ ein **Superschlüssel** falls gilt $A \rightarrow R$
 - D.h. A bestimmt alle anderen Attributwerte innerhalb der Relation R
- Achtung
 - Superschlüssel nicht notwendig minimal, $R \rightarrow R$ gilt immer

Superschlüssel vs. Kandidatenschlüssel

- Achtung
 - Mit Hilfe der vollen funktionalen Abhängigkeit können wir Kandidatenschlüssel von Superschlüssel abgrenzen
- Kandidatenschlüssel $K \subseteq R$:
 - $K \rightarrow R$ (Vollständigkeit), es gibt keinen $K' \subset K$ so dass $K' \rightarrow R$ (Minimalität)
- D.h. eine Attributenmenge K ist Kandidatenschlüssel falls gilt:
 - K ist Superschlüssel
 - Es gibt kein $K' \subset K$ so dass K' Superschlüssel ist
- Ein Attribut heißt **prim** in R , wenn es in einem Schlüssel (Kandidatenschlüssel) von R enthalten ist.

Superschlüssel - Beispiel

- *Filmliste (Titel, Regisseur, Kino, Telefonnummer, Zeit)*
- FDs:
 - *Kino, Zeit \rightarrow Titel*
 - *Kino \rightarrow Telefonnummer*
 - *Titel \rightarrow Regisseur*
- *{Kino, Zeit}* ist der einzige Schlüssel von *Filmliste*
- *Kino* und *Zeit* sind die einzige prime Attribute
- Jede Obermenge von *{Kino, Zeit}* in R ist ein Superschlüssel von *Filmliste*

Membership-Problem

- Fragestellung
 - Sei F eine Menge funktionaler Abhängigkeiten und $A \rightarrow B$ gegeben
 - Gilt $A \rightarrow B \in F^+$?
- Lösung
 - Explizite Berechnung von F^+ ist sehr aufwendig (kann exponentiell groß sein)
 - Effizienter: Berechnung der Hülle A^+ der Attributmenge A bzgl. der Menge F :
 - A^+ besteht aus allen Attributen, die von A funktional bestimmt werden
 - Falls $B \subseteq A^+$ gilt, dann gilt auch $A \rightarrow B \in F^+$

Algorithmus Hülle(F,A)

- Input: F, A
- Output: A^+

Erg := A;

WHILE (Änderungen an Erg) **DO**

FOREACH FD $(B \rightarrow C) \in F$ **DO**

 IF $B \subseteq \text{Erg}$ **THEN** Erg := Erg \cup C;

RETURN $A^+ = \text{Erg}$

- Anwendung: Überprüfung auf Eigenschaft als Schlüsselkandidat eines Attribtes A: Hülle (F, A) = R ?

Hülle - Beispiel

- FDs für Relation R:
 - $A \rightarrow C$
 - $B \rightarrow C$
 - $CD \rightarrow E$ (Abk. Von $C \cup D \rightarrow E$)
- Gilt: $AD \rightarrow E$? \Rightarrow zu berechnen: AD^+

| i | Erg | FD |
|---|------|--------------------|
| 0 | AD | |
| 1 | ACD | $A \rightarrow C$ |
| 2 | ACD | $B \rightarrow C$ |
| 3 | ACDE | $CD \rightarrow E$ |
| 4 | ACDE | $B \rightarrow C$ |

Zerlegung(Dekomposition) eines Relationenschema

- Um Anomalien zu beseitigen, wird das Schema RS einer Relation R in eine Vielzahl kleinerer Relationenschemata RS_1, \dots, RS_n zerlegt, so dass:
 - $RS_i \subseteq RS, 1 \leq i \leq n$
 - $RS = \bigcup_i RS_i$
- Die Relation R wird in den Relationen R_1, \dots, R_n zerlegt, wobei
 - $R_i = \pi_{RS_i}(R)$

Zerlegung - Beispiel

- *Filmliste (Titel, Regisseur, Kino, Telefonnummer, Zeit)*
- Zerlegung:
 - *(Kino, Zeit, Titel)*
 - *(Titel, Regisseur)*
 - *(Kino, Telefonnummer)*

Korrektheitskriterien

- Verlustlosigkeit
 - Die Ausprägungen $r(R)$ der Ursprungsrelation R müssen aus den Relationen R_1, \dots, R_n wieder rekonstruierbar sein
 - D.h., keine Informationen dürfen verloren gehen und keine zusätzliche Informationen dürfen auftauchen
- Abhängigkeitserhaltung
 - Die für R geltenden funktionalen Abhängigkeiten müssen auf die Relationen R_1, \dots, R_n übertragbar sein

Natürlicher Verbund (Natural Join)

- Das Kreuzprodukt wird gebildet, aus dem dann nur diejenigen Tupel selektiert werden, deren Attributwerte für gleichbenannte Attribute gleich sind
- Formell:

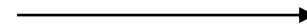
$$R \bowtie S = \prod_{A_1, \dots, A_r, R.B_1, \dots, R.B_k, C_1, \dots, C_n} (\sigma_{R.B_1=S.B_1 \wedge \dots \wedge R.B_k=S.B_k} (R \times S))$$

R

| B | C | A |
|---|---|---|
| b | c | a |
| b | c | d |
| b | F | b |
| a | D | c |

S

| B | C | D |
|---|---|---|
| b | c | d |
| b | c | e |
| a | d | b |



R ⋈ S

| B | C | A | D |
|---|---|---|---|
| b | c | a | d |
| b | c | a | e |
| b | c | d | d |
| b | c | d | e |
| a | d | c | b |

Verlustlose Zerlegung (Lossless-Join Decomposition)

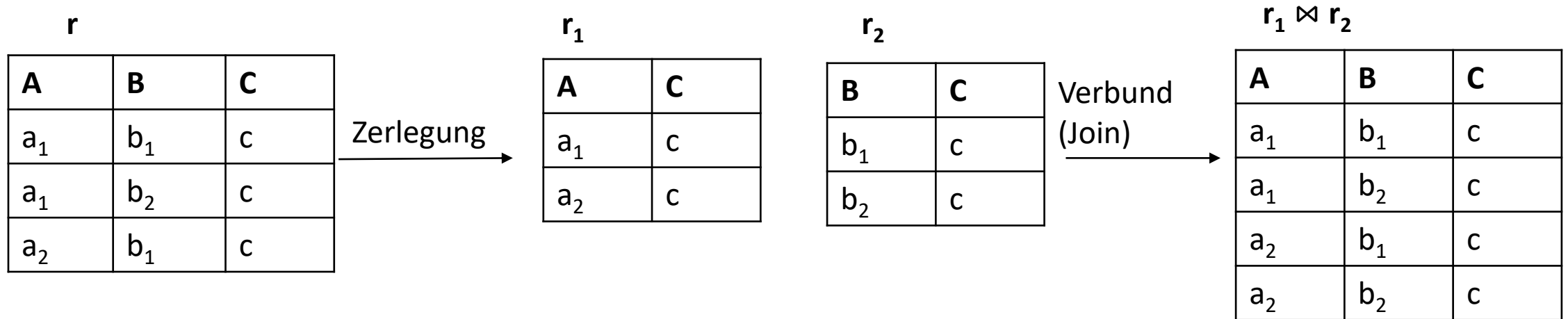
- Sei die Zerlegung der Relation R in R_1, \dots, R_n
- Es gilt immer: $r \subseteq \pi_{R_1}(r) \bowtie \dots \bowtie \pi_{R_n}(r)$
- Die Zerlegung hat keinen Informationsverlust (lossless-join decomposition) wenn für jede gültige Ausprägung r in R gilt:

$$r = \pi_{R_1}(r) \bowtie \dots \bowtie \pi_{R_n}(r)$$

- D.h. die in R erhaltene Information muss über den natürlichen Verbund der Relationen R_1, \dots, R_n rekonstruierbar sein

Verlustlose Zerlegung – Gegenbeispiel

$R(A,B,C) \rightarrow R_1(A,C), R_2(B,C)$



- $r \subset r_1 \bowtie r_2 \Rightarrow$ die Zerlegung ist **nicht verlustlos**

Kriterien für die Verlustlosigkeit

- Charakterisierung verlustloser Zerlegungen auf Basis von FD
- **Theorem.** Eine Zerlegung von R in R_1 und R_2 hat keinen Informationsverlust, falls einer der folgenden Bedingungen gilt:
 - $(R_1 \cap R_2 \rightarrow R_1) \in F_R^+$ oder
 - $(R_1 \cap R_2 \rightarrow R_2) \in F_R^+$, wobei F_R = Menge der FDs aus R .
- Mit anderen Worten: es gelte $RS = \alpha \cup \beta \cup \gamma$, $RS_1 = \alpha \cup \beta$, $RS_2 = \alpha \cup \gamma$ mit paarweisen disjunkten Attributmengen. Dann muss:
$$\beta \in \alpha^+ \text{ oder } \gamma \in \alpha^+$$
- **Corollary.** Wenn für R $\alpha \rightarrow \beta$ gilt, dann ist die Zerlegung $\{R - \beta, \alpha \cup \beta\}$ eine verlustlose Zerlegung.

Verlustlose Zerlegung - Beispiel

- $R(A,B,C)$
- FDs: $F = \{A \rightarrow B\}$
- Die Zerlegung $\{AB, AC\}$ ist verlustlos weil:
 - $AB \cap AC = A$ und $A \rightarrow AB$
- Die Zerlegung $\{AB, BC\}$ ist nicht verlustlos weil:
 - $AB \cap BC = B$ und keiner der FDs $B \rightarrow AB$ oder $B \rightarrow BC$ gilt in R

Verlustlose Zerlegung – Transitivität

- **Theorem.** Wenn $\{R_1, R_2\}$ eine verlustlose Zerlegung von R und $\{R_{11}, R_{12}\}$ eine verlustlose Zerlegung von R_1 sind, dann ist $\{R_{11}, R_{12}, R_2\}$ eine verlustlose Zerlegung von R .

Erster Schritt der Zerlegung der Relation *Filmliste* basierend auf der FD :
Titel → Regisseur

Filmliste

| Titel | Regisseur | Kino | Tel. | Zeit |
|-------------------------|-----------|-------------|--------|-------|
| The Hobbit | Jackson | Cinema City | 441111 | 11:30 |
| The Lord of the Rings 3 | Jackson | Cinema City | 441111 | 14:30 |
| Adventures of Tintin | Spielberg | Odeon | 442222 | 11:30 |
| War Horse | Spielberg | Odeon | 442222 | 14:00 |
| The Lord of the Rings 3 | Jackson | Odeon | 442222 | 16:30 |

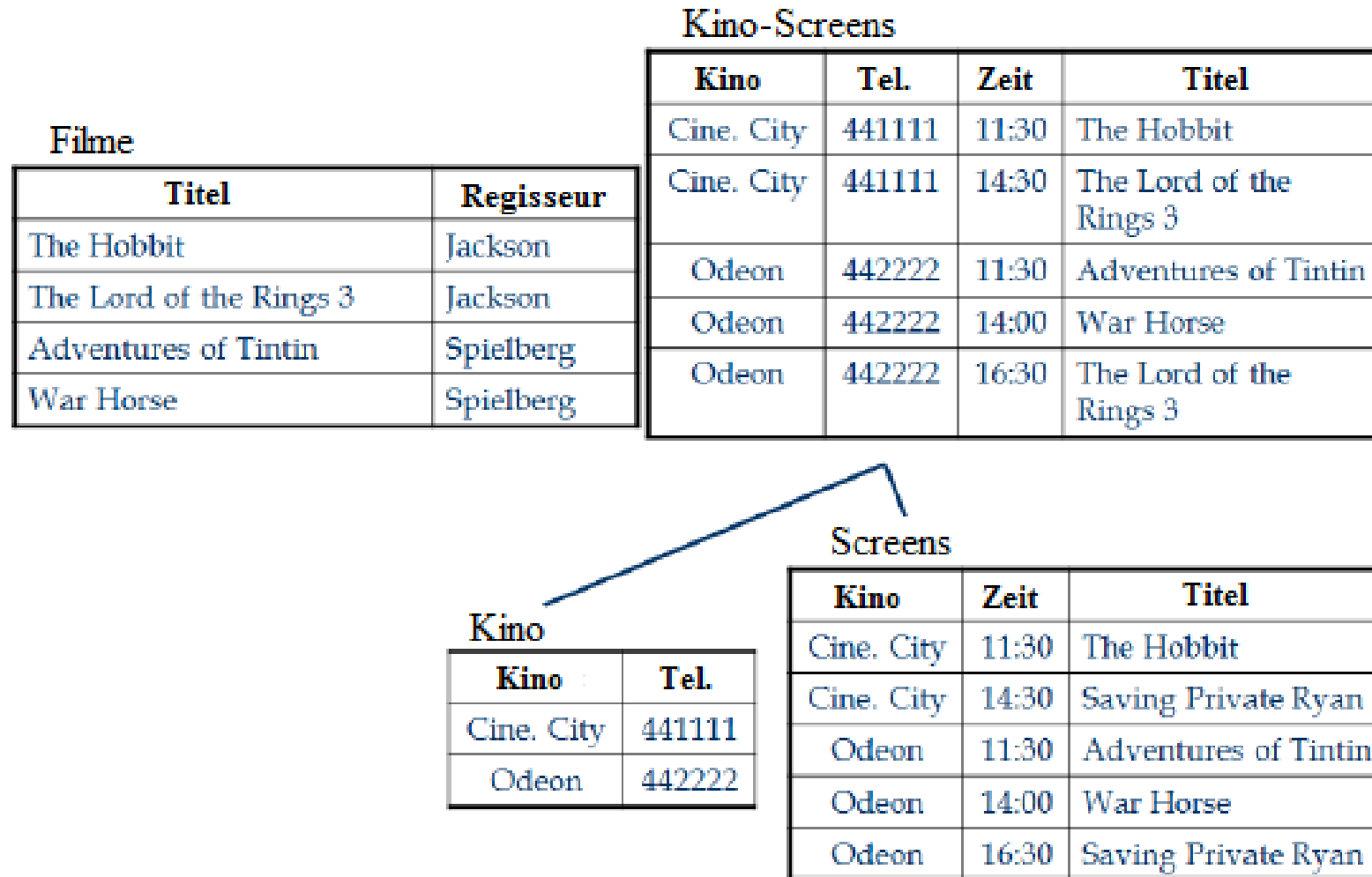
Filme

| Titel | Regisseur |
|-------------------------|-----------|
| The Hobbit | Jackson |
| The Lord of the Rings 3 | Jackson |
| Adventures of Tintin | Spielberg |
| War Horse | Spielberg |

Screens

| Kino | Tel. | Zeit | Titel |
|------------|--------|-------|-------------------------|
| Cine. City | 441111 | 11:30 | The Hobbit |
| Cine. City | 441111 | 14:30 | The Lord of the Rings 3 |
| Odeon | 442222 | 11:30 | Adventures of Tintin |
| Odeon | 442222 | 14:00 | War Horse |
| Odeon | 442222 | 16:30 | The Lord of the Rings 3 |

Zweiter Schritt der Zerlegung der Relation *Filmliste* basierend auf der FD :
 Kino → Telefonnummer (Tel.)



Abhängigkeitsbewahrung

- Idee: Alle FDs, die für die Relation R gelten, sollen *lokal* auf den Relationen R_1, \dots, R_n aus der Zerlegung überprüfbar sein.
- Die Projektion von F auf α (F_α) ist die Menge von FDs aus F^+ die nur Attribute aus α enthalten:

$$F_\alpha = \{ \beta \rightarrow \gamma \in F^+ \mid \beta\gamma \subseteq \alpha \}$$

Abhängigkeitsbewahrung

- Berechnung von FD Projektionen :

Input: F, α

Output: F_α

Erg = \emptyset

For each $\beta \subseteq \alpha$ do

$T = \beta^+$ (bzgl. F)

 Erg = Erg $\cup \{ \beta \rightarrow T \cap \alpha \}$

Return Erg

Hüllentreue Zerlegung

- **Definition.**

Die Zerlegung der Relation R in Relationen R_1, \dots, R_n wird als hüllentreue Zerlegung bezeichnet falls:

$$F_R^+ = (F_{R_1} \cup \dots \cup F_{R_n})^+$$

- Intuitiv, müssen $F_{R_1} \cup \dots \cup F_{R_n}$ und F_R äquivalent sein

Zerlegung Beispiel

- PlzVerzeichnis(Straße, Ort, BLand, PLZ)
- Bedingungen:
 - Orte werden durch *Ort* und *BLand* eindeutig charakterisiert
 - Innerhalb einer Straße ändert sich *PLZ* nicht
 - PLZ-Gebiete gehen nicht über Ortsgrenzen, Orte nicht über Bundeslandgrenzen
- FDs: $\{PLZ\} \rightarrow \{Ort, BLand\}$ und $\{Straße, Ort, BLand\} \rightarrow \{PLZ\}$
- Die Zerlegung $\{PLZ, Straße\}$ und $\{PLZ, Ort, BLand\}$ ist:
 - Verlustlos (Corollary)
 - Nicht abhängigkeiterhaltend (die zweite FD ist *lokal* nicht überprüfbar)

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 abhängigkeiterhaltend möglich?
 - Ja: OK
 - Nein: Ok, aber „unschön“, da Überprüfung der FDs nur nach der Rekonstruktion von R möglich ist (Effizientverlust)