

Teil IV

Datenbankentwurf

1. Phasen des Datenbankentwurfs
2. Weiteres Vorgehen beim Entwurf
3. Kapazitätserhaltende Abbildungen
4. ER-auf-RM-Abbildung

Lernziele für heute . . .

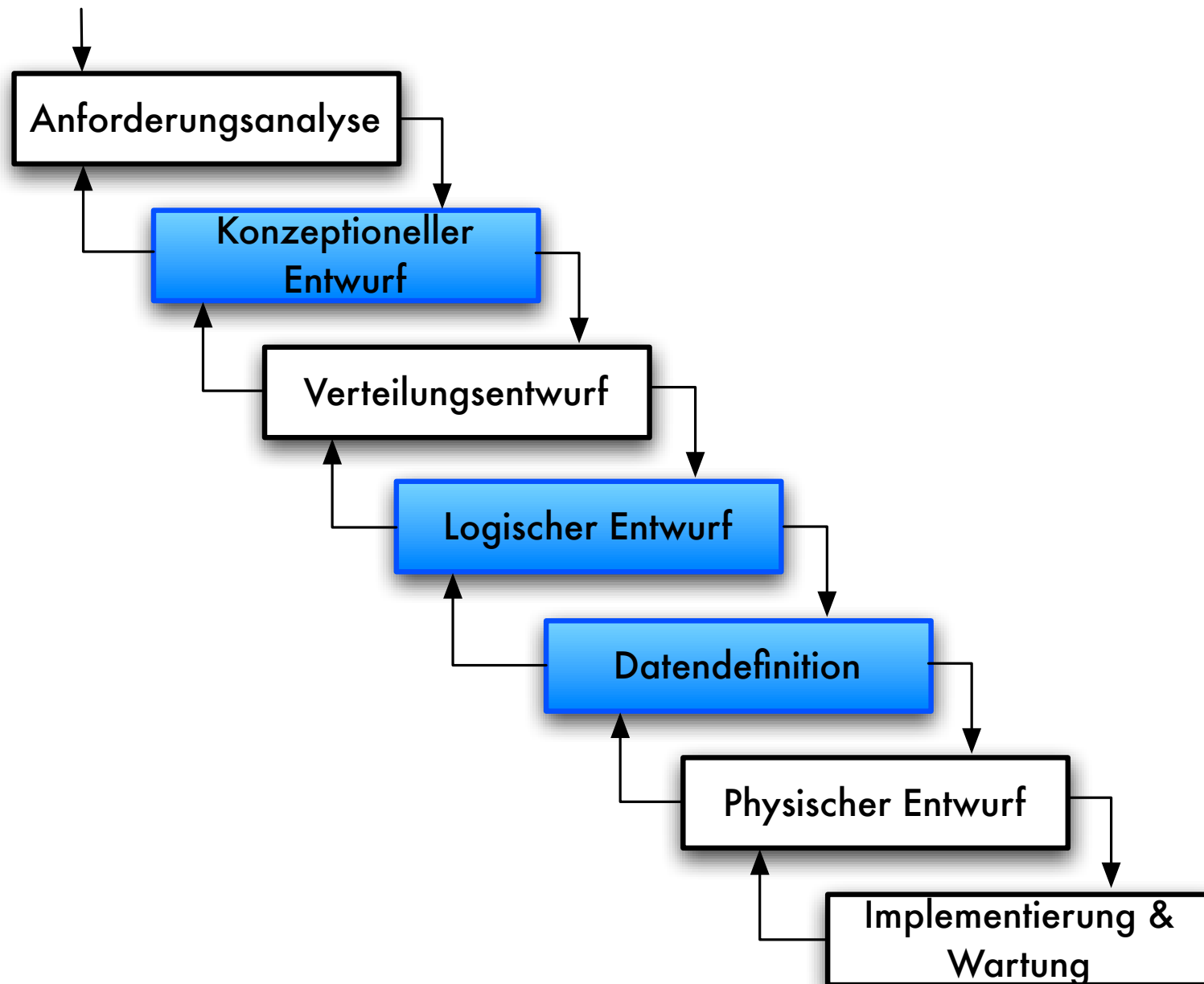
- Kenntnisse über Ziele und Ablauf des Datenbankentwurfsprozesses
- Kenntnisse der Regeln zur Abbildung von ER-Schemata auf Relationenschemata



Phasen des Datenbankentwurfs

- Datenhaltung für mehrere Anwendungssysteme und mehrere Jahre
- daher: besondere Bedeutung
- Anforderungen an Entwurf
 - Anwendungsdaten jeder Anwendung sollen aus Daten der Datenbank ableitbar sein (und zwar möglichst effizient)
 - nur „vernünftige“ (wirklich benötigte) Daten sollen gespeichert werden
 - nicht-redundante Speicherung

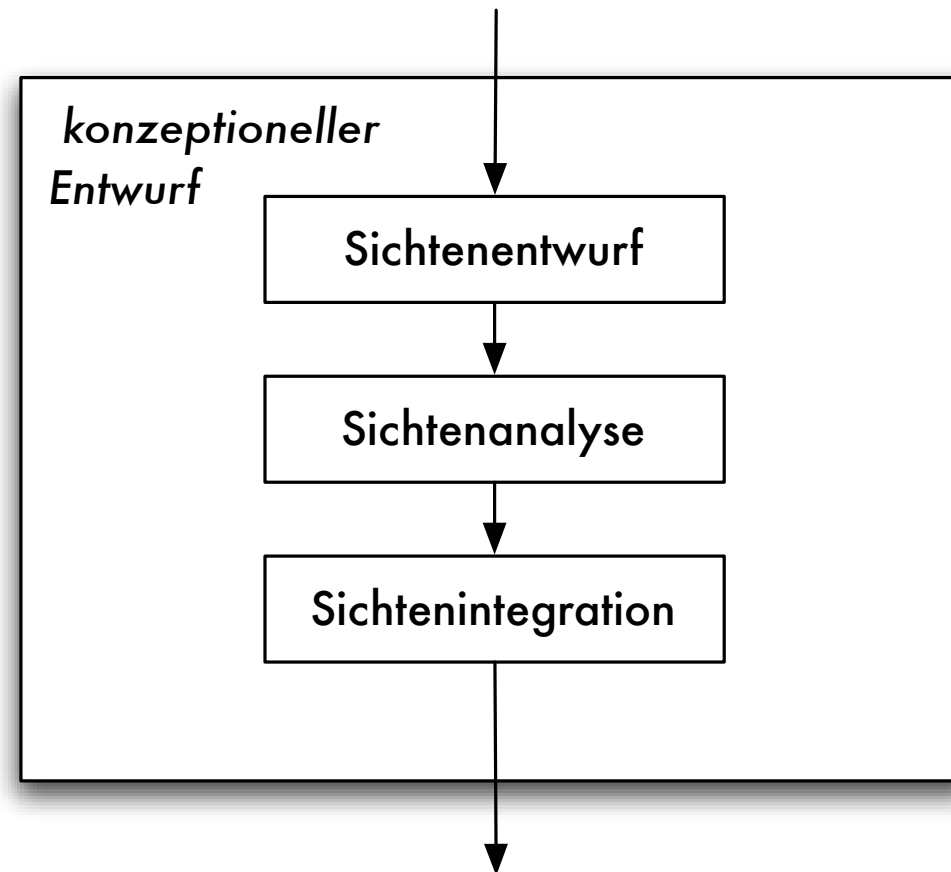
Phasenmodell



- **Vorgehensweise:** Sammlung des Informationsbedarfs in den Fachabteilungen
- **Ergebnis:**
 - informale Beschreibung (Texte, tabellarische Aufstellungen, Formblätter, usw.) des Fachproblems
 - Trennen der Information über Daten (Datenanalyse) von den Information über Funktionen (Funktionsanalyse)
- **„Klassischer“ DB-Entwurf:**
 - nur Datenanalyse und Folgeschritte
- **Funktionsentwurf:**
 - siehe Methoden des Software Engineering

- erste formale Beschreibung des Fachproblems
- **Sprachmittel:** semantisches Datenmodell
- **Vorgehensweise:**
 - Modellierung von Sichten z.B. für verschiedene Fachabteilungen
 - Analyse der vorliegenden Sichten in Bezug auf Konflikte
 - Integration der Sichten in ein Gesamtschema
- **Ergebnis:** konzeptionelles Gesamtschema, z.B. ER-Diagramm

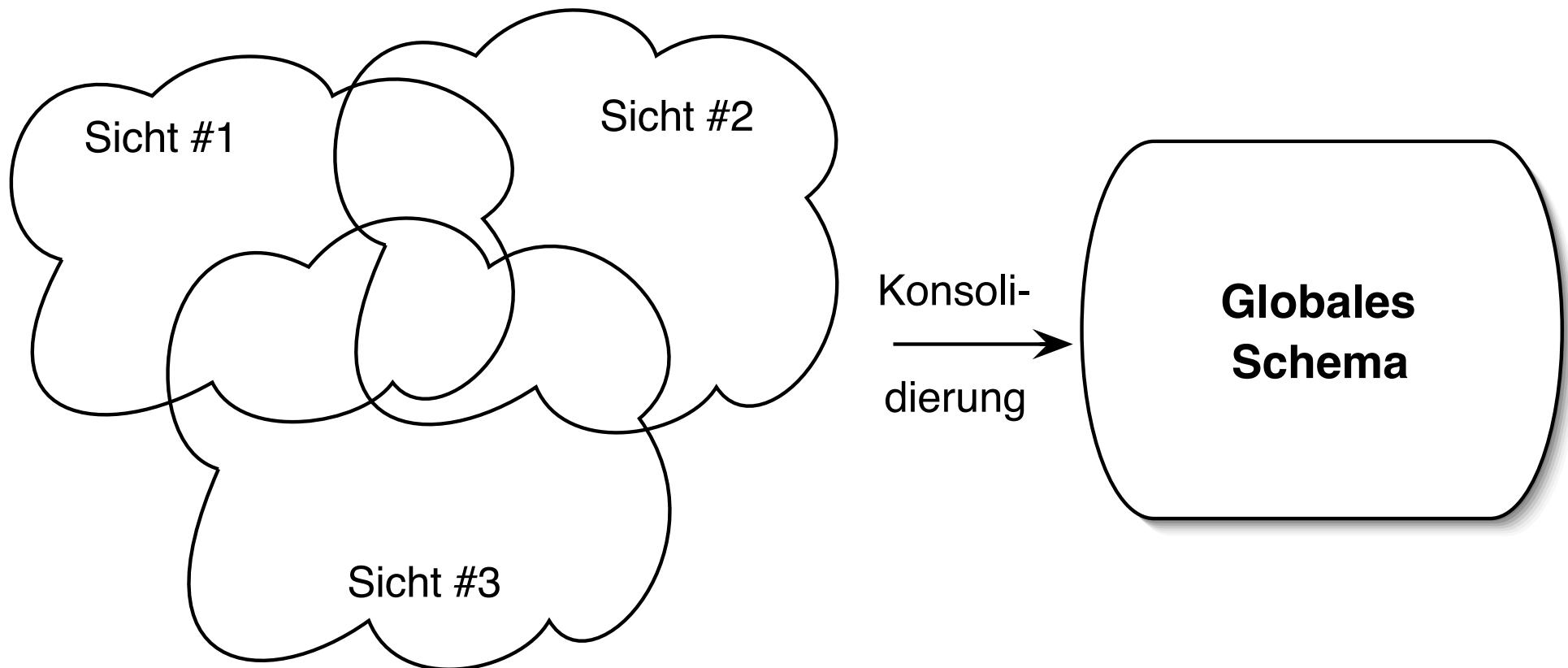
Phasen des konzeptionellen Entwurf



Weiteres Vorgehen beim Entwurf

- ER-Modellierung von verschiedenen **Sichten** auf Gesamtinformation, z.B. für verschiedene Fachabteilungen eines Unternehmens \rightsquigarrow **konzeptueller Entwurf**
 - Analyse und Integration der Sichten
 - Ergebnis: konzeptionelles Gesamtschema
- Verteilungsentwurf bei verteilter Speicherung
- Abbildung auf konkretes Implementierungsmodell (z.B. Relationenmodell) \rightsquigarrow **logischer Entwurf**
- Datendefinition, Implementierung und Wartung \rightsquigarrow **physischer Entwurf**

- Analyse der vorliegenden Sichten in Bezug auf Konflikte
- Integration der Sichten in ein Gesamtschema



- **Namenskonflikte:** Homonyme / Synonyme
 - Homonyme: Schloss; Kunde
 - Synonyme: Auto, KFZ, Fahrzeug
- **Typkonflikte:** verschiedene Strukturen für das gleiche Element
- **Wertebereichskonflikte:** verschiedene Wertebereiche für ein Element
- **Bedingungskonflikte:** z.B. verschiedene Schlüssel für ein Element
- **Strukturkonflikte:** gleicher Sachverhalt durch unterschiedliche Konstrukte ausgedrückt

- sollen Daten auf mehreren Rechnern verteilt vorliegen, muss Art und Weise der **verteilten Speicherung** festgelegt werden
- z.B. bei einer Relation
KUNDE (KNr, Name, Adresse, PLZ, Konto)
 - **horizontale** Verteilung:
KUNDE_1 (KNr, Name, Adresse, PLZ, Konto)
where PLZ < 50.000
KUNDE_2 (KNr, Name, Adresse, PLZ, Konto)
where PLZ >= 50.000
 - **vertikale** Verteilung (Verbindung über KNrAttribut):
KUNDE_Adr (KNr, Name, Adresse, PLZ)
KUNDE_Konto (KNr, Konto)

- **Sprachmittel:** Datenmodell des ausgewählten „Realisierungs“-DBMS z.B. relationales Modell
- **Vorgehensweise:**
 1. (automatische) Transformation des konzeptionellen Schemas z.B. ER \rightarrow relationales Modell
 2. Verbesserung des relationalen Schemas anhand von Gütekriterien (Normalisierung, siehe Kapitel 5):
Entwurfsziele: Redundanzvermeidung, ...
- **Ergebnis:** logisches Schema, z.B. Sammlung von Relationenschemata

- Umsetzung des logischen Schemas in ein konkretes Schema
- **Sprachmittel:** DDL und DML eines DBMS z.B. Oracle, DB2, SQL Server
 - Datenbankdeklaration in der DDL des DBMS
 - Realisierung der Integritätssicherung
 - **Definition der Benutzersichten**

- Ergänzen des physischen Entwurfs um Zugriffsunterstützung bzgl. Effizienzverbesserung, z.B. Definition von Indexen
- Index
 - Zugriffspfad: Datenstruktur für zusätzlichen, schlüsselbasierten Zugriff auf Tupel ($\langle \textit{Schlüsselattributwert}, \textit{Tupeladresse} \rangle$)
 - meist als B*-Baum realisiert
- **Sprachmittel:** *Speicherstruktursprache* SSL

Indexe in SQL

```
create [ unique ] index indexname
  on relname (
    attrname [ asc | desc ],
    attrname [ asc | desc ],
    ...
  )
```

- Beispiel

```
create index WeinIdx on WEINE (Name)
```

Notwendigkeit für Zugriffspfade



- Beispiel: Tabelle mit 100 GB Daten, Festplattentransferrate ca. 50 MB/s
- Operation: Suchen eines Tupels (Selektion)
- Implementierung: sequentielles Durchsuchen
- Aufwand: $102.400/50 = 2.048 \text{ sec.} \approx 34 \text{ min.}$

- Phasen
 - der Wartung,
 - der weiteren Optimierung der physischen Ebene,
 - der Anpassung an neue Anforderungen und Systemplattformen,
 - der Portierung auf neue Datenbankmanagementsysteme
 - etc.

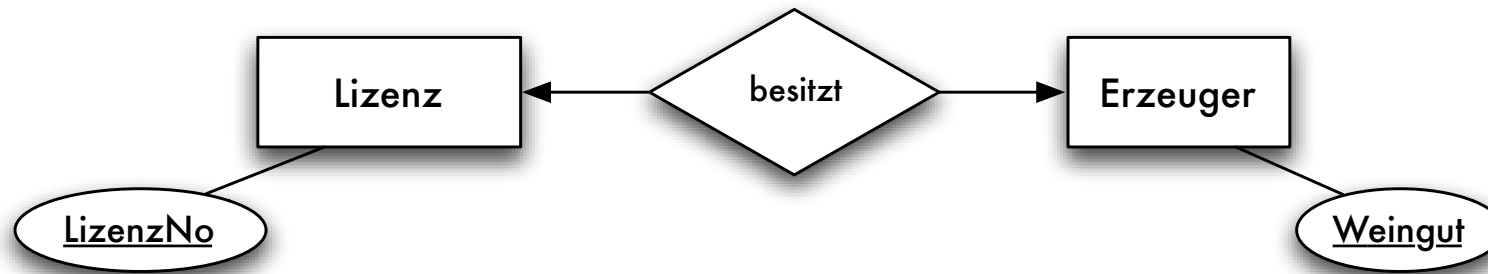
Kapazitätserhaltende Abbildungen

Umsetzung des konzeptionellen Schemas



- Umsetzung auf logisches Schema
 - Beispiel: ER \rightarrow RM
 - korrekt?
 - Qualität der Abbildung?
- Erhaltung der **Informationskapazität**
 - Kann man nach der Abbildung genau die selben Daten abspeichern wie vorher?
 - ... oder etwa mehr?
 - ... oder etwa weniger?

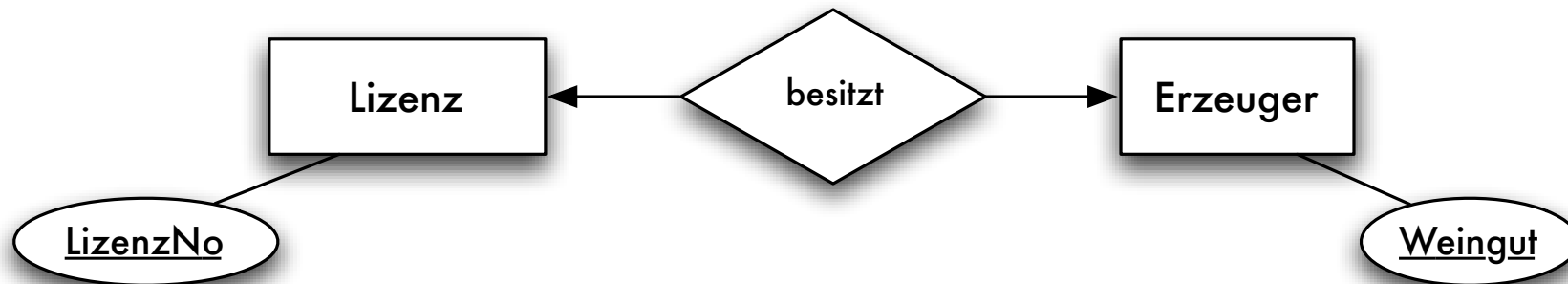
Kapazitätserhöhende Abbildung



- Abbildung auf $R = \{\underline{\text{LizenzNo}}, \text{Weingut}\}$ mit genau einem Schlüssel $K = \{\{\text{LizenzNo}\}\}$
- mögliche ungültige Relation:

| BESITZT | LizenzNo | Weingut |
|---------|----------|---------|
| | 007 | Helena |
| | 42 | Helena |

Kapazitätserhaltende Abbildung



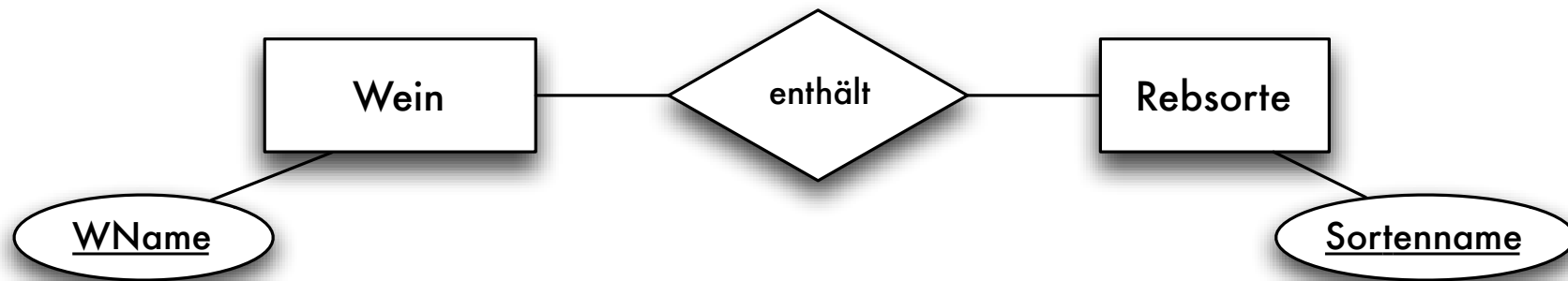
- korrekte Ausprägung

| BESITZT | LizenzNo | Weingut |
|---------|-----------|------------------|
| | 007 42 | Helena Müller |

- korrekte Schlüsselmenge

$$K = \{\{LizenzNo\}, \{Weingut\}\}$$

Kapazitätsvermindernde Abbildung



- Relationenschema mit einem Schlüssel {WName}
- als Ausprägung nicht mehr möglich:

| ENTHÄLT | WName | Sortenname |
|---------|---|---|
| | Zinfandel Red Blossom Bordeaux Blanc Bordeaux Blanc | Zinfandel Cabernet Sauvignon Muscadelle |

Kapazitätserhaltende Abbildung

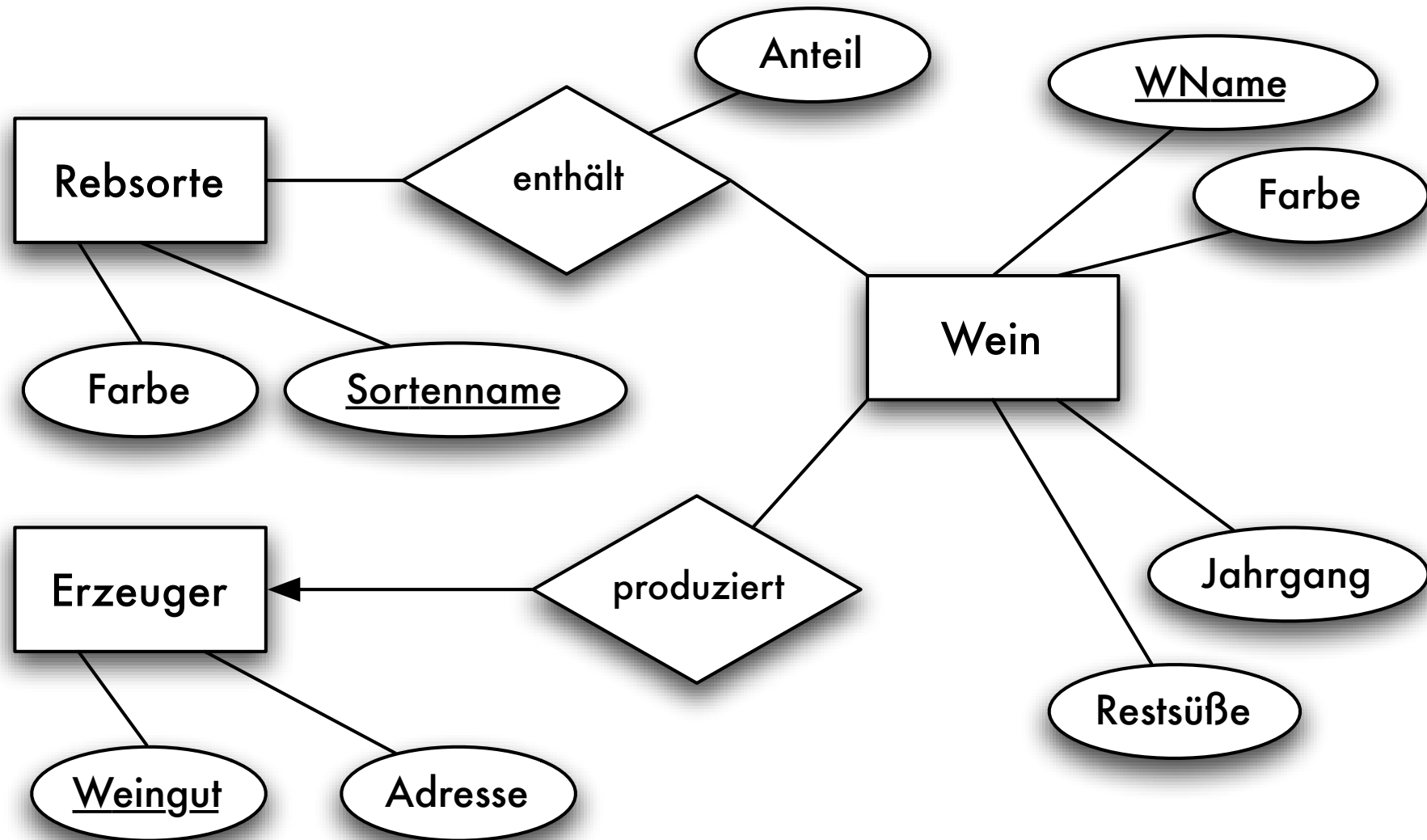


- kapazitätserhaltend mit Schlüssel beider Entity-Typen im Relationenschema als neuer Schlüssel

$$K = \{\{WName, Sortenname\}\}$$

ER-auf-RM-Abbildung

Beispielabbildung ER-RM: Eingabe



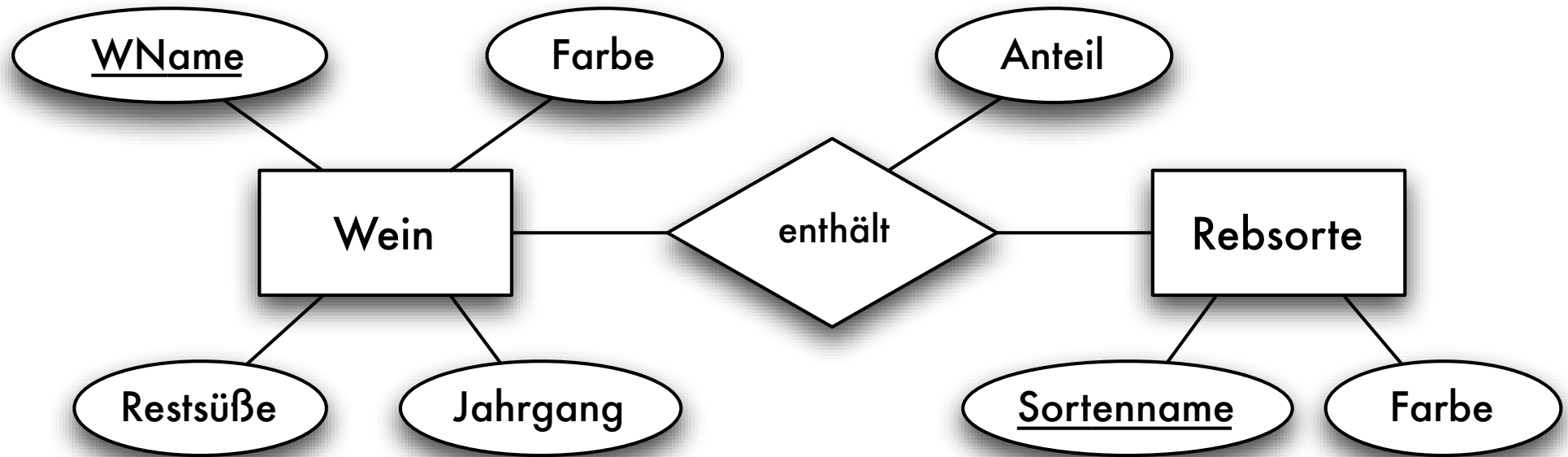
Beispielabbildung ER-RM: Ergebnis



1. REBSORTE(Farbe, Sortenname)
2. ENTHÄLT(Sortenname → REBSORTE, WName → WEIN, Anteil)
3. WEIN(Farbe, WName, Jahrgang, Restsüße)
4. PRODUZIERT(WName → WEIN, Weingut → ERZEUGER)
5. ERZEUGER(Weingut, Adresse)

- **Entity-Typen und Beziehungstypen:** jeweils auf Relationenschemata
- **Attribute:** Attribute des Relationenschemas, **Schlüssel** werden übernommen
- **Kardinalitäten** der Beziehungen: durch Wahl der Schlüssel bei den zugehörigen Relationenschemata ausgedrückt
- in einigen Fällen: **Verschmelzen** der Relationenschemata von Entity- und Beziehungstypen
- zwischen den verbleibenden Relationenschemata diverse **Fremdschlüsselbedingungen** einführen

- neues Relationenschema mit allen Attributen des Beziehungstyps, zusätzlich Übernahme aller Primärschlüssel der beteiligten Entity-Typen
- **Festlegung der Schlüssel:**
 - **m:n-Beziehung:** beide Primärschlüssel zusammen werden Schlüssel im neuen Relationenschema
 - **1:n-Beziehung:** Primärschlüssel der n-Seite (bei der funktionalen Notation die Seite ohne Pfeilspitze) wird Schlüssel im neuen Relationenschema
 - **1:1-Beziehung:** beide Primärschlüssel werden je ein Schlüssel im neuen Relationenschema, der Primärschlüssel wird dann aus diesen Schlüsseln gewählt

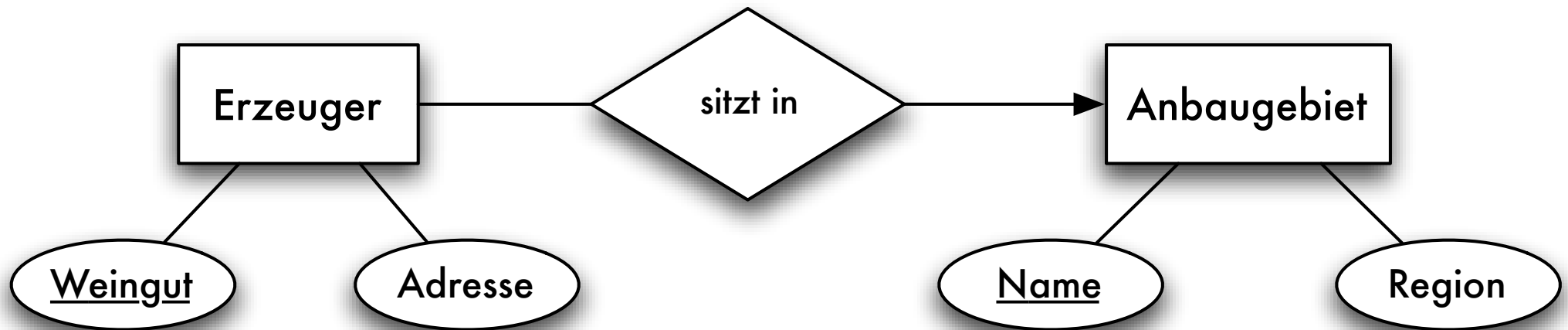


- Umsetzung

1. REBSORTE(Farbe, Sortenname)
2. ENTHÄLT(Sortenname → REBSORTE, WName → WEIN, Anteil)
3. WEIN(Farbe, WName, Jahrgang, Restsüße)

- Attribute Sortenname und WName sind gemeinsam Schlüssel

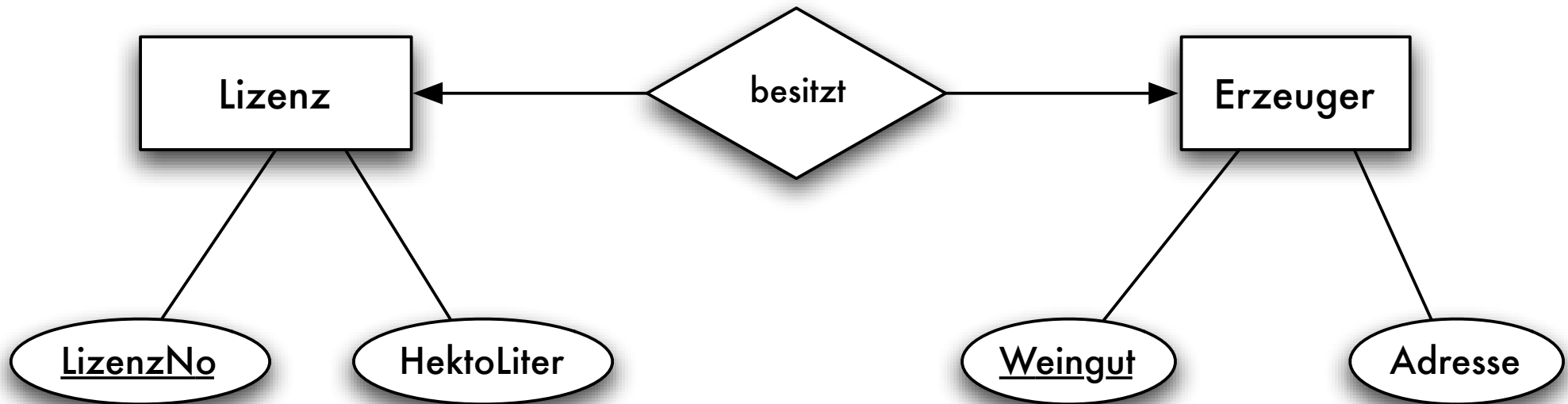
1:n-Beziehungen



- Umsetzung (zunächst)
 - ERZEUGER mit den Attributen Weingut und Adresse,
 - ANBAUGEBIET mit den Attributen Name und Region und
 - SITZT_IN mit den Attributen Weingut und Name und dem Primärschlüssel der n -Seite Weingut als Primärschlüssel dieses Schemas.

- **optionale Beziehungen** ($[0,1]$ oder $[0,n]$) werden nicht verschmolzen
- bei Kardinalitäten $[1,1]$ oder $[1,n]$ (**zwingende Beziehungen**) Verschmelzung möglich:
 - **1:n-Beziehung**: das Entity-Relationenschema der n-Seite kann in das Relationenschema der Beziehung integriert werden
 - **1:1-Beziehung**: beide Entity-Relationenschemata können in das Relationenschema der Beziehung integriert werden

1:1-Beziehungen



- Umsetzung (zunächst)
 - ERZEUGER mit den Attributen Weingut und Adresse
 - LIZENZ mit den beiden Attributen LizenzNo und Hektoliter
 - BESITZT mit den Primärschlüsseln der beiden beteiligten Entity-Typen jeweils als Schlüssel dieses Schemas, also LizenzNo und Weingut

1:1-Beziehungen: Verschmelzung

| ERZEUGER | Weingut | Adresse | LizenzNo | Hektoliter |
|----------|----------------|----------|----------|------------|
| | Rotkäppchen | Freiberg | 42-007 | 10.000 |
| | Weingut Müller | Dagstuhl | 42-009 | 250 |

1:1-Beziehungen: Verschmelzung /2

Erzeuger ohne Lizenz erfordern Nullwerte:

| | | | | |
|----------|-------------------------------|----------------------|-------------|-------------|
| ERZEUGER | Weingut | Adresse | LizenzNo | Hektoliter |
| | Rotkäppchen Weingut Müller | Freiberg Dagstuhl | 42-007 ⊥ | 10.000 ⊥ |

freie Lizenzen führen zu weiteren Nullwerten:

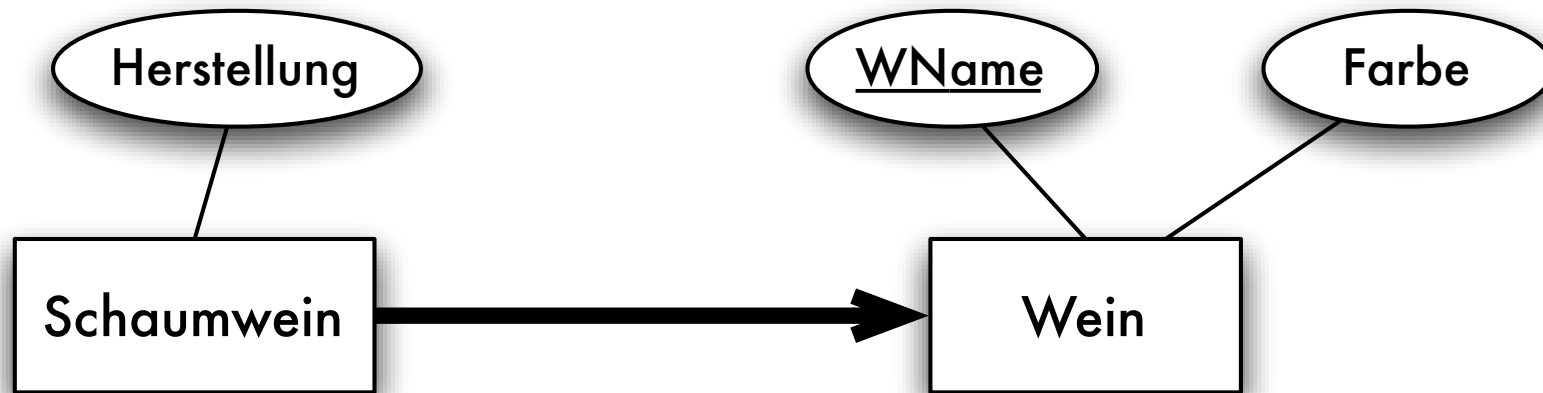
| | | | | |
|----------|------------------------------------|---------------------------|-----------------------|------------------------|
| ERZEUGER | Weingut | Adresse | LizenzNo | Hektoliter |
| | Rotkäppchen Weingut Müller ⊥ | Freiberg Dagstuhl ⊥ | 42-007 ⊥ 42-003 | 10.000 ⊥ 100.000 |



- Umsetzung

1. $\text{WEINJAHRGANG}(\underline{\text{WName}} \rightarrow \text{WEIN}, \text{Jahr}, \text{Restsüße})$
2. $\text{WEIN}(\text{Farbe}, \underline{\text{WName}})$

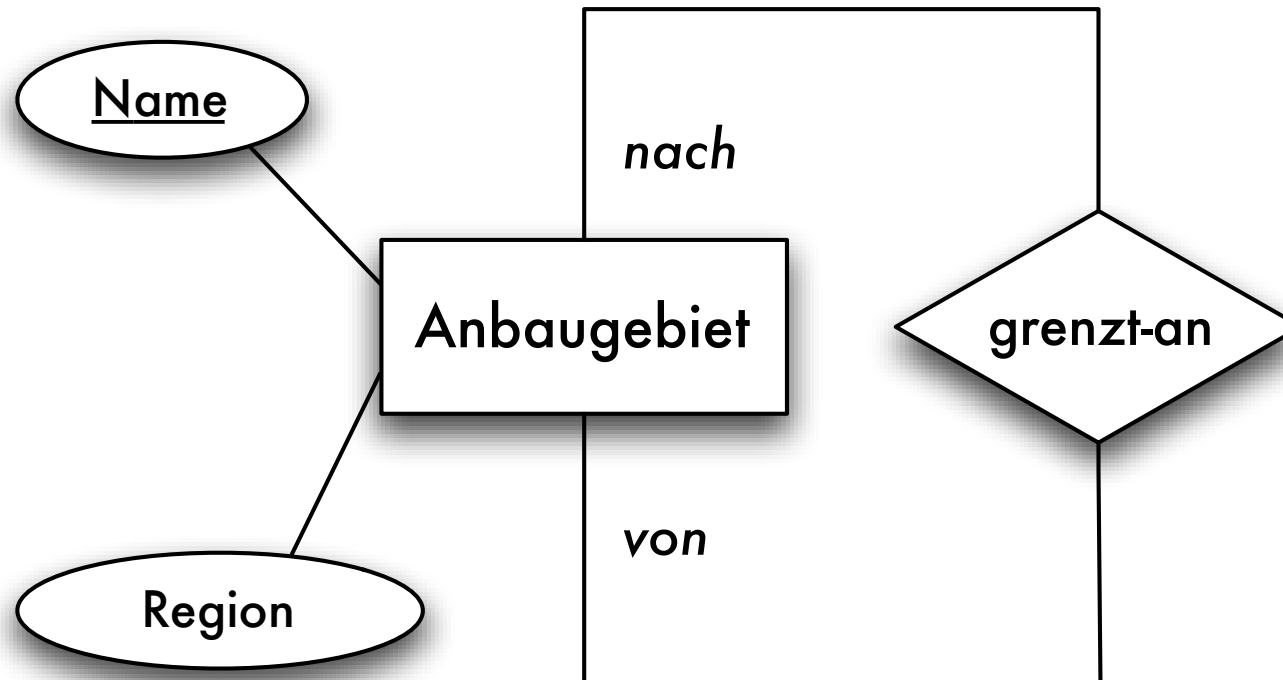
- Attribut **WName** in **WEINJAHRGANG** ist Fremdschlüssel zur Relation **WEIN**



- Umsetzung

1. WEIN(Farbe, WName, Jahrgang, Restsüße)
2. SCHAUMWEIN(WName → WEIN, Herstellung)

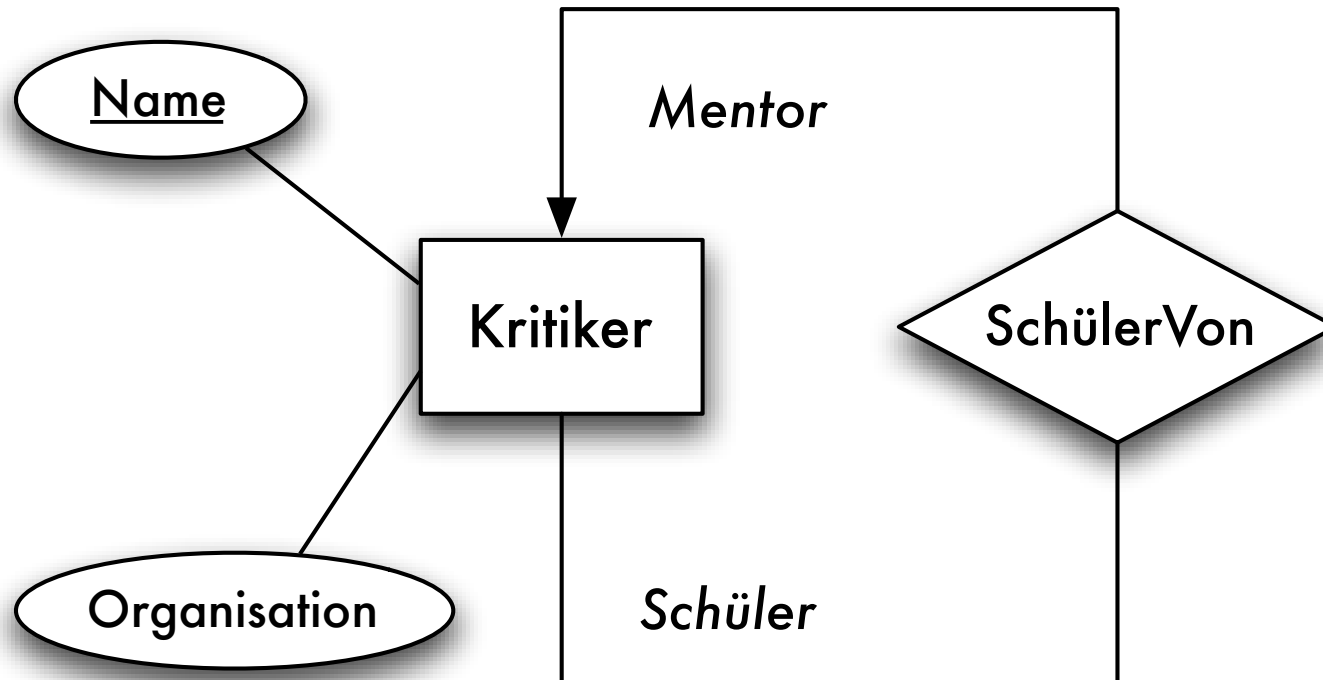
- WName in SCHAUMWEIN ist Fremdschlüssel bezüglich der Relation WEIN



- Umsetzung

1. ANBAUGEBIET(Name, Region)
2. GRENZT_AN(nach → ANBAUGEBIET, von → ANBAUGEBIET)

Rekursive funktionale Beziehungen

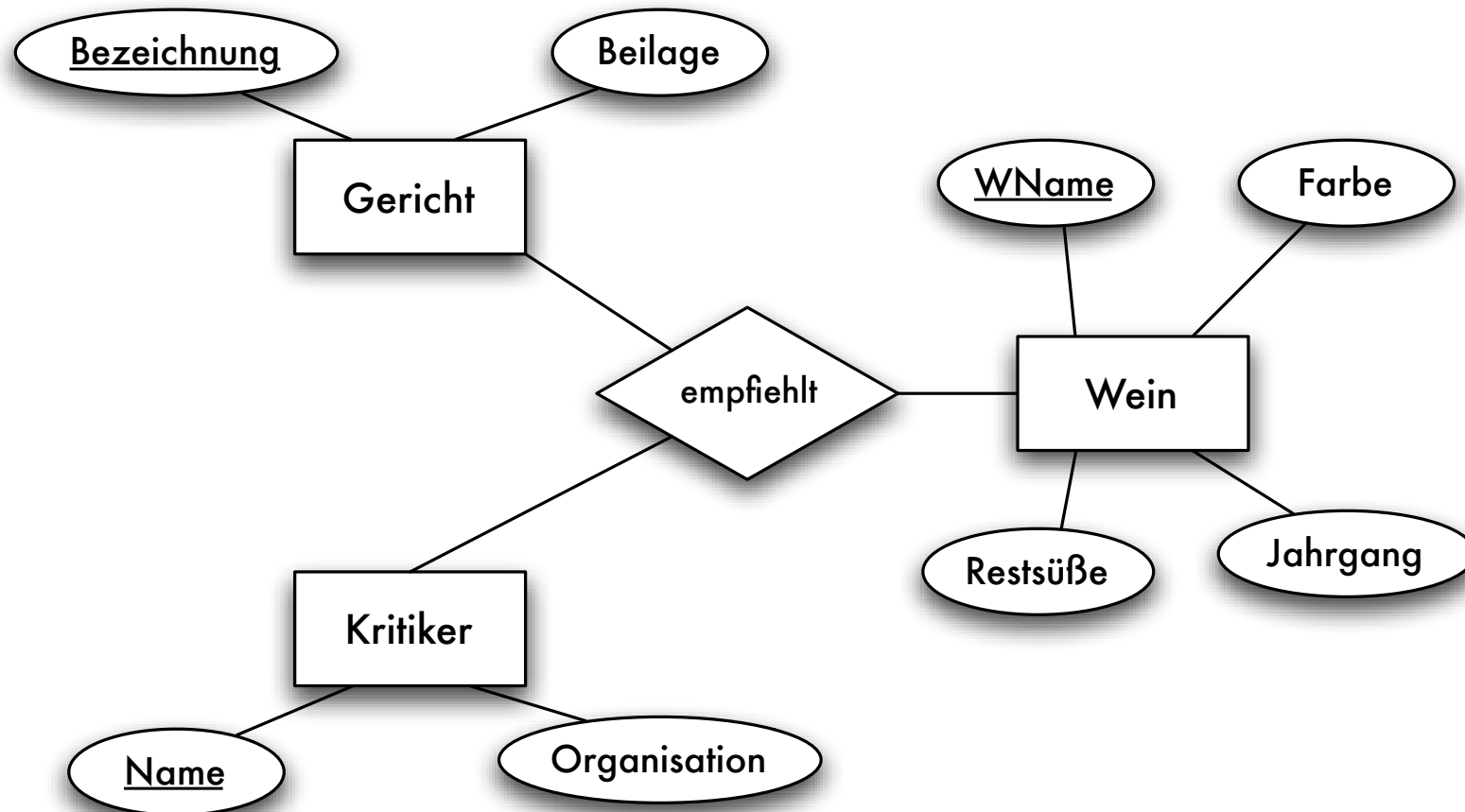


- Umsetzung

1. KRITIKER(Name, Organisation, Mentorname \rightarrow KRITIKER)

- Mentorname ist Fremdschlüssel auf das Attribut Name der Relation KRITIKER.

Mehrstellige Beziehungen



Mehrstellige Beziehungen: Ergebnis



- jeder beteiligte Entity-Typ wird nach den obigen Regeln behandelt
- für Beziehung *Empfiehl*t werden Primärschlüssel der drei beteiligten Entity-Typen in das resultierende Relationenschema aufgenommen
- Beziehung ist allgemeiner Art (k:m:n-Beziehung): alle Primärschlüssel bilden zusammen den Schlüssel
 1. $\text{EMPFIEHLT}(\underline{\text{WName}} \rightarrow \text{WEIN}, \underline{\text{Bezeichnung}} \rightarrow \text{GERICHT}, \underline{\text{Name}} \rightarrow \text{KRITIKER})$
 2. $\text{GERICHT}(\underline{\text{Bezeichnung}}, \text{Beilage})$
 3. $\text{WEIN}(\text{Farbe}, \underline{\text{WName}}, \text{Jahrgang}, \text{Restsüße})$
 4. $\text{KRITIKER}(\underline{\text{Name}}, \text{Organisation})$
- Die drei Schlüsselattribute von EMPFIEHLT sind wiederum Fremdschlüssel

Übersicht über die Transformationen

| ER-Konzept | wird abgebildet auf relationales Konzept |
|--|---|
| Entity-Typ E_i Attribute von E_i Primärschlüssel P_i | Relationenschema R_i Attribute von R_i Primärschlüssel P_i |
| Beziehungstyp dessen Attribute $1 : n$ $1 : 1$ $m : n$ | Relationenschema Attribute: P_1, P_2 weitere Attribute P_2 wird Primärschlüssel der Beziehung P_1 und P_2 werden Schlüssel der Beziehung $P_1 \cup P_2$ wird Primärschlüssel der Beziehung |
| IST-Beziehung | R_1 erhält zusätzlichen Schlüssel P_2 |

E_1, E_2 : an Beziehung beteiligte Entity-Typen,

P_1, P_2 : deren Primärschlüssel,

$1 : n$ -Beziehung: E_2 ist n -Seite,

IST-Beziehung: E_1 ist speziellerer Entity-Typ

- Phasen des Datenbankentwurfs
- Informationskapazität
- Abbildung vom Entity-Relationship-Modell auf das Relationale Modell

Kontrollfragen

- Welche Schritte umfasst der Datenbankentwurfsprozess?
- Welche Forderungen müssen die Abbildungen (Transformationen) zwischen den einzelnen Entwurfsschritten erfüllen? Warum?
- Wie werden die Konzepte des ER-Modells auf die des Relationenmodell abgebildet?
- Wie werden die verschiedenen Kardinalitäten von Beziehungstypen bei der Abbildung berücksichtigt?

