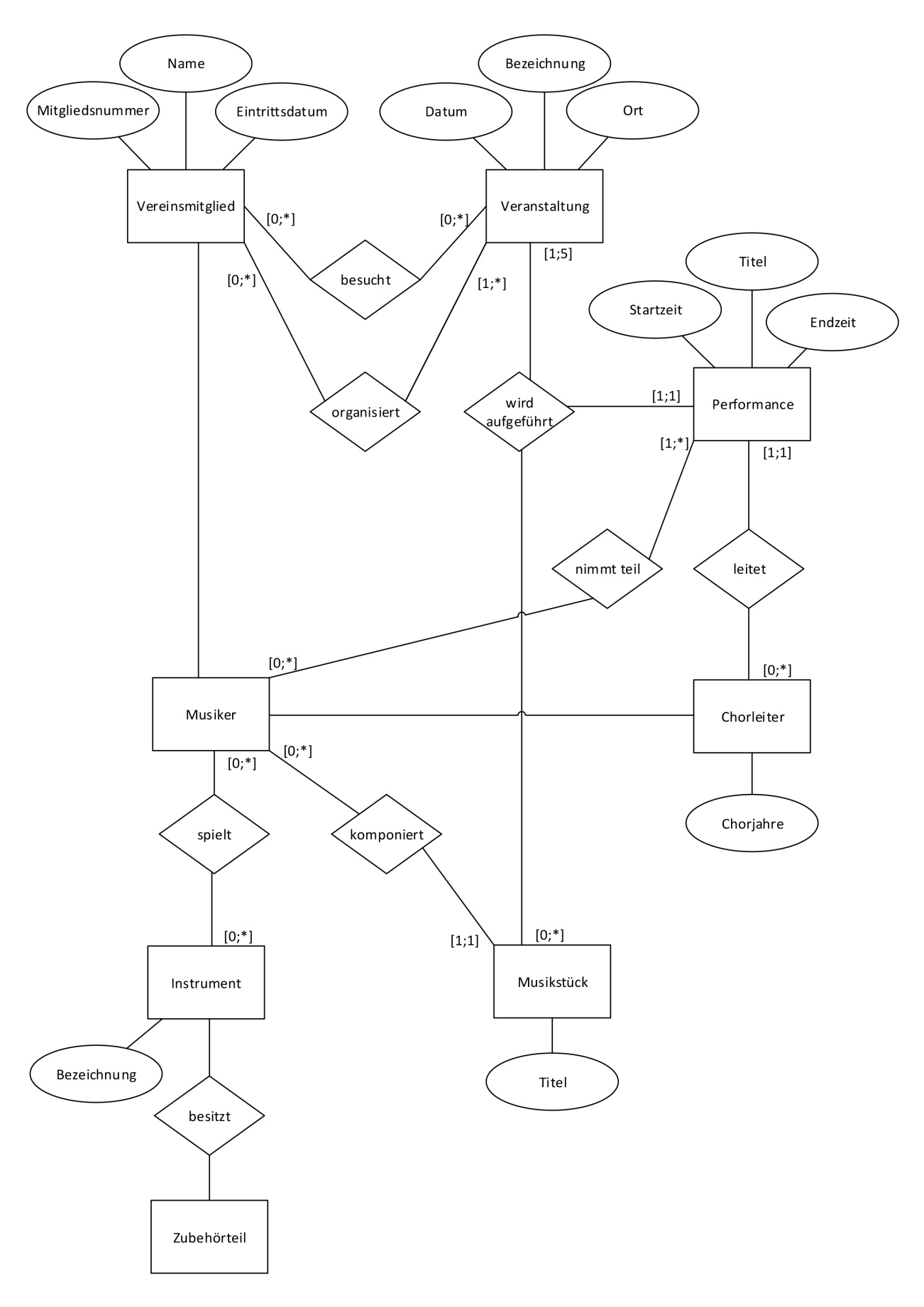
GDB Aufgabenblatt 3

Abgegeben von Timon Back, Fabian Behrendt, Nicolai Stäger

# Aufgabe 1: Informationsmodellierung mit dem Entity-Relationship-Modell



# Aufgabe 2: Abbildung eines ER-Diagramms auf das relationale Datenmodell

## a) Entwickeln Sie aus dem dargestellten ER-Diagramm ein entsprechendes relationales Datenbankschema anhand der in der Vorlesung erläuterten Abbildungsregeln.

Farbe (RGB, CMYK)

Baustein (Form, Farbe->Farbe.RGB)

Bild (Baustein->Form, Bild)

Modell (Name, Datum, Grad)

Thema (Bez)

Thema-Modell-Zugehörigkeit (Thema->Thema.Bez, Modell->Modell(Name, Datum))

Set (SNr, Alter, Thema->Thema.Bez)

Verkaufsset (SNr, Alter, Thema->Thema.Bez, LPreis)

Werbeset (SNr, Alter, Thema->Thema.Bez, Firma)

Teil (Teil-Anzahl, Baustein->Baustein.Form, Farbe->Farbe.RGB, Modell->Modell(Name, Datum), Set->Set.SNr)

## b) Wieso ist die Verwendung des Hausklassenmodells problematisch?

Die Verwendung des Hausklassenmodells ist problematisch, da die Relationen zwar weiterhin die minimale Anzahl von Relationen aufweisen, jedoch nicht mehr redundanzfrei sind. Somit werden Daten doppelt gespeichert, was zu einem höherem Speicherverbrauch führt, als auch zu potenziellen Inkonsistenzen in den Daten, wenn die Daten nicht in allen Relationen aktualisiert werden.

# Aufgabe 3: Relationale Algebra und SQL

## 3.A

Übersetzen Sie die folgenden umgangssprachlich formulierten Anfragen in einen zugehörigen Ausdruck der relationalen Algebra. Werten Sie die Ausdrücke aus und geben Sie jeweils die Ergebnisrelation an.

1. „Die Titel der Bücher, die mehr als 200 Seiten besitzen und nach 1950 erschienen sind.“

|  |
| --- |
| Titel |
| Hundert Jahre Einsamkeit |
| Requiem für einen Traum |
| Der Talisman |

1. „Die Vor- und Nachnamen der Personen, die das Buch mit dem Titel ’Der Talisman’ geschrieben haben.“

|  |  |
| --- | --- |
| Vorname | Nachname |
| Stephen | King |
| Peter | Straub |

1. „Die Vor- und Nachnamen der Personen, die ihr Lieblingsbuch begutachtet haben.“

|  |  |
| --- | --- |
| Vorname | Nachname |
| Leo | Tolstoi |
| Fjodor | Dostojewski |
| Gabriel | Gracía Márquez |

## 3.B

Interpretieren Sie die folgenden relationen Ausdrücke, indem Sie eine umgangssprachliche Beschreibung sowie die Ergebnisrelation angeben (Hinweis: die Syntax entspricht dem relationalen Umbennenungs-Operator, der ein Attribut mit dem Namen B in A umbenennt).

1. „Alle Bücher, die noch nicht begutachtet worden sind“

|  |  |  |  |
| --- | --- | --- | --- |
| Titel | Erscheinungsjahr | Seitenzahl | Verlag |
| Schall und Wahn | 1929 | 304 | Diogenes |
| Der Talisman | 1984 | 714 | Heyne |

1. „Die Vor- und Nachnamen aller Personen, die ein Buch, das sie geschrieben haben, auch begutachtet haben.“

|  |  |
| --- | --- |
| Vorname | Nachname |
| --- | ---- |

1. „Die Vor- und Nachnamen aller Personen, die sowohl ein Buch geschrieben, als auch begutachtet haben“

|  |  |
| --- | --- |
| Vorname | Nachname |
| Leo | Tolstoi |
| Fjodor | Dostojewski |
| Albert | Camus |
| William | Faulkner |
| Stephen | King |
| Peter | Straub |
| Gabriel | García Márquez |

## 3.C

Übersetzen Sie die folgenden Ausdrücke in entsprechende SQL-Ausdrücke und geben Sie die jeweiligen Ergebnisrelationen an.

1. Vor- und Nachnamen der Autoren ohne Duplikate von Büchern mit mehr als 500 Seiten.

SELECT DISTINCT Person.Vorname, Person.Nachname FROM Person, Schreibt, Buch WHERE Buch.Seitenzahl > 500 AND Buch.Titel = Schreibt.Buch AND Schreibt.Autor = Person.PID

|  |  |
| --- | --- |
| Vorname | Nachname |
| Leo | Tolstoi |
| Fjodor | Dostojewski |
| Stephen | King |
| Peter | Straub |

1. Die Titel aller Bücher, die einen Autor haben, der schon mal ein Buch begutachtet hat.

SELECT Schreibt.Titel FROM Schreibt, Begutachtet WHERE Schreibt.Autor = Begutachtet.Lektor

|  |
| --- |
| Titel |
| Krieg und Frieden |
| Anna Karenina |
| Schuld und Sühne |
| Der Fremde |
| Schall und Wahn |
| Als ich im Sterben lag |
| Der Talisman |
| Der Talisman |
| Hindert Jahre Einsamkeit |

„Vor- und Nachname aller Personen die noch nicht Begutachter/Lektor waren“

SELECT Person.Vorname, Person.Nachname FROM Person WHERE Person.PID NOT IN (SELECT Begutachtet.Lektor FROM Begutachtet)

|  |  |
| --- | --- |
| Vorname | Nachname |
| Huber | Selby |

# Aufgabe 4: Algebraische Optimierung