



Kommunikation und Datenhaltung

3. Übung zur Datenhaltung

Relationaler Entwurf & DDL



Agenda

- **Ergänzung zur Vorlesung:
Normalformen**
- Ergänzungen zum zweiten Übungsblatt
- Besprechung des dritten Übungsblatts



Normalformen: 1NF

- Relation in erster Normalform, wenn alle Attribute der Relation atomar.
- Als Attributwerte sind nur Standarddatentypen wie *integer* oder *string* erlaubt.
- **Nicht** erlaubt sind mengenwertige Ausprägungen von Konstruktoren wie *array* oder *set*.

Normalformen 1NF

- Wir legen fest:
 - Ist ein Relationenschema in 2NF, 3NF, BCNF oder 4NF, so ist es auch in 1NF!
 - Die folgenden Definitionen bezüglich 2NF, 3NF, BCNF und 4NF gehen immer davon aus, dass das zugehörige Relationenschema in 1NF ist!
- Um eine Entscheidung treffen zu können, ob ein Relationenschema in 1NF ist, muss bekannt sein, ob alle Attribute des Schemas atomar sind!

Normalformen: 2NF

- Relation in 2NF, wenn keine partiellen Abhängigkeiten zwischen Schlüsseln des Relationenschemas und Nicht-Primattributen bestehen.
- Beispiel
 - $R = ABCD$
 - Schlüssel: A, B
 - Funktionale Abhängigkeiten:
 - $AB \rightarrow C$ Nicht-Primattribut D abhängig von B.
 - $B \rightarrow D$ B allerdings nur **Teil** des Schlüssels!
Daher nicht in 2NF.

Normalformen: 3NF

- Eine Relation ist in 3NF, wenn es keine transitiven Abhängigkeiten $K \rightarrow X \rightarrow Y$ (K Schlüssel, Y Nicht-Primattribut) gibt.
- Verbundtreue und abhängigkeitsbewahrende Zerlegung in 3NF immer möglich.
- Beispiel für transitive Abhängigkeit:
 - $R = ABC$
 - Schlüssel: A
 - Funktionale Abhängigkeiten:
 - $A \rightarrow B$ Nicht-Primattribut C transitiv abhängig von A (Schlüssel).
 - $B \rightarrow C$ Daher nicht in 3NF.

Normalformen: 3NF

- Eine Relation ist in 3NF, wenn es keine transitiven Abhängigkeiten $K \rightarrow X \rightarrow Y$ (K Schlüssel, Y Nicht-Primattribut) gibt.
- Wählt X als Teil eines Schlüssels, so sieht man, dass partielle Abhängigkeiten ein Spezialfall der transitiven sind.
- 2NF wird somit von 3NF impliziert.

Normalformen: BCNF

- Eine Relation ist in BCNF, wenn kein Attribut transitiv von einem Schlüssel abhängt.
- 3NF betrachtet nur transitive Abhängigkeiten von Nicht-Primattributen.
- BCNF widmet sich Redundanzen innerhalb der Schlüsselattribute.
- Verbundtreue Zerlegung in BCNF immer möglich. Nicht notwendigerweise aber eine abhängigkeitsbewahrende!
- Beispiel für transitive Abhängigkeit von einem Schlüssel:
 - $R = ABX$, Schlüssel: A, B
 - Funktionale Abhängigkeiten:
 - $AB \rightarrow X$ Primattribut B transitiv abhängig vom Schlüssel AB.
 - $X \rightarrow B$ Daher nicht in BCNF.

Normalformen: BCNF

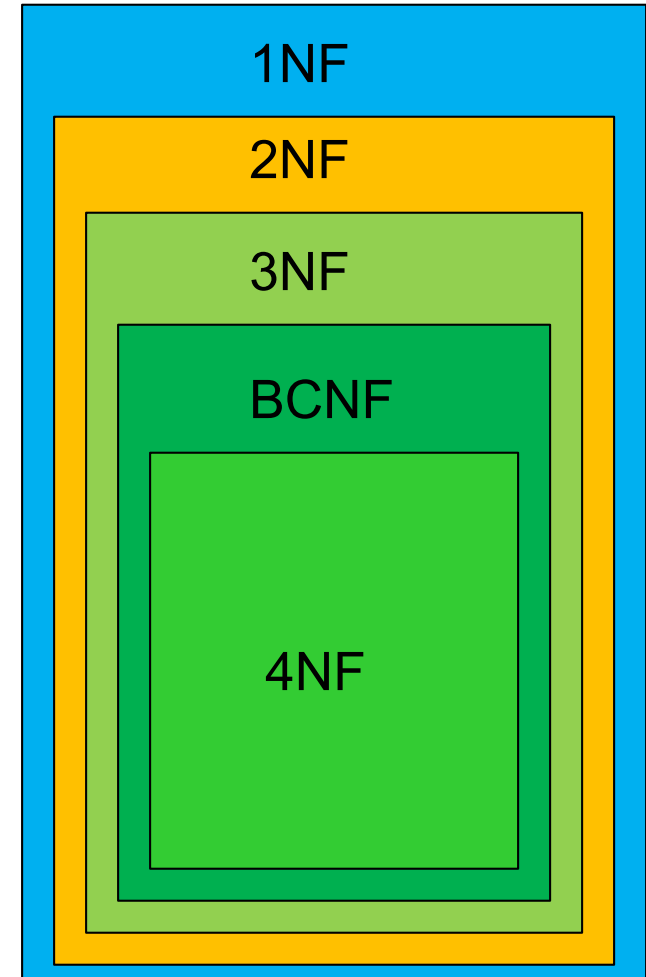
- Definition BCNF
 - $\nexists A \in R$: A transitiv abhängig von einem Schlüssel
- Alternative Def. der 3NF (äquivalent zu der in Vorlesung vorgestellten Def.):
 - $\nexists A \in R$: A transitiv abhängig von einem Schlüssel \wedge A ist Nicht-Primattribut in R
- Klar ersichtlich:
 - Relation in BCNF \Rightarrow Relation in 3NF

Normalformen: 4NF

- Eine Relation ist in 4NF, wenn es keine mehrwertigen, nicht-trivialen Abhängigkeiten zwischen Attributen einer Relation gibt.
- Abhängigkeit $X \twoheadrightarrow Y$ nicht-trivial, wenn außer X und Y noch weitere Attribute im zugehörigen Relationenschema enthalten sind.
- Man kann zeigen:
 - Ist eine Relation in 4NF, so ist sie auch in BCNF.
 - Verbundtreue Zerlegung einer beliebigen Relation in 4NF möglich.

Normalformen in der Übersicht

- Mathematisch beweisbar (gegeben unsere Definition bezüglich 1NF):
 - Relation in 2NF \Rightarrow Relation in 1NF
 - Relation in 3NF \Rightarrow Relation in 2NF
 - Relation in BCNF \Rightarrow Relation in 3NF
 - Relation in 4NF \Rightarrow Relation in BCNF



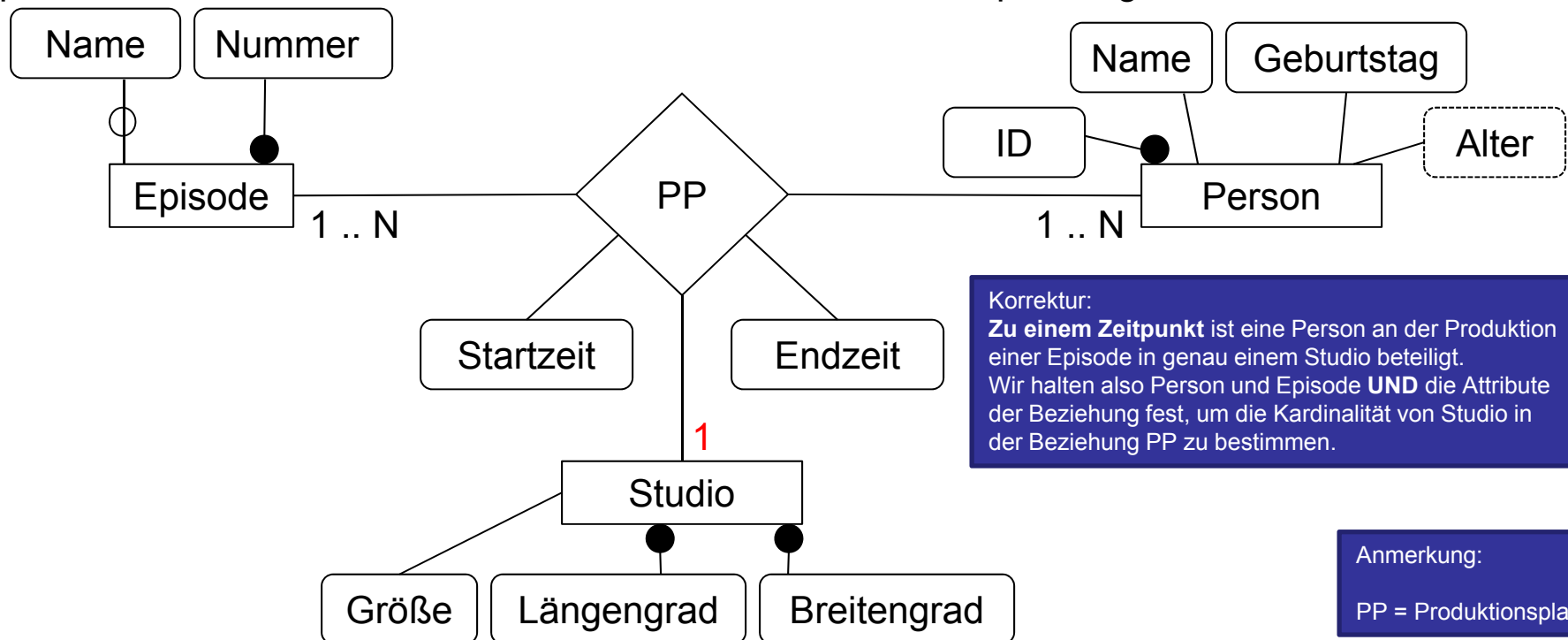


Agenda

- Ergänzung zur Vorlesung:
Normalformen
- **Ergänzungen zum zweiten Übungsblatt**
- Besprechung des dritten Übungsblatts

Kardinalitäten bei dreistelliger Beziehung

Ein Produktionsplan sieht vor, von wann bis wann welche Episode produziert wird, welche Personen beteiligt sind und in welchem Studio gedreht wird. An einer Produktion ist mindestens eine Person beteiligt. Zu einem Zeitpunkt findet eine Produktion immer nur in einem Studio statt. Während einer Produktion können (Teile für) verschiedene Episoden produziert werden, mindestens wird aber Material für eine Episode gedreht.



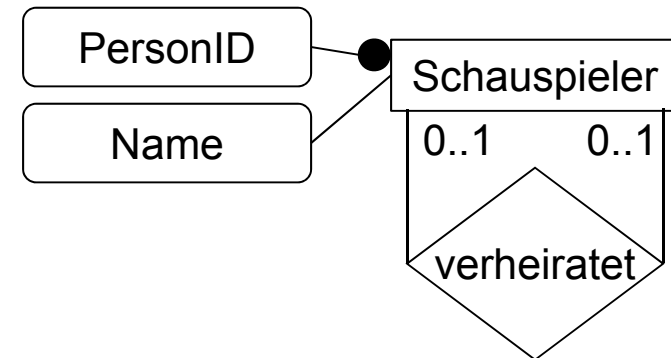
Schlüsselwahl

Setzen Sie alle Entity- und Beziehungstypen kapazitätserhaltend in das Relationenmodell um und geben Sie die notwendigen Schlüssel an. Verwenden Sie dazu die Notation aus der Vorlesung.

Hier Angabe von K wichtig, da es die Information enthält, dass beide Attribute Kandidaten für Schlüssel sind und daher eindeutig sein müssen. Eines der Attribute wird zum Primärschlüssel gewählt (in diesem Fall PersonA_ID), das andere Attribut ist aber immer noch Schlüssel(kandidat) und muss eindeutig sein.

verheiratet (PersonA_ID, PersonB_ID)

$K = \{\{PersonA_ID\}, \{PersonB_ID\}\}$

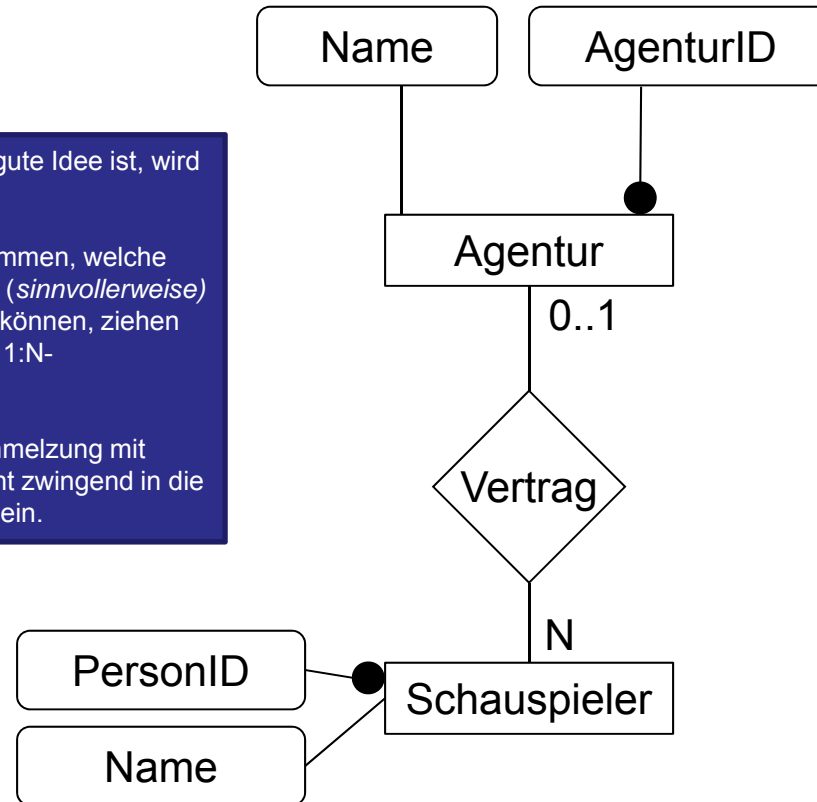


Verschmelzung

Warum das möglicherweise keine gute Idee ist, wird in Aufgabe 2d) behandelt.

Sollte in der Klausur eine Frage kommen, welche Entity-Typen und Beziehungstypen (*sinnvollerweise*) miteinander verschmolzen werden können, ziehen Sie bitte nur 1:1- Beziehungen und 1:N- Beziehungen in Betracht.

Welche Nachteile aber eine Verschmelzung mit einer Entität mit sich bringt, die nicht zwingend in die Relation eingeht, sollte Ihnen klar sein.



Schauspieler (PersonID, Schauspielernamenname, AgenturID, Agenturnamenname)

Nachteile Verschmelzung

PersonID	Schauspielername	AgenturID	Agenturname
1	Al Pacino	1	Some Agency
2	John Travolta	NULL	NULL
3	Joaquin Phoenix	NULL	NULL
4	Jessica Alba	1	Some Agency
5	Julia Roberts	NULL	NULL

- Jeder NULL-Wert kostet Speicherplatz
- Kenntnis über Datenbestand nötig, für sinnvolle Entscheidung, ob Verschmelzung oder nicht
- Redundanzen im Datenbestand (Agenturname)
- Weniger schöne Modellierung: Person und Agentur verschiedene Dinge, nun aber eine Entität



Agenda

- Ergänzung zur Vorlesung:
Normalformen
- Ergänzungen zum zweiten Übungsblatt
- **Besprechung des dritten Übungsblatts**

Aufgabe 1

Gegeben sei eine Datenbasis zur Mensa, die eine einzige Relation R mit folgenden Attributen enthält: Linie (A), Datum (B), Hauptgericht (C), Anzahl Essensausgabestellen (D), Salat (E) und Beilage (F). Alle Attribute sind atomar, also nicht mengenwertig. Es existieren folgende funktionalen Abhängigkeiten:

Funktionale Abhängigkeiten

$AB \rightarrow C$

$A \rightarrow D$

$B \rightarrow EF$

$F \rightarrow E$

Aufgabe 1a: Schlüsselfindung

Funktionale Abhängigkeiten

$AB \rightarrow C$

$A \rightarrow D$

$B \rightarrow EF$

$F \rightarrow E$

- Welches Attribut hängt funktional von keinem anderen ab?
- A steht nur auf der linken Seite!

Aufgabe 1a: Schlüsselfindung

Funktionale Abhängigkeiten

$AB \rightarrow C$

$A \rightarrow D$

$B \rightarrow EF$

$F \rightarrow E$

- Hülle $A^+ =$

Aufgabe 1a: Schlüsselfindung

Funktionale Abhängigkeiten

$AB \rightarrow C$

$A \rightarrow D$

$B \rightarrow EF$

$F \rightarrow E$

- $A^+ = \{ A, D \}$
- Funktional hängen von A also nur A selbst und D ab. A alleine bestimmt nicht alle Attribute (A, B, C, D, E, F).

Aufgabe 1a: Schlüsselfindung

Funktionale Abhängigkeiten

$AB \rightarrow C$

$A \rightarrow D$

$B \rightarrow EF$

$F \rightarrow E$

- Auch B hängt funktional von keinem anderen Attribut ab.
- Hülle $B^+ =$

Schlüsselfindung

Funktionale Abhängigkeiten

$AB \rightarrow C$

$A \rightarrow D$

$B \rightarrow EF$

$F \rightarrow E$

- $B^+ = \{ B, E, F \}$
- Auch B bestimmt nicht alle Attribute A, B, C, D, E und F.

Aufgabe 1a: Schlüsselfindung

Funktionale Abhängigkeiten

$AB \rightarrow C$

$A \rightarrow D$

$B \rightarrow EF$

$F \rightarrow E$

- Außer A oder B weiteres Attribut nur auf linker Seite?
- Nein!
- Bestimmen A und B alle Attribute?
- $AB^+ =$

Aufgabe 1a: Schlüsselfindung

- A oder B alleine bestimmen nicht alle Attribute.
- A und B bestimmen alle Attribute der Relation:
 $AB^+ = \{ A, B, C, D, E, F \}$
- A und B lassen sich nicht aus anderen Attributen ableiten.
- Daher AB Schlüssel.

Aufgabe 1b: Höchste Normalform

Funktionale Abhängigkeiten

$AB \rightarrow C$

$A \rightarrow D$

$B \rightarrow EF$

$F \rightarrow E$

- 1NF?
 - Ja, da nach Aufgabenstellung alle Attribute atomar
- 2NF?
 - Gibt es partielle Abhängigkeiten zwischen Schlüsseln des Relationenschemas und weiteren Attributen?

Aufgabe 1b: Höchste Normalform

- Relation in 2NF, wenn keine partiellen Abhängigkeiten zwischen Schlüsseln des Relationenschemas und Nicht-Primattributen bestehen.

Funktionale Abhängigkeiten

$AB \rightarrow C$

$A \rightarrow D$

$B \rightarrow EF$

$F \rightarrow E$

D hängt funktional vom Primattribut A ab.
Damit hängt D **partiell** von einem Schlüssel ab, nämlich dem Schlüssel AB! D selbst ist kein Primattribut. Daher nicht in 2NF!

Auch $B \rightarrow EF$ verstößt gegen diese Bedingung.



Aufgabe 1b: Höchste Normalform

- Da Relation nicht in 2NF, aber in 1NF, ist 1NF die höchste Normalform, in der die Relation ist.

Aufgabe 1c: 3NF

Geben Sie eine verbundtreue und abhängigkeitstreue Zerlegung von R an, deren Relationen in dritter Normalform sind.

- **Ausgangssituation**

- R = ABCDEF
- 1 NF, da alle Attribute nicht mengenwertig
- Menge funktionaler Abhängigkeiten:

Funktionale Abhängigkeiten

AB → C

A → D

B → EF

F → E

Aufgabe 1c: 3NF

Funktionale Abhängigkeiten

$AB \rightarrow C$

$A \rightarrow D$

$B \rightarrow EF$

$F \rightarrow E$

D hängt funktional vom Primattribut A ab.
Damit hängt B **partiell** von einem Schlüssel ab, nämlich dem Schlüssel AB! D selbst ist kein Primattribut. Daher nicht in 2NF!

Auch $B \rightarrow EF$ verstößt gegen diese Bedingung.

- Elimination der rechten Seite der partiellen Abhängigkeit und Kopie der linken Seite:
 - $R1 = ABC$
 - $R2 = AD$
 - $R3 = BEF$
- $R1, R2, R3$ nun in 2NF.

Aufgabe 1c: 3NF

Funktionale Abhängigkeiten

$AB \rightarrow C$

$A \rightarrow D$

$B \rightarrow EF$

$F \rightarrow E$

Transitive Abhängigkeit des Nicht-Primattributs E vom Primattribut B:

$B \rightarrow F \rightarrow E$

\Rightarrow Nicht in 3NF

- Elimination von E und Kopie von F:
 - $R3_1 = BF$
 - $R3_2 = FE$
- $R1, R2, R3_1, R3_2$ nun in 3NF.

Aufgabe 1d: BCNF

- Zerlegte Relationen in BCNF?
 - $R1 = ABC$
 - $R2 = AD$
 - $R3_1 = BF$
 - $R3_2 = FE$
- Ja, da in allen Relationen kein Attribut transitiv abhängig von einem anderen Attribut ist!
- $R1, R2, R3$ ist nicht in BCNF!

Aufgabe 2a: Updateanomalie

PersonID	Person	Geburtstag	AlbumID	Album	Jahr
1234	Johnny Cash	26.02.1932	42382	At Folsom Prison	1968
1234	Johnny Cash	26.02.1932	45556	At San Quentin	1969
1234	Johnny Cash	26.02.1932	42536	American Recordings	1994
3456	Elvis Costello	25.08.1954	57834	My Aim Is True	1977
7342	Peter Fox	03.09.1971	85989	Stadtaffe	2008

Geben Sie ein Beispiel, wie es bei dieser Relation zu einer Updateanomalie kommen kann.

Aufgabe 2a: Updateanomalie

PersonID	Person	Geburtstag	AlbumID	Album	Jahr
1234	Johnny Cash	26.02.1932	42382	At Folsom Prison	1968
1234	Johnny Cash	12.12.2000	45556	At San Quentin	1969
1234	Johnny Cash	26.02.1932	42536	American Recordings	1994
3456	Elvis Costello	25.08.1954	57834	My Aim Is True	1977
7342	Peter Fox	03.09.1971	85989	Stadtaffe	2008

```
UPDATE Musik
SET    Geburtstag = '12.12.2000'
WHERE Album = 'At San Quentin'
```

Aufgabe 2a: Updateanomalie

PersonID	Person	Geburtstag	AlbumID	Album	Jahr
1234	Johnny Cash	26.02.1932	42382	At Folsom Prison	1968
1234	Johnny Cash	12.12.2000	45556	At San Quentin	1969
1234	Johnny Cash	26.02.1932	42536	American Recordings	1994
3456	Elvis Costello	25.08.1954	57834	My Aim Is True	1977
7342	Peter Fox	03.09.1971	85989	Stadtaffe	2008

- Zu einer Person liegen nun mehrere Angaben über den Geburtstag vor!
- Inkonsistenzen im Datenbestand die Folge.

Aufgabe 2b: Updateanomalien

Neben potentiellen Updateanomalien bringen die redundanten Einträge weitere Nachteile mit sich:

- Erhöhter Speicherbedarf wegen der redundant zu speichernden Informationen.
- Leistungseinbußen bei Änderungen, da mehrere Einträge abgeändert werden müssen.

Aufgabe 2c: Einfügeanomalien

PersonID	Person	Geburtstag	AlbumID	Album	Jahr
1234	Johnny Cash	26.02.1932	42382	At Folsom Prison	1968
1234	Johnny Cash	26.02.1932	45556	At San Quentin	1969
1234	Johnny Cash	26.02.1932	42536	American Recordings	1994
3456	Elvis Costello	25.08.1954	57834	My Aim Is True	1977
7342	Peter Fox	03.09.1971	85989	Stadtaffe	2008

Kann es zu Einfügeanomalien kommen?

Aufgabe 2c: Einfügeanomalie

PersonID	Person	Geburtstag	AlbumID	Album	Jahr
1234	Johnny Cash	26.02.1932	42382	At Folsom Prison	1968
1234	Johnny Cash	26.02.1932	45556	At San Quentin	1969
1234	Johnny Cash	26.02.1932	42536	American Recordings	1994
3456	Elvis Costello	25.08.1954	57834	My Aim Is True	1977
7342	Peter Fox	03.09.1971	85989	Stadtaffe	2008
2222	Michael Jackson	29.08.1958	NULL	NULL	NULL

```
INSERT INTO Musik ('PersonID', 'Person', 'Geburtstag')  
VALUES ('2222', 'Michael Jackson', '29.08.1958')
```

Aufgabe 2c: Einfügeanomalien

PersonID	Person	Geburtstag	AlbumID	Album	Jahr
1234	Johnny Cash	26.02.1932	42382	At Folsom Prison	1968
1234	Johnny Cash	26.02.1932	45556	At San Quentin	1969
1234	Johnny Cash	26.02.1932	42536	American Recordings	1994
3456	Elvis Costello	25.08.1954	57834	My Aim Is True	1977
7342	Peter Fox	03.09.1971	85989	Stadtaffe	2008
2222	Michael Jackson	29.08.1958	NULL	NULL	NULL

- Um die Person Michael Jackson in die Datenbank einzutragen, müssen Angaben über Alben gemacht werden (wieder viele NULL-Werte die Folge).
- Ja, es kann zu Einfügenomalien kommen.

Aufgabe 2d: Löschanomalien

PersonID	Person	Geburtstag	AlbumID	Album	Jahr
1234	Johnny Cash	26.02.1932	42382	At Folsom Prison	1968
1234	Johnny Cash	26.02.1932	45556	At San Quentin	1969
1234	Johnny Cash	26.02.1932	42536	American Recordings	1994
3456	Elvis Costello	25.08.1954	57834	My Aim Is True	1977
7342	Peter Fox	03.09.1971	85989	Stadtaffe	2008

Ist es möglich, Informationen über das Album *Stadtaffe* zu löschen, ohne den Interpreten *Peter Fox* aus der Relation zu entfernen?

Aufgabe 2d: Löschanomalien

PersonID	Person	Geburtstag	AlbumID	Album	Jahr
1234	Johnny Cash	26.02.1932	42382	At Folsom Prison	1968
1234	Johnny Cash	26.02.1932	45556	At San Quentin	1969
1234	Johnny Cash	26.02.1932	42536	American Recordings	1994
3456	Elvis Costello	25.08.1954	57834	My Aim Is True	1977
7342	Peter Fox	03.09.1971	85989	Stadtaffe	2008

```
DELETE FROM Musik WHERE Person = 'Peter Fox'
```

Würde auch alle Informationen über dessen Alben löschen!

Aufgabe 2d: Löschanomalien

PersonID	Person	Geburtstag	AlbumID	Album	Jahr
1234	Johnny Cash	26.02.1932	42382	At Folsom Prison	1968
1234	Johnny Cash	26.02.1932	45556	At San Quentin	1969
1234	Johnny Cash	26.02.1932	42536	American Recordings	1994
3456	Elvis Costello	25.08.1954	57834	My Aim Is True	1977
7342	Peter Fox	03.09.1971	NULL	NULL	NULL

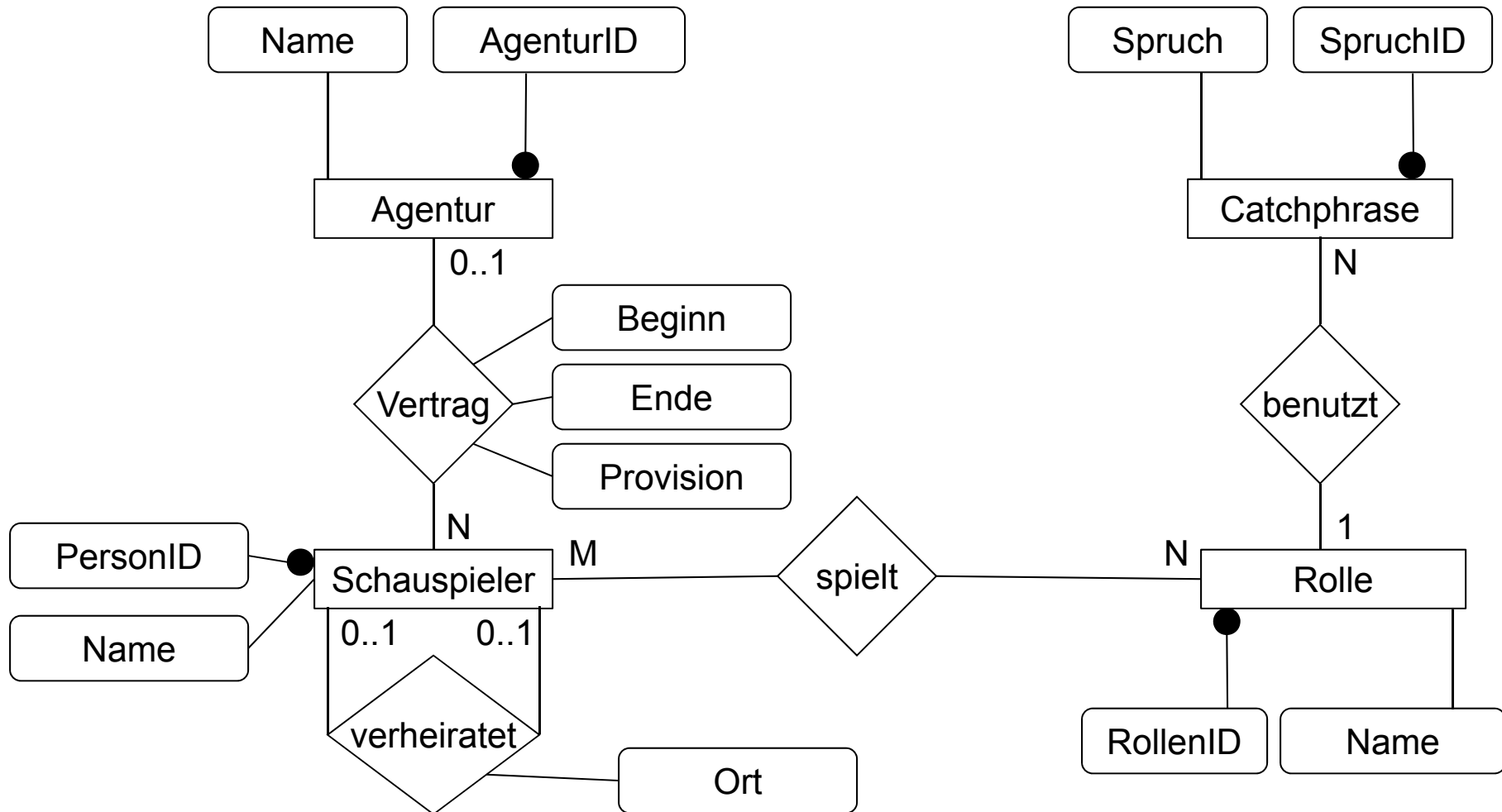
```
UPDATE Musik
SET AlbumID = NULL, Album = NULL, Jahr = NULL
WHERE Person = 'Peter Fox'
```

Aufgabe 2d: Löschanomalien

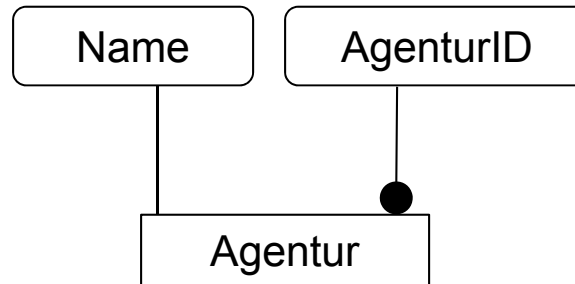
PersonID	Person	Geburtstag	AlbumID	Album	Jahr
1234	Johnny Cash	26.02.1932	42382	At Folsom Prison	1968
1234	Johnny Cash	26.02.1932	45556	At San Quentin	1969
1234	Johnny Cash	26.02.1932	42536	American Recordings	1994
3456	Elvis Costello	25.08.1954	57834	My Aim Is True	1977
7342	Peter Fox	03.09.1971	NULL	NULL	NULL

- Unintuitiv
- Fehleranfällig
- Prinzipiell aber möglich, Informationen über Personen zu löschen, ohne deren Alben mitzulöschen

Aufgabe 3a: Schemaerweiterung

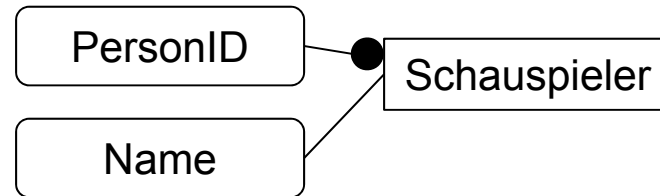


Aufgabe 3b: DDL



```
CREATE TABLE Agentur (  
    Name VARCHAR(100),  
    AgenturID INTEGER,  
    PRIMARY KEY (AgenturID)  
);
```

Aufgabe 3b: DDL



```
CREATE TABLE Schauspieler (  
    Name VARCHAR(100),  
    PersonID INTEGER,  
    PRIMARY KEY (PersonID)  
);
```

Aufgabe 3b: DDL

```
CREATE TABLE Vertrag (  
    Beginn DATE,  
    CHECK (Beginn NOT BETWEEN (SELECT Beginn FROM Vertrag V WHERE  
    PersonID = V.PersonID) AND (SELECT Ende [...])),  
    Ende DATE,  
    CHECK (Ende NOT BETWEEN [...]),  
    Provision DECIMAL(5, 2), CHECK (Provision>0 AND Provision<100),  
    PersonID INTEGER,  
    AgenturID INTEGER,  
    PRIMARY KEY (PersonID, AgenturID, Beginn),  
    FOREIGN KEY (PersonID)  
        REFERENCES Schauspieler (PersonID)  
        ON UPDATE CASCADE ON DELETE RESTRICT,  
    FOREIGN KEY (AgenturID)  
        REFERENCES Agentur (AgenturID)  
        ON UPDATE CASCADE ON DELETE RESTRICT  
);
```



Alternativ:
Provision DECIMAL(4, 2),
CHECK (Provision > 0),

Aufgabe 3b: DDL

```
CREATE TABLE verheiratet (  
  Ort VARCHAR(100) NOT NULL,  
  PersonA INTEGER,  
  PersonB INTEGER,  
  PRIMARY KEY (PersonA),  
  UNIQUE (PersonB),  
  FOREIGN KEY (PersonA)  
    REFERENCES Schauspieler (PersonID)  
    ON UPDATE CASCADE ON DELETE RESTRICT,  
  FOREIGN KEY (PersonB)  
    REFERENCES Schauspieler (PersonID)  
    ON UPDATE CASCADE ON DELETE RESTRICT,  
  CHECK(NOT EXISTS(  
    (SELECT PersonA_ID FROM verheiratet)  
    INTERSECT  
    (SELECT PersonB_ID FROM verheiratet) )) );
```

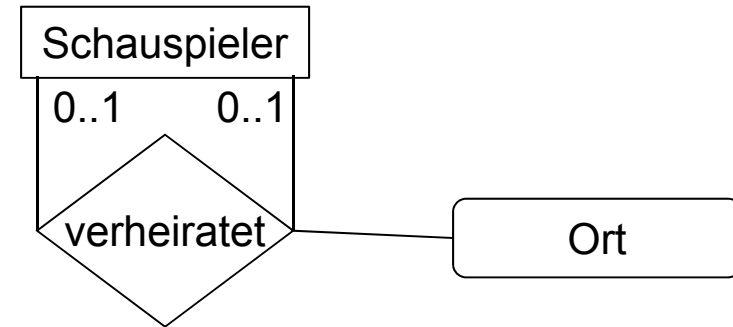


Illustration Integritätsbedingungen

```
CHECK (  
  NOT EXISTS (  
    (SELECT PersonA_ID  
     FROM verheiratet)  
  INTERSECT  
    (SELECT PersonB_ID  
     FROM verheiratet)  
  )  
)
```

```
PRIMARY KEY PersonA_ID,  
UNIQUE PersonB_ID
```

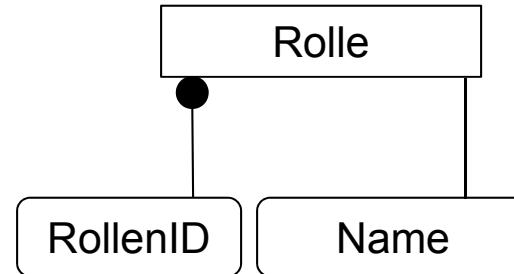
verhindert

PersonA_ID	PersonB_ID
1	2
2	3

verhindert

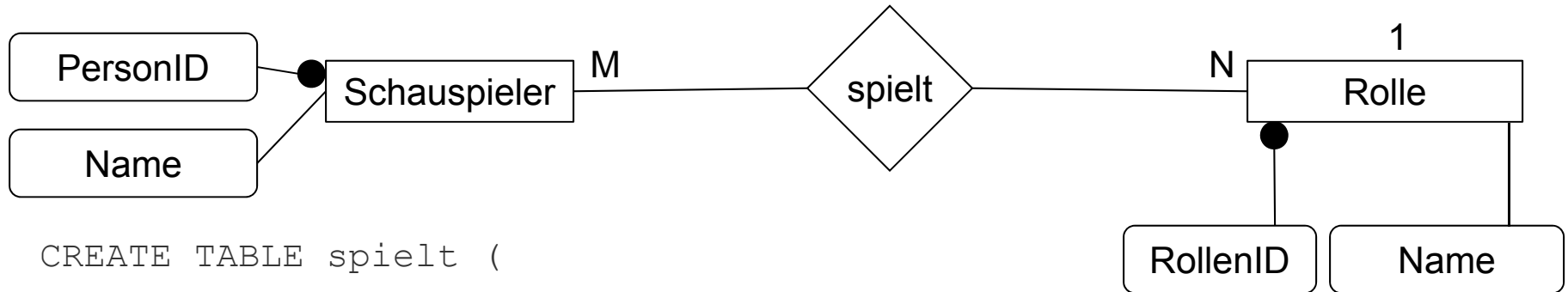
PersonA_ID	PersonB_ID
2	1
3	1

Aufgabe 3b: DDL



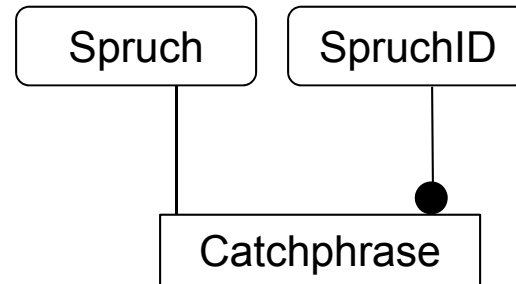
```
CREATE TABLE Rolle (  
    Name VARCHAR(100),  
    RollenID INTEGER,  
    PRIMARY KEY (RollenID)  
);
```

Aufgabe 3b: DDL



```
CREATE TABLE spielt (  
    PersonID INTEGER,  
    RollenID INTEGER,  
    PRIMARY KEY (PersonID, RollenID),  
    FOREIGN KEY (PersonID)  
        REFERENCES Schauspieler (PersonID)  
        ON UPDATE CASCADE,  
    FOREIGN KEY (RollenID)  
        REFERENCES Rolle (RollenID)  
        ON UPDATE CASCADE  
);
```

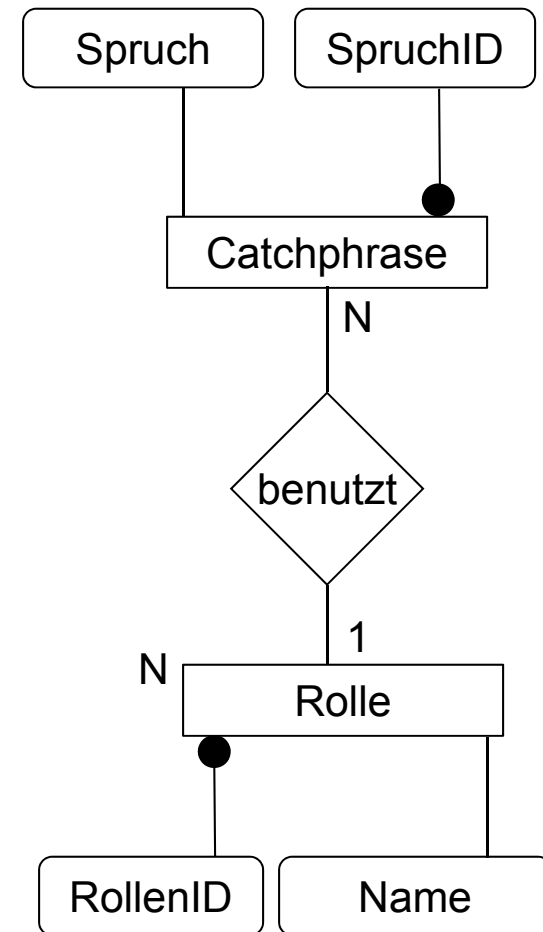
Aufgabe 3b: DDL



```
CREATE TABLE Catchphrase (  
    Spruch VARCHAR(100),  
    SpruchID INTEGER,  
    PRIMARY KEY (SpruchID)  
);
```

Aufgabe 3b: DDL

```
CREATE TABLE benutzt (  
  SpruchID INTEGER,  
  RollenID INTEGER,  
  PRIMARY KEY (SpruchID),  
  FOREIGN KEY (SpruchID)  
    REFERENCES Catchphrase (SpruchID)  
    ON UPDATE CASCADE,  
  FOREIGN KEY (RollenID)  
    REFERENCES Rolle (RollenID)  
    ON UPDATE CASCADE  
);
```





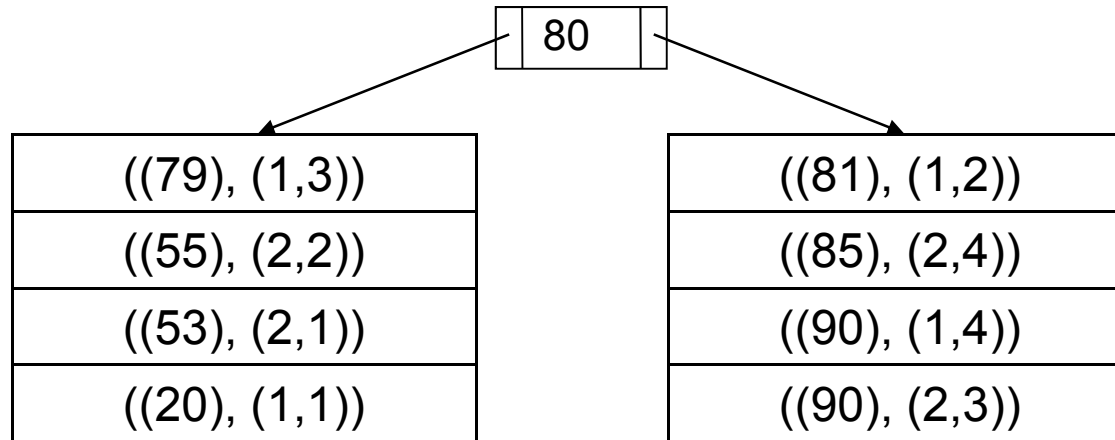
Aufgabe 3c: Index

Erzeugen Sie mittels eines SQL-Befehls eine Indexstruktur, die das Auffinden von Schauspielern erleichtert, denen Agenturen eine besonders hohe Provision abverlangen.

```
CREATE INDEX ProvisionIndex  
ON Vertrag (Provision desc);
```

Index – Illustration

- Index für (Provision):



Vertrag (AgenturID, SchauspielerID, Beginn, Ende, Provision) :

1	100	01.01.2002	01.01.2005	20
5	200	01.01.1999	01.10.1999	81
3	100	01.01.2009	01.01.2011	79
4	400	01.01.2002	01.01.2005	90

10	500	01.01.2002	01.01.2005	53
20	200	01.01.1998	01.10.2007	55
30	100	01.05.2009	01.11.2011	90
40	300	01.07.2004	01.01.2005	85