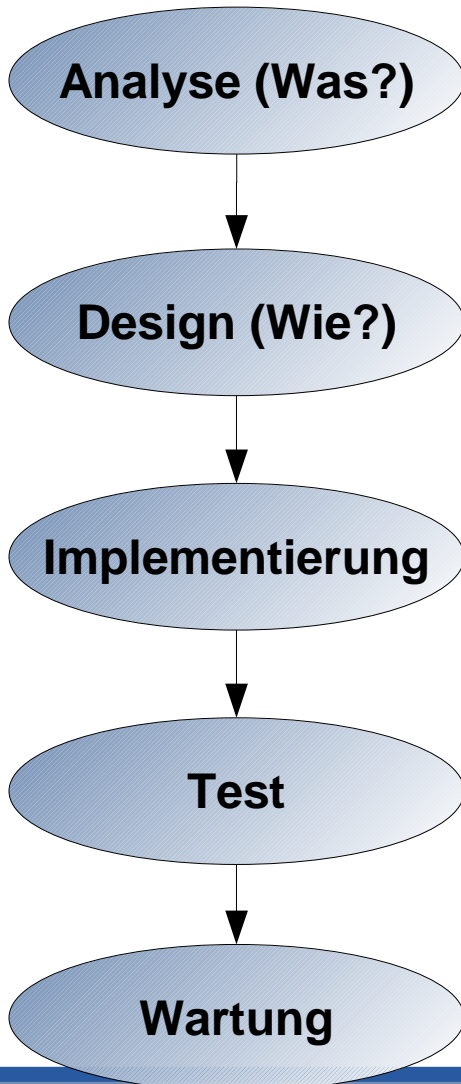


SQL (Structured Query Language)

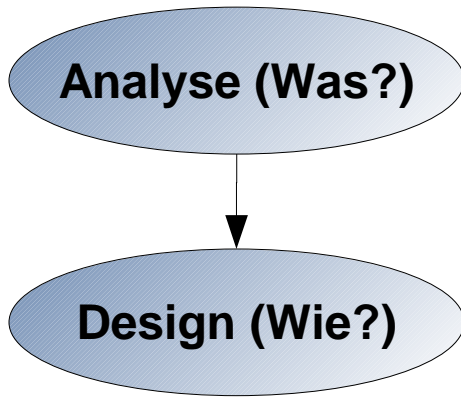
**Datenbanken planen
Beziehungen zwischen Tabellen**



Analyse (Was?)

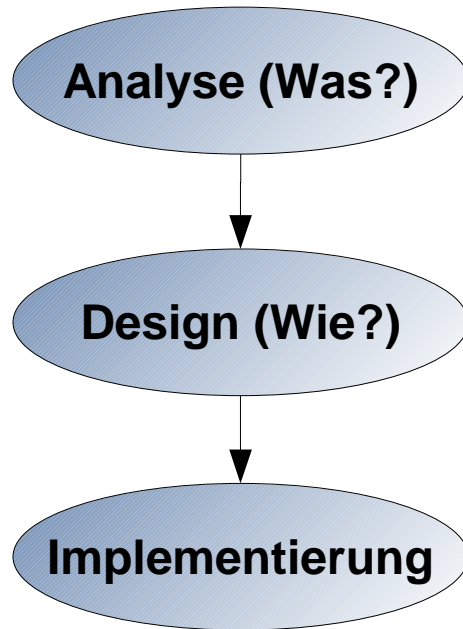
Analyse

- Welche Aufgaben soll die Datenbank erfüllen?
- Welche Arbeitsprozesse müssen abgebildet werden?
- Welche Anforderungen werden an die Datenbank gestellt?
- Festlegung der Benutzergruppen.
- Welche Randbedingungen und Qualitätsmerkmale müssen erfüllt werden?
- ➔ Erstellung eines Lastenheftes, Pflichtenheftes etc.



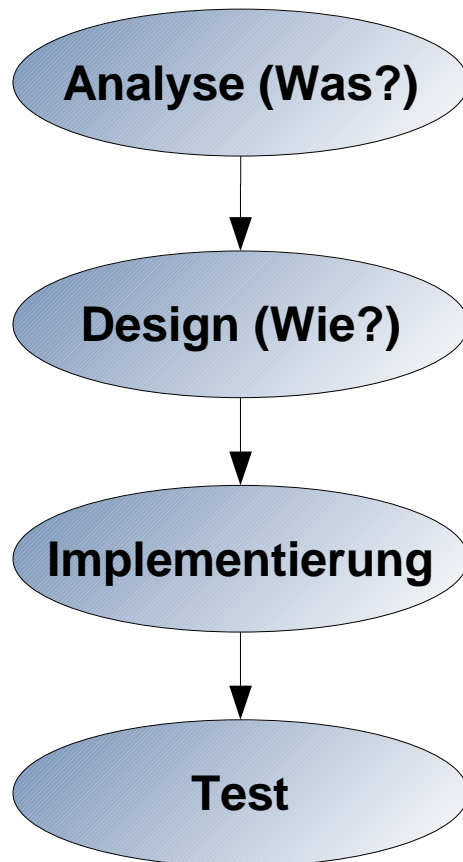
Design

- Konstruktion eines Modells.
- Welche Benutzeroberflächen werden benötigt?
- Welche Daten werden nach welchen Kriterien zusammengefasst?
- Welche Fehler können auftreten?
- Schnittstellen zu anderen Systemen definieren.
- ➔ Nutzung von Normalformen.
- ➔ Entity-Relationshipship-Modell, um Beziehungen zu verdeutlichen.



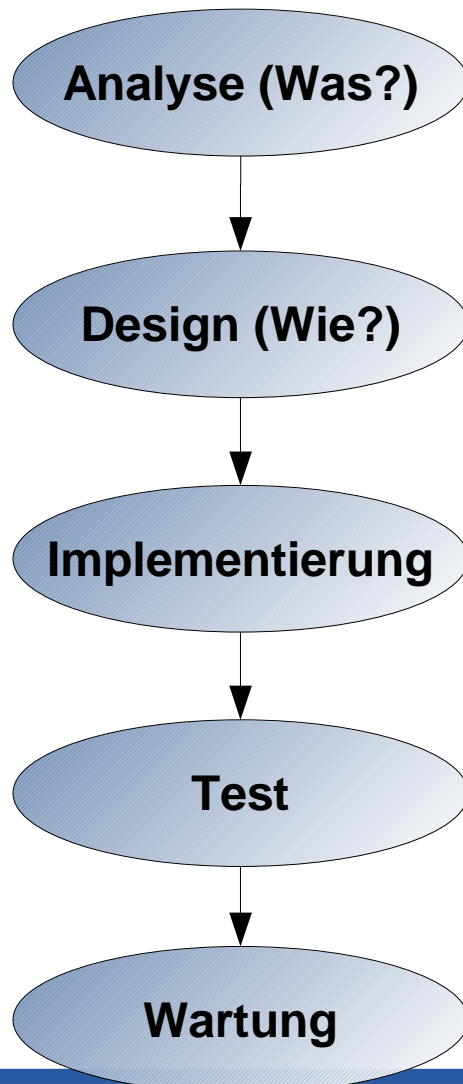
Implementierung

- Umsetzung des Modells in eine konkrete Datenbank.
- Welche Software erfüllt meine vorgegebenen Bedingungen?



Test

- Erfüllt die erstellte Datenbank alle Bedingungen?
- Arbeitsabläufe testen.
- Mögliche Benutzerfehler simulieren.
- Usability testen.
- ➔ Erstellung eines Textberichts zur Dokumentation.
- ➔ Vorhandene Fehler werden dokumentiert und behoben.



Betrieb / Wartung

- Übergabe der Datenbank an den Kunden.
- Mitarbeiter werden geschult.
- Anpassung des Systems, falls nötig.
- ➔ Übergabeprotokoll anfertigen.

- Eine schlecht geplante Datenbank kann mehr Arbeit machen als erwünscht!
- Nachträgliche Änderungen an der Datenstruktur und damit an den Aufbau der Tabellen werden auf ein Minimum reduziert.
- Datenredundanzen werden vermieden.
- Eine Veränderung der Daten führt bei einer guten Planung nicht zu Dateninkonsistenzen.

- ... beschreiben immer einen Akteur und nicht die Aktion.
- listen die verschiedenen Eigenschaften eines bestimmten Objekts spaltenweise auf.
- Beispiele:
 - In einer Tabelle werden Bestellungen gesammelt, aber nicht die Lieferungen der Waren.
 - In einer Tabelle werden alle Informationen zu Kunden gesammelt, aber nicht die Adressen der Lieferanten.
 - In einer Tabelle werden Informationen zu einer CD wie Titel, Preis Herausgeber; Cover etc. gesammelt.

- Welche Szenarien sollen mit Hilfe der Tabellen abgebildet werden? Welche Akteure werden dazu benötigt? Für jeden Akteur und deren Eigenschaften wird eine Tabelle angelegt.
- Welche Objekte werden in der zugrunde liegenden Projektbeschreibung genutzt?
- Welche Eigenschaften der Objekte werden für die Abbildung der Aufgabe benötigt?
- Wie beeinflussen sich die Objekte untereinander? Welche Beziehungen haben die Objekte zueinander?

- ... kann der Auslöser der Aktion sein (zum Beispiel Kunde am Bankautomat).
- ... kann als Gegenstand an der Aktion beteiligt sein (zum Beispiel Bankautomat).
- ... kann auf die angestoßene Aktion reagieren (zum Beispiel Bank).

- ... werden meist durch Substantive symbolisiert.
 - Das *Konto* wird für den *Inhaber* xyz von der *Bank* a eröffnet.
 - Das *Auto* HI hat den *Besitzer* xyz.
 - *Mitarbeiter* Müller arbeitet an dem *Projekt* Ausschreibung.
 - Die *Bestellung* wird über einen *Laserdrucker* xyz ausgedruckt.
 - *Außendienstmitarbeiter* Meier betreut *Kunden* im *Bezirk* Unterweser.
- Für jedes Substantiv (Objekt) sollten Sie eine Tabelle erstellen.
- Redundante oder für die Aufgabenstellung unnötige Objekte werden nicht implementiert.

- Folgende Projektbeschreibung ist gegeben:
 - Jeder *Getränkeautomat* hat einen bestimmten *Standort*.
 - In Abhängigkeit des Standorts werden verschiedene *Heiß- und Kaltgetränke* angeboten.
 - Jedes *Getränk* wird mit dem Namen und dem Preis angezeigt.
 - Jeder *Getränkeautomat* kann maximal fünf verschiedene *Kaltgetränke* anbieten.
 - Die *Heißgetränke* können noch um Zucker oder Milch ergänzt werden.
 - In gewissen Abständen werden die *Automaten* überprüft.
 - Jeder *Standort* wird durch eine Nummer und eine Anschrift beschrieben.

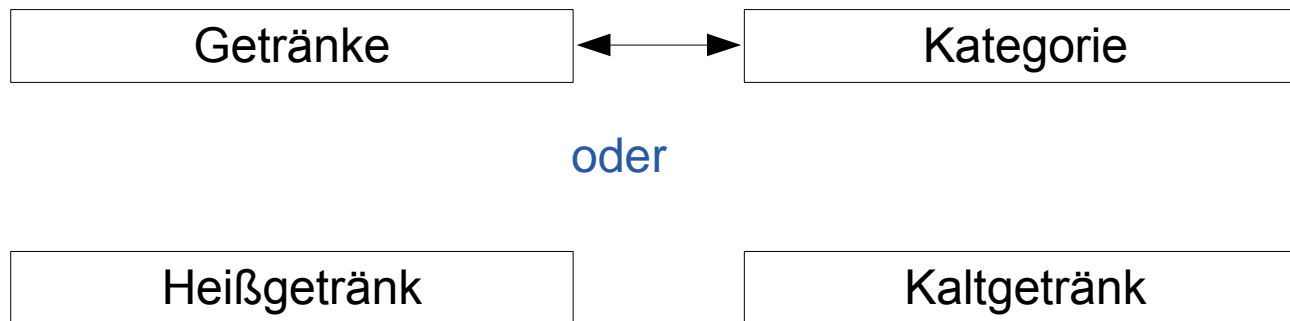
Getränkeautomat

Standort

Getränke

Es werden für die Abbildung des Projekts drei Tabellen benötigt

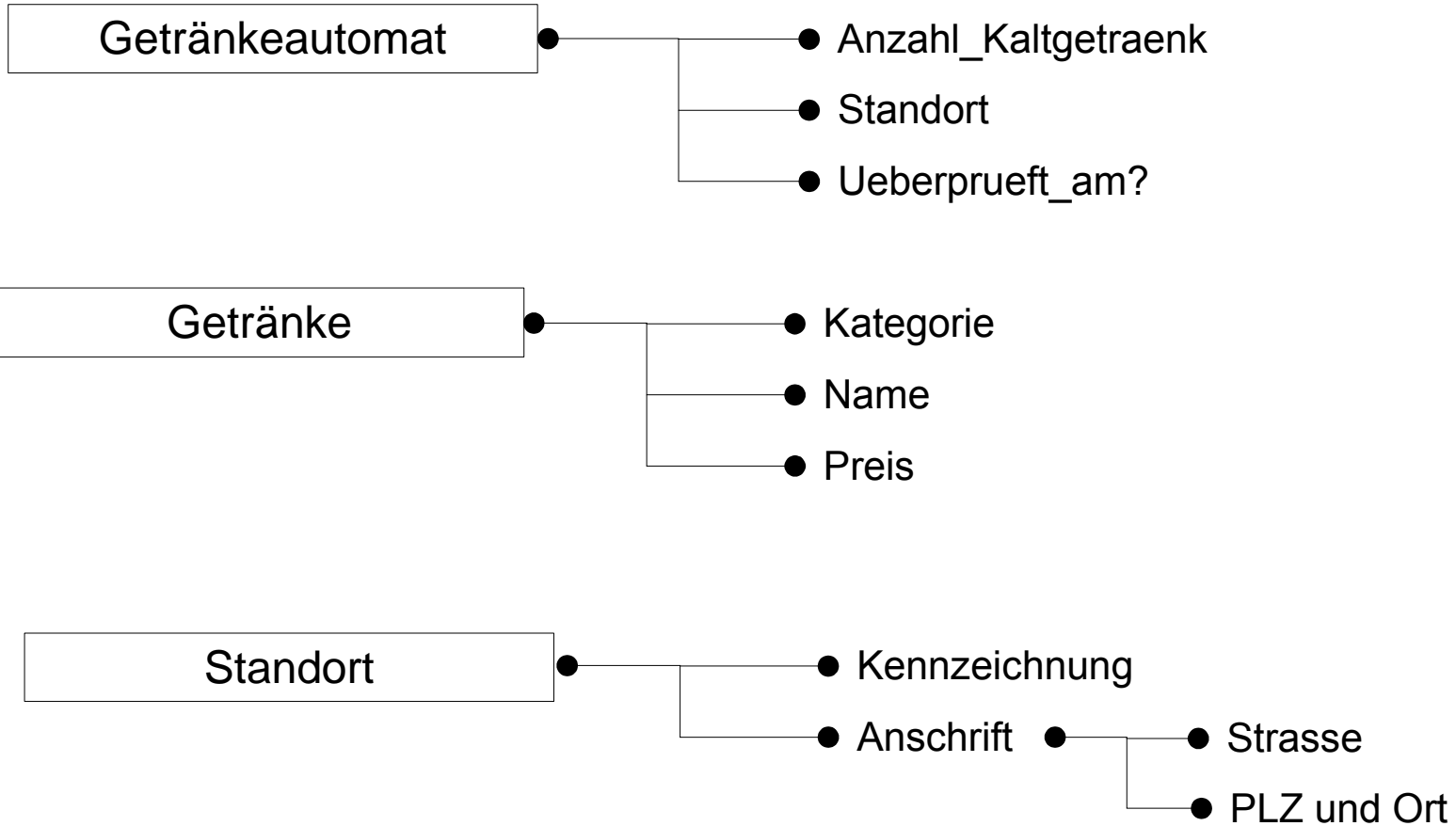
- Folgende Projektbeschreibung ist gegeben:
 - Jeder *Getränkeautomat* hat einen bestimmten *Standort*.
 - In Abhängigkeit des Standorts werden verschiedene *Heiß- und Kaltgetränke* angeboten.
 - Jedes *Getränk* wird mit dem Namen und dem Preis angezeigt.
 - Jeder *Getränkeautomat* kann maximal fünf verschiedene *Kaltgetränke* anbieten.
 - Die *Heißgetränke* können noch um Zucker oder Milch ergänzt werden.
 - In gewissen Abständen werden die *Automaten* überprüft.
 - Jeder *Standort* wird durch eine Nummer und eine Anschrift beschrieben.



- Jedes Objekt in einer Tabelle hat die gleichen Attribute.
 - Pro Tabellenspalte wird immer eine Eigenschaft beschrieben.
 - Jede Tabellenspalte hat eine eindeutige Bezeichnung, die das Attribut widerspiegelt.
- Jedes Objekt unterscheidet sich mindestens von allen anderen Objekte in einem Attributwert .
 - Die Attributwerte in einer Spalte haben das gleiche Format.
 - Das Format wird durch die Art des Wertes oder der Nutzung des Wertes bestimmt.
 - Jedes Format benötigt eine bestimmte Speichergröße.
- Die Informationen sind in jedem Datenfeld atomar. Das heißt die Informationen sind für die Aufgabe in ihre kleinste Einheit aufgebrochen.
- Alle, für die Projektbeschreibung nicht benötigten Attribute werden gestrichen.

- ... entsprechen meist der Beschreibungen eines Objekts.
 - Das Auto ist *rot* und hat einen *KM-Stand* von 130.000.
 - Das Konto wird durch eine *Nummer* identifiziert und durch die *Bankleitzahl* einer Bank zugeordnet.
 - Die Milch ist *pasteurisiert*.
- ... sind Substantive im Genitiv (wessen).
 - Die Menge *des* Artikels.
 - Ausleihdatum *des* Buches.
 - Anmeldung *eines* Autos.
- ... sind Verben, die die Eigenschaft eines Objekts verändern.
 - Wenn der Kunde solvent ist, kann ein Konto *eröffnet* werden.
 - Geld wird vom Konto *abgehoben*.
 - Sobald der Urlaubsantrag genehmigt wurde, wird die *Anzahl* der Urlaubstage *vermindert*.
 - Sobald ein Buch im Lager vorhanden ist, wird es an den Kunden *ausgeliefert*.

- Folgende Projektbeschreibung ist gegeben:
 - Jeder *Getränkeautomat* hat einen bestimmten *Standort*.
 - In Abhängigkeit des Standorts werden verschiedene *Heiß- und Kaltgetränke angeboten*.
 - Jeder *Getränkeautomat* kann maximal *fünf* verschiedene *Kaltgetränke anbieten*.
 - Jedes *Getränk* wird mit dem *Namen* und dem *Preis* angezeigt.
 - Die *Heißgetränke* können noch um Zucker oder Milch ergänzt werden.
 - In gewissen Abständen werden die *Automaten überprüft*.
 - Jeder *Standort* wird durch eine *Nummer* und eine *Anschrift* beschrieben.



- ... beschreiben die Interaktion zwischen Objekten.
- ... beschreiben verschiedene Arten ein und des selben Objekts.
 - Ein Buch kann in einem Festeinband oder als Taschenbuch geliefert werden.
 - Joghurt kann in 125-g-Bechern oder 500-g-Gläsern geliefert werden.
- ... verknüpfen zwei oder mehr Tabellen. Beispiel:
 - In einer Tabelle werden alle Kunden aufgelistet.
 - In einer anderen Tabelle werden alle Bestellungen aufgelistet. Der Kunde kann mehr als eine Bestellung aufgeben.

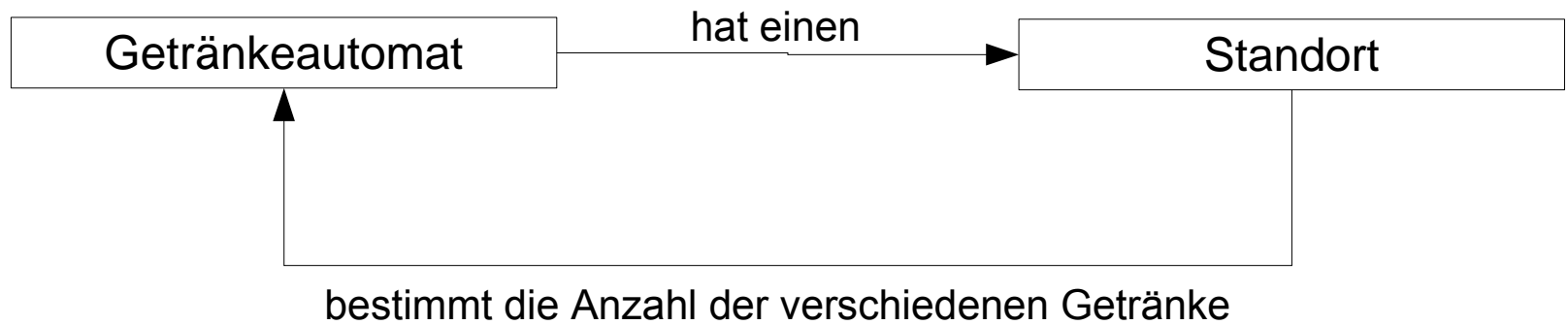
- Durch die Verknüpfung von Tabellen
 - ... werden die Informationen nur einmal in einer Tabelle in einer Datenbank abgelegt.
 - ... werden die gleichen Informationen nicht mehrmals in einer Datenbank abgelegt.
- Rechtschreibfehler werden vermieden.
- Änderungen können leichter durchgeführt werden, weil sie nur an einer Stelle vorgenommen werden müssen.

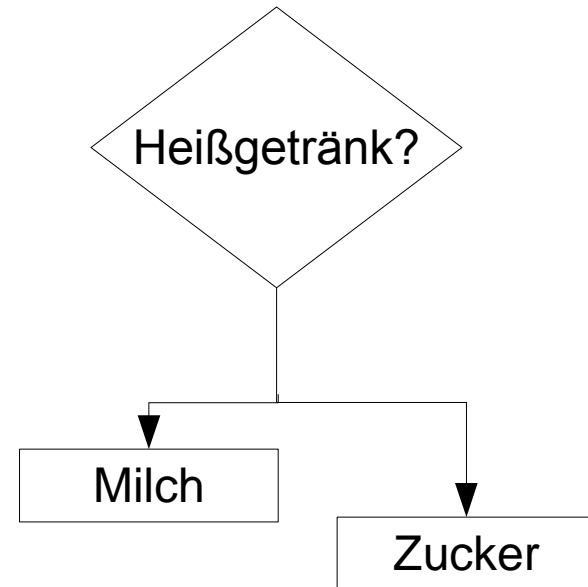
- Kundenadressen werden für einen Lieferschein und den Bestellschein benötigt.
 - Alle Kundenadressen werden in einer Tabelle gesammelt.
 - Jeder Kunde hat einen eindeutigen Schlüssel (Kundennummer), die ihn eindeutig identifiziert.
 - Dieser Schlüssel wird in der Tabelle aller Bestellungen sowie aller Lieferungen genutzt.
- Ein Buchautor hat viele verschiedene Bücher geschrieben.
 - Jeder Autorenname wird einmal in einer Tabelle gespeichert und über ein Schlüssel mit seinen Büchern verknüpft.
- Auf einer CD kommen viele Lieder vor.
 - Einige Lieder kommen auf mehreren CDs vor.
 - Die Informationen zu Liedern und CDs werden in zwei Tabellen gespeichert, die in einer dritten Tabelle über Schlüssel miteinander verknüpft werden.

- Verben, die ein anderes Substantiv beeinflussen, sind ein Hinweis auf eine Beziehung.
- Präpositionen die in Bezug zu einem Substantiven stehen. Präpositionen können eine Beziehung beschreiben.
- Substantive, die einen Rückbezug auf ein anderes Objekt besitzen, haben eine Beziehung zu einem anderen Objekt.
- Substantive wie zum Beispiel „hat“ geben einen Hinweis auf eine Beziehung.

- *Kunde bestellt Bücher.*
- Der *Auftrag* wird *an* das *Lager* gegeben. Entsprechend des Auftrags werden die *Bücher ausgeliefert.*
- Der *KFZ-Halter meldet* ein *Fahrzeug an* der *KFZ-Stelle* an.
- Ein *Buch* besteht *aus* einer bestimmten Anzahl von *Seiten.*
- Der *Kontoinhaber des Kontos xyz* ist kreditwürdig.
- Der *Besitzer hat* ein *Auto*. Aber: Das *Auto hat* vier Räder beschreibt eine Eigenschaft.

- Folgende Projektbeschreibung ist gegeben:
 - Jeder *Getränkeautomat* hat einen bestimmten *Standort*.
 - In *Abhängigkeit des Standorts* werden verschiedene *Heiß- und Kaltgetränke angeboten*.
 - Jeder *Getränkeautomat* kann maximal *fünf* verschiedene *Kaltgetränke anbieten*.
 - Jedes *Getränk* wird mit dem *Namen* und dem *Preis* angezeigt.
 - Die *Heißgetränke* können noch um *Zucker oder Milch ergänzt* werden.
 - In gewissen Abständen werden die *Automaten überprüft*.
 - Jeder *Standort* wird durch eine *Nummer* und eine *Anschrift* beschrieben.





- ... werden mit Hilfe von eindeutigen Werten dargestellt.
- Die Tabellen werden mit Hilfe von Schlüsselwerten verknüpft.
- In einer Tabelle existiert ein Schlüssel, der eindeutig einen Datensatz identifiziert. Dieser Schlüssel wird als Primärschlüssel bezeichnet. Dieser Schlüssel wird als Fremdschlüssel in einer anderen Tabelle genutzt. Der Fremdschlüssel stellt eine Verknüpfung zu einem bestimmten Datensatz in einer anderen Tabelle her.

Automat	Standort	Heißgetränk	Kaltgetränk	ÜberprüftAm
B2304.34	S1	Ja	Ja	
C2305.35	S2	Ja	Nein	

ID	Getränk	Kategorie
34	Kaffee	Heiss
35	Tee	Heiss
36	Limo	Kalt
37	Wasser	Kalt

Automat	Getränk
B2304.34	34
B2304.34	36
C2305.35	37

IAmtNr	Vorname	Nachname	Straße	PLZ	Wohnort
123	Heinz	Peters	Friesenstr. 4	xxxxx	Hannover
456	Sonja	Müller	Eichenstr. 5	xxxxx	Hannover
753	Marlis	Schulze	Lerchenstr. 2	xxxxx	Mellendorf
789		Scholz		xxxxx	Hildesheim

Der Primärschlüssel wird ...

Mastertabelle

Student	Algorithmen 1	Theoretische Inf.	Algebra 1
123	1	2	1
456	3		4
789		2	

Der Student kann eindeutig über das Schlüsselfeld identifiziert werden.

... in dieser Tabelle als Fremdschlüssel genutzt.

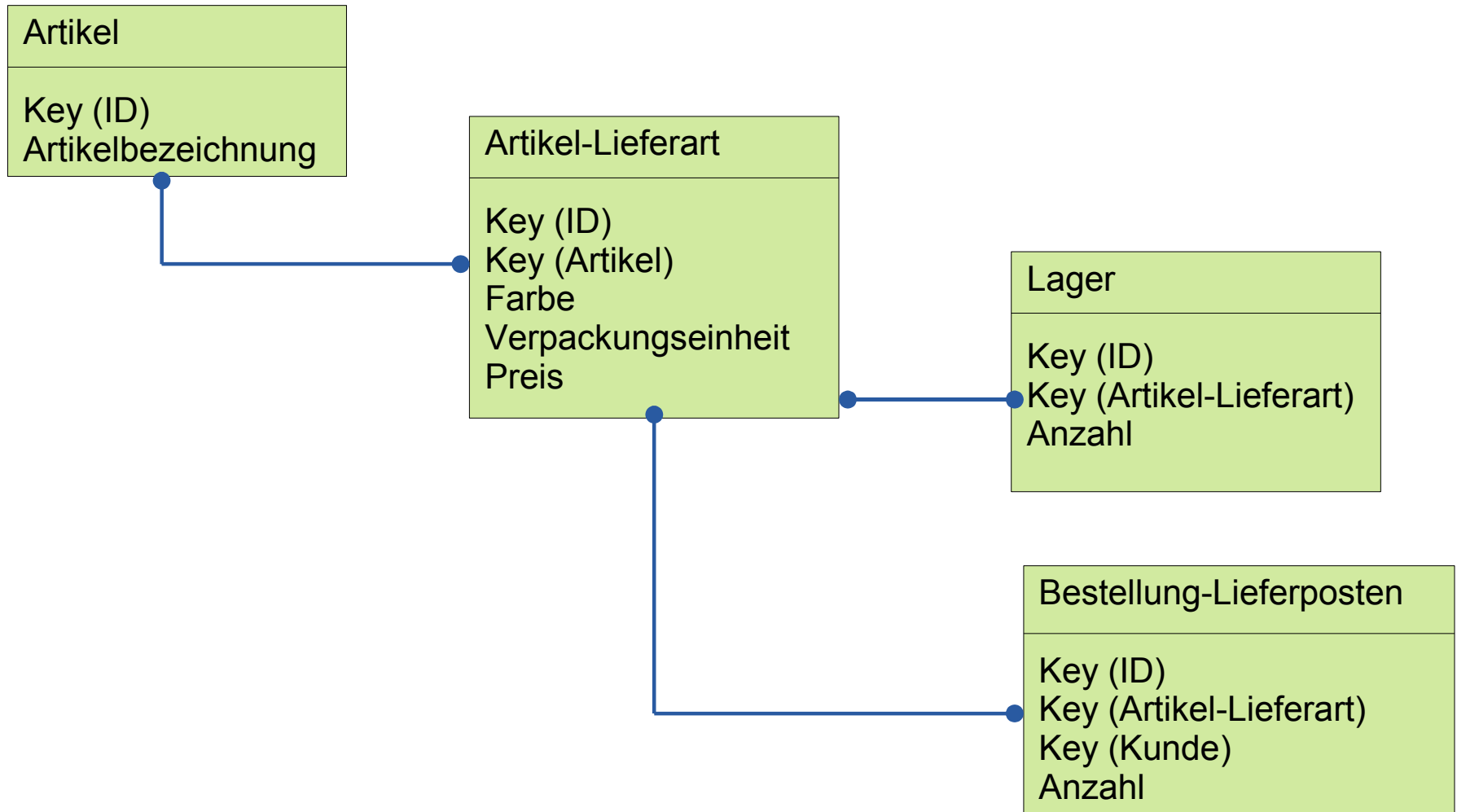
Detailtabelle

■ Die Mastertabelle

- ... enthält alle Hauptdatensätze.
- ... besitzt einen Primärschlüssel, der die Datensätze eindeutig identifiziert.
- ... stehen in der Hierarchie oben. Unterhalb von ihnen befinden sich andere Tabellen.

■ Die Detailtabelle

- ... beschreibt Besonderheiten oder Eigenschaften eines Elements.
- ... bildet eine Beziehung zwischen Objektbeschreibungen ab.
- ... enthalten Fremdschlüssel, die aus einer anderen Tabelle kommen.
- ... enthalten aber selber ein Primärschlüssel, die die Datensätze eindeutig identifiziert.
- ... sind in der Hierarchie anderen Tabellen untergeordnet.



- ... werden in einer Mastertabelle erzeugt.
- ... müssen eindeutig sein. Es existieren keine Duplikate von diesem Wert.
- ... werden in anderen Tabellen für die Identifizierung eines bestimmten Datensatzes in einer anderen Tabelle genutzt.
- ... kann aus mehreren Attributen zusammengesetzt werden.
- ... sind häufig künstliche Attribute wie Personalnummer, Kundennummer etc. Das heißt: Es sind keine sichtbaren Eigenschaften eines Objekts.

- Ein Auto hat Räder, Lenkrad, Farbe, etc. Die Fahrzeugnummer ist nicht wichtig für den Gebrauch eines Autos und beschreibt auch keine Eigenschaften. Die Fahrzeugnummer identifiziert aber ein Auto.
- Ein Buch kann über die ISBN-Nummer eindeutig bestellt werden. Ein Autor kann mehr als ein Buch geschrieben haben. Ein Titel kann in verschiedenen Versionen und Arten erscheinen.
- Ein Mitarbeiter einer Firma ist eindeutig über seine Personalnummer identifizierbar.

Kunde

ID	Firma	Straße	Hausnummer	Ort
1	RRZN	Schloßwender Str.	5	Hannover

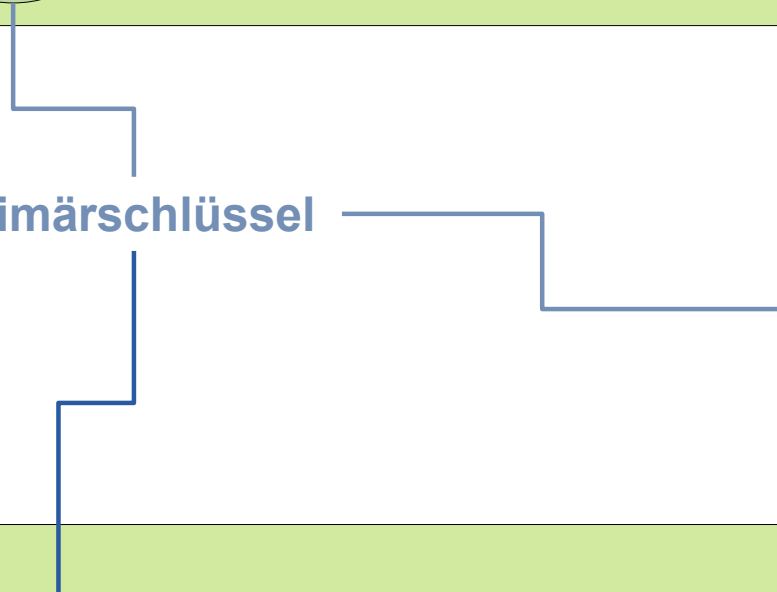
Lagerware

ID	Artikelname	Beschreibung
1	Druckerpatrone	schwarz
2	CD	Read / Write

Bestellung

ID	Firma	Anzahl	Ware	BestelltAm
1	1	5	1	15.01.07
2	1	10	2	16.01.07

Primärschlüssel



```
CREATE TABLE mitarbeiter  
(  
  
    mitarbeiter_nachname VARCHAR(50),  
    mitarbeiter_vorname VARCHAR(20),  
    mitarbeiter_eintritt DATE  
  
);
```

- ... dürfen nicht undefiniert (NULL) sein.
- ... müssen eine Eingabe haben.
- ... dürfen nicht geändert werden. Das Feld muss mit einem Schreibschutz versehen werden.
- ... sollte zu klein wie möglich und so groß wie nötig gewählt werden

```
CREATE TABLE mitarbeiter
(
  mitarbeiter_ID INTEGER NOT NULL
  mitarbeiter_nachname VARCHAR(50) NOT NULL,
  mitarbeiter_vorname VARCHAR(20),
  mitarbeiter_eintritt DATE NOT NULL,

  PRIMARY KEY(mitarbeiter_ID)
);
```

- Die Syntax der Definition ist abhängig von der gewählten SQL-Implementierung.
- Zuerst werden die Spalten definiert. Einige Spalten erzwingen eine Eingabe.
- Mit Hilfe von `PRIMARY KEY` wird eine definierte Spalte als Primärschlüssel gekennzeichnet.

```
CREATE TABLE mitarbeiter
(
  mitarbeiter_ID INTEGER NOT NULL AUTO_INCREMENT,
  mitarbeiter_nachname VARCHAR(50) NOT NULL,
  mitarbeiter_vorname VARCHAR(20),
  mitarbeiter_eintritt DATE NOT NULL,

  PRIMARY KEY(mitarbeiter_ID)
);
```

- Mit Hilfe des Befehls `AUTO_INCREMENT` wird ein Wert erzeugt, der bei eins beginnt und für jede neue Zeile automatisch um eins erhöht wird.
- ... sind nur mit `INTEGER`-Werten möglich.
- `AUTO_INCREMENT`-Werte gibt es pro Tabelle nur einmal.
- Der Befehl ist abhängig von der gewählten SQL-Implementierung.

- ... kommen in einer Detailtabelle vor.
- ... sind in einer Mastertabelle Primärschlüssel.
- Jeder Fremdschlüssel muss eine eindeutige Entsprechung in der Mastertabelle haben.
- Der Fremdschlüssel in der Detailtabelle besitzt den gleichen Datentyp wie der dazugehörige Primärschlüssel in der Mastertabelle.
- ... können einmal, keinmal oder mehrmals in einer Detailtabelle vorkommen.

- Der Kunde bestellt ein Buch. In einer Tabelle werden alle Kunden aufgelistet. In der Tabelle "Bestellungen" werden die Kunden mit Hilfe des Schlüsselwertes eingebunden.
- In einer Tabelle werden die Attribute von Waren in einem Laden allgemein beschrieben. Zur Identifizierung wird die Artikelnummer genutzt. In der Tabelle, die das Warenlager symbolisiert, werden die Artikelnummer sowie die Verpackungseinheit angegeben. Eine Ware kann im Lager in verschiedenen Verpackungseinheiten vorkommen.
- In einer Tabelle werden alle Mitarbeiter einer Firma aufgelistet. In einer weiteren Tabelle werden alle Projekte aufgelistet. Jedes Projekt hat eine Projektnummer. Der Projektleiter wird durch eine Personalnummer identifiziert, die einen Mitarbeiter aus der entsprechenden Tabelle identifiziert. Des weiteren werden Projektnummer und Personalnummer in einer weiteren Tabelle genutzt, um die Mitarbeiter eines Projektes zu identifizieren.

Kunde

ID	Firma	Straße	Hausnummer	Ort
1	RRZN	Schloßwender Str.	5	Hannover

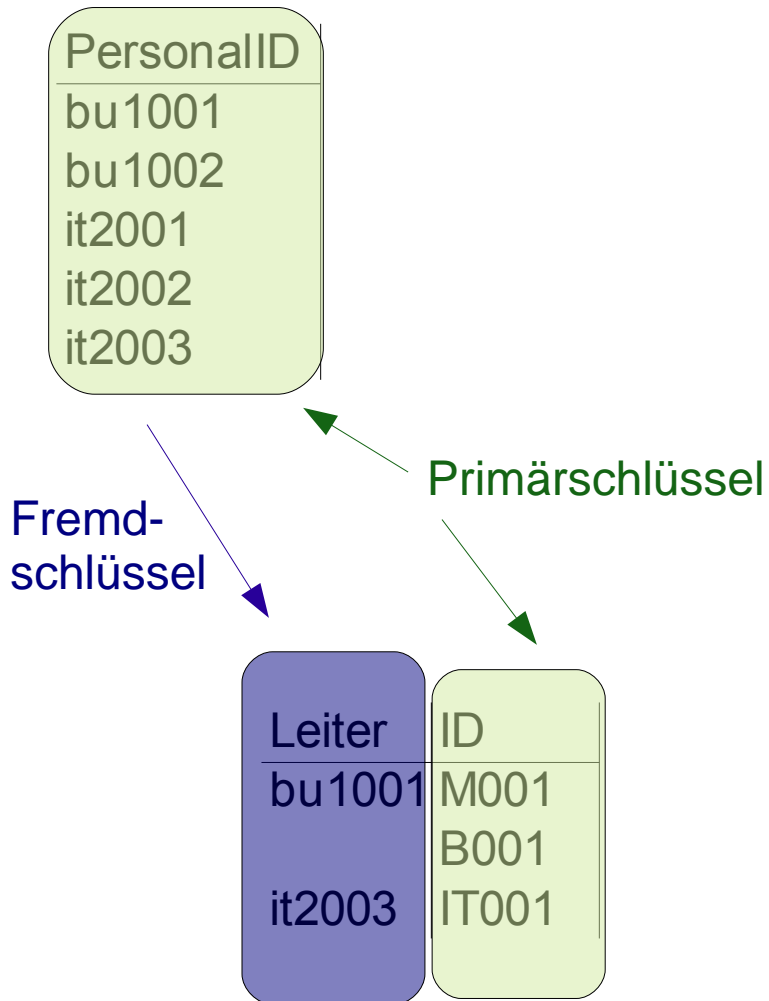
Lagerware

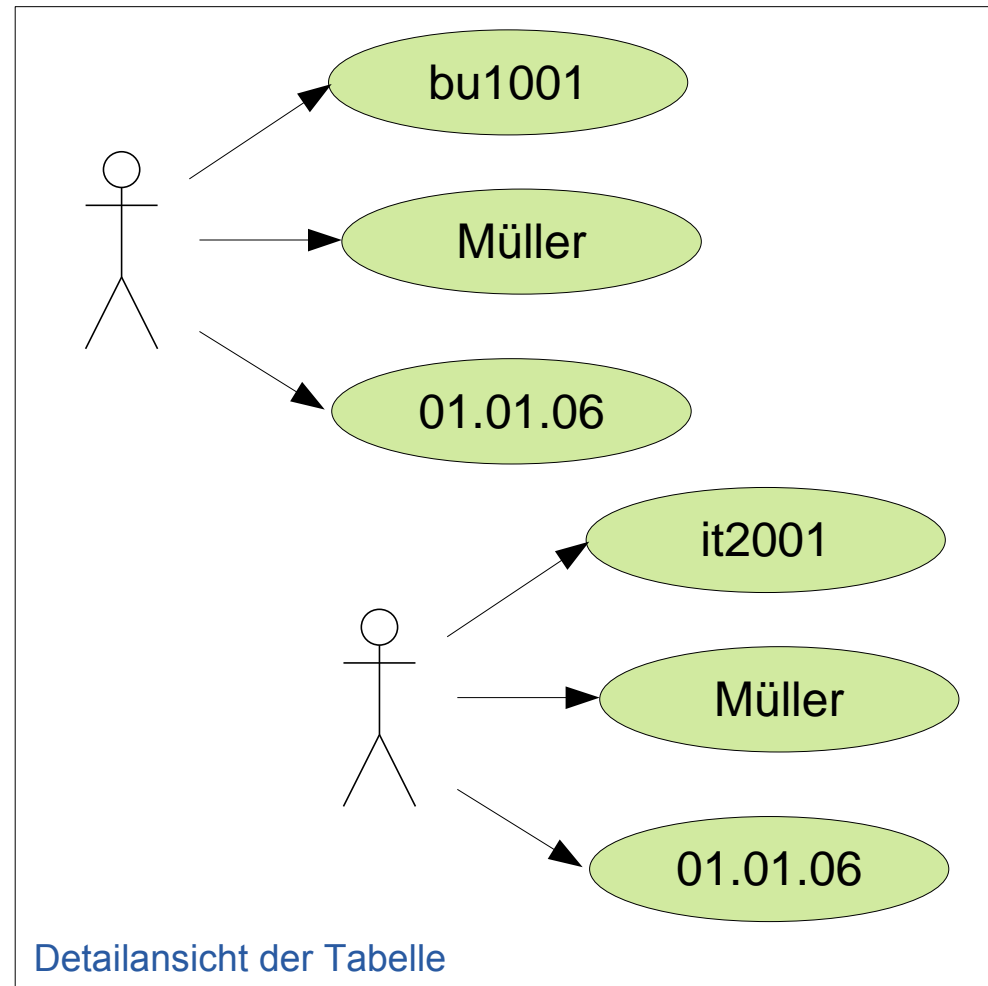
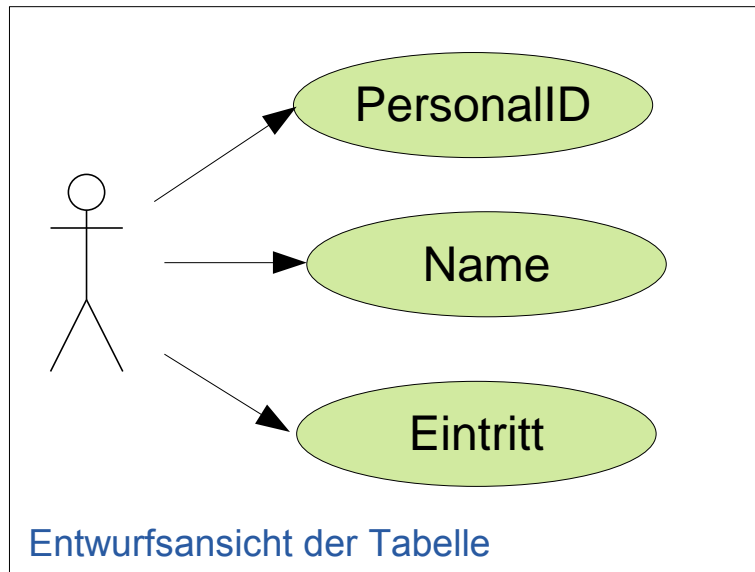
ID	Artikelname	Beschreibung
1	Druckerpatrone	schwarz
2	CD	Read / Write

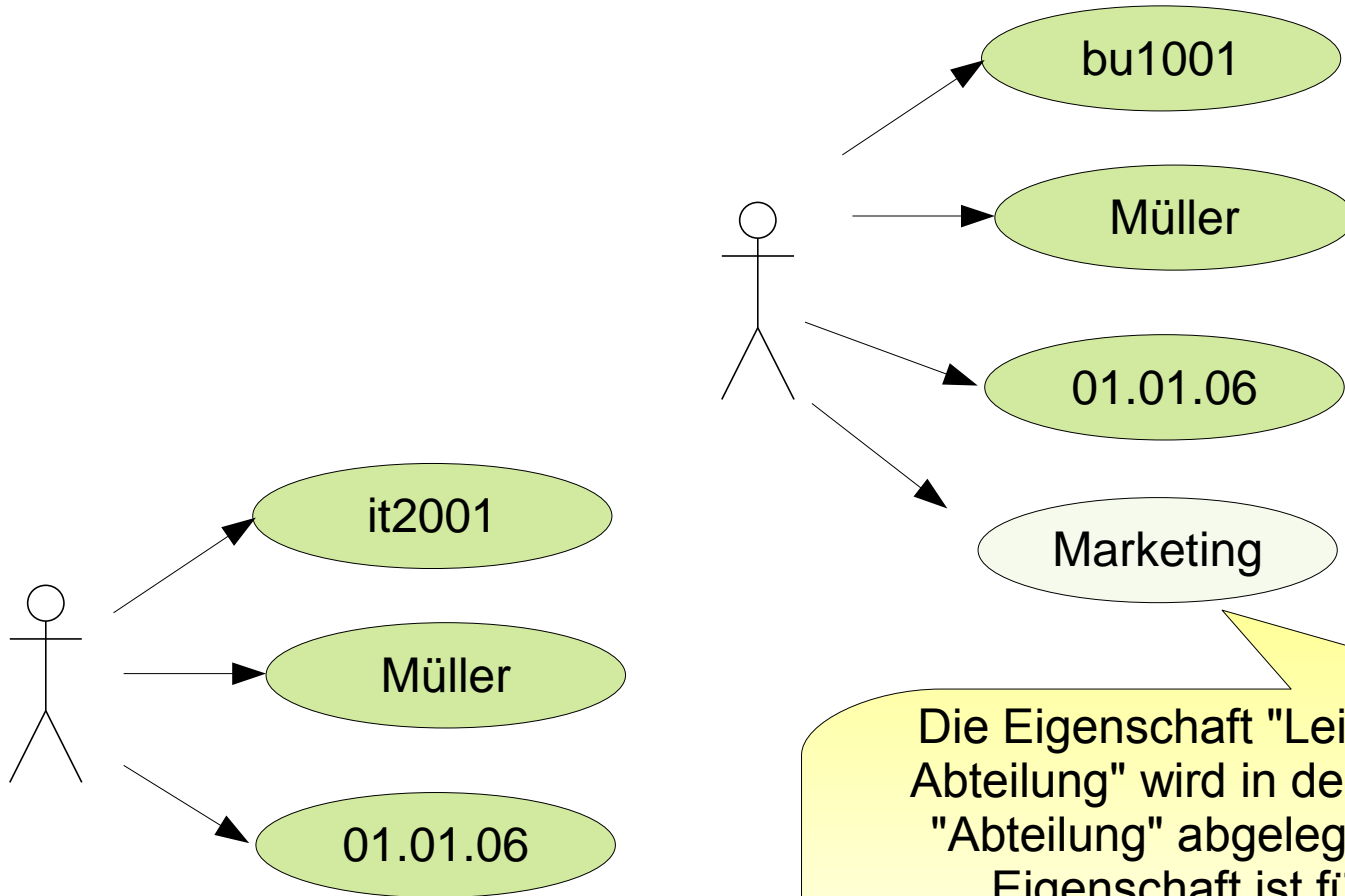
Fremdschlüssel

Bestellung

ID	Firma	Ware	Anzahl	BestelltAm
1	1	1	5	15.01.07
2	1	2	10	16.01.07







Die Eigenschaft "Leiter einer Abteilung" wird in der Tabelle "Abteilung" abgelegt. Diese Eigenschaft ist für die Beschreibung eines Mitarbeiters nicht relevant, aber für die Beschreibung einer Abteilung.

```
CREATE TABLE projektliste
(
  projektnummer CHAR(12) NOT NULL,
  projektname VARCHAR(50) NOT NULL,
  auftraggeber CHAR(10) NOT NULL,
  projektleiter INT NOT NULL,

  PRIMARY KEY(projektnummer),

  FOREIGN KEY(auftraggeber)
             REFERENCES kunden(kundenr),
  FOREIGN KEY(projektleiter) REFERENCES
             mitarbeiter(mitarbeiter_ID)
);
```

- Nicht alle SQL-Implementierungen nutzen Fremdschlüssel.
- MySQL achtet zum Beispiel nicht auf die Integrität der Daten.

- Jede Information wird nur an einer Stelle in der Datenbank gespeichert.
- Vermeiden sie redundante Daten.
- Beachten Sie die referentielle Integrität:
 - Jeder Datensatz in der Detailtabelle benötigt einen gültigen Fremdschlüssel aus der Mastertabelle.
 - Bevor Sie ein Primärschlüssel aus der Mastertabelle löschen, müssen alle dazugehörigen Datensätze aus der Detailtabelle gelöscht werden.

```
CREATE TABLE abteilung
(
  abteilung_ID VARCHAR(10) NOT NULL
  abteilung_name VARCHAR(50) NOT NULL,
  mitarbeiter INTEGER NOT NULL,

  PRIMARY KEY(abteilung_ID),

  UNIQUE(abteilung_name)
);
```

- Werte, die nicht für die Identifizierung des Datensatzes benötigt werden, aber eindeutig sein sollen, werden mit `UNIQUE` bezeichnet.
- `UNIQUE(datenfeld, datenfeld)`. Die Elemente in einer Datenfeldliste werden immer durch Kommata getrennt.
- Die Datenfelder werden teilweise als Sekundärschlüssel bezeichnet.

abteilung_ID	abteilung_name	mitarbeiter
<i>buch_12</i>	<i>Buchhaltung</i>	<i>Müller</i>
<i>itSoft_13</i>	<i>Software</i>	<i>Schulz</i>
<i>itHard_14</i>	<i>Hardware</i>	
<i>buch_12</i>	<i>Marketing</i>	<i>Maier</i>
<i>itSoft_14</i>	<i>Software</i>	<i>Schmidt</i>

Fehler: Felder dürfen keinen undefinierten Inhalt besitzen.

Fehler: Der Primärschlüssel kommt doppelt vor.

Fehler: Aufgrund der Unique-Anweisung darf ein Abteilungsname nur einmal vorkommen. Damit hat jede Abteilung auch nur einen Abteilungsleiter.

- ... beschleunigen die Suche in großen Datenbeständen.
 - Wenn kein Index vorhanden ist, wird die Datenbank sequentiell vom ersten bis zum letzten Datensatz durchsucht.
 - Falls ein Index vorhanden ist, wird das Suchwort mit den Wörtern im Index verglichen und direkt zu dem dazugehörigen Datensatz gesprungen.
- ... bilden ein Stichwortverzeichnis. Zu jedem Stichwort ist ein Verweis auf einen bestimmten Datensatz vorhanden.
- Primär- und Fremdschlüsselfelder werden meist automatisch indiziert.

```
CREATE TABLE projektliste
(
  projektnummer CHAR(12) NOT NULL,
  projektname VARCHAR(50) NOT NULL,
  auftraggeber CHAR(10) NOT NULL,
  projektleiter INT NOT NULL,

  PRIMARY KEY(projektnummer),

  FOREIGN KEY(auftraggeber)
    REFERENCES kunden(kundenr),
  FOREIGN KEY(projektleiter) REFERENCES
    mitarbeiter(mitarbeiter_ID)
);

CREATE INDEX index_projekt
  ON projektliste(projektname, auftraggeber);
```

Es wird ein Index über n Datenfelder einer Tabelle erzeugt. Der Index ist aufsteigend sortiert.

```
CREATE INDEX indexname  
ON tabelle(datenfeld, datenfeld, ...);
```

```
CREATE DESC INDEX indexname  
ON tabelle(datenfeld, datenfeld, ...);
```

```
DROP INDEX indexname ON tabelle;
```

```
SHOW INDEX,
```

Es wird ein Index über n Datenfelder einer Tabelle erzeugt. Der Index ist absteigend sortiert.

Alle definierten Indizes einer Datenbank werden angezeigt.

Der Index einer Tabelle wird gelöscht.

- Falls Daten in der Tabelle vorhanden sind, sollten am Entwurf von bestehenden Spalten keine Änderungen vorgenommen werden. Es können aber ohne Probleme neue Spalten hinzugefügt werden. Ein Löschen von Spalten führt zu einem Datenverlust und kann nicht rückgängig gemacht werden.
- Die Feldgröße eines Feldes kann ohne Probleme vergrößert werden. Wenn Daten vorhanden sind, sollte die Größe eines Feldes aber nie minimiert werden.
- Der Feldname einer Spalte darf nicht geändert werden. Es kann zu einer Inkonsistenz von Daten kommen.
- Der Typ eines Feldes sollte nicht verändert werden. Nach einer Änderung können eventuell vorhandene Daten nicht mehr dargestellt werden und werden dadurch entfernt.

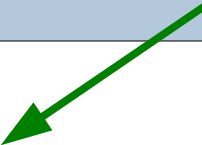
```
ALTER TABLE old_name  
    RENAME TO new_name;
```

- Mit Hilfe der ALTER-Anweisungen kann die Tabellenstruktur verändert werden.
- In diesem Beispiel wird eine Tabelle umbenannt.
- ALTER-Anweisungen sind nicht in allen SQL-Implementierungen vorhanden.

```
ALTER TABLE tabellename  
    ADD COLUMN spalte datentyp NOT NULL FIRST;
```

```
CREATE TABLE mitarbeiter  
(  
    mitarbeiter_nachname VARCHAR(50),  
    mitarbeiter_vorname VARCHAR(20),  
    mitarbeiter_eintritt DATE  
);  
  
ALTER TABLE mitarbeiter  
    ADD COLUMN mitarbeiter_ID INTEGER NOT NULL;
```

Die Spalte wird am Ende der Tabelle
angefügt.



Die Spalte wird als erstes angezeigt.
Das Argument ist optional.

```
CREATE TABLE mitarbeiter
(
  mitarbeiter_nachname VARCHAR(50),
  mitarbeiter_vorname VARCHAR(20),
  mitarbeiter_eintritt DATE
);

ALTER TABLE mitarbeiter
  ADD COLUMN mitarbeiter_ID INTEGER NOT NULL FIRST;

ALTER TABLE mitarbeiter
  ADD COLUMN geburt DATE AFTER mitarbeiter_vorname;
```

Die Spalte wird nach einer bestimmten
Spalte eingefügt.

```
ALTER TABLE tabellenname  
    ADD COLUMN spalte datentyp NOT NULL FIRST;
```

```
CREATE TABLE mitarbeiter  
(  
    mitarbeiter_nachname VARCHAR(50),  
    mitarbeiter_vorname VARCHAR(20),  
    mitarbeiter_eintritt DATE  
);  
  
ALTER TABLE mitarbeiter  
    ADD COLUMN mitarbeiter_ID INTEGER NOT NULL FIRST  
    ADD PRIMARY KEY (mitarbeiter_ID);
```

```
ALTER TABLE tabelle
    CHANGE COLUMN oldname newname datentyp;

ALTER TABLE mitarbeiter
    CHANGE COLUMN nr mitarbeiter_ID CHAR(10) NOT NULL;

ALTER TABLE tabelle
    MODIFY COLUMN newdatentyp;

ALTER TABLE mitarbeiter
    MODIFY COLUMN mitarbeiter_ID CHAR(10) NOT NULL;
```

- CHANGE COLUMN ändert den Spaltennamen sowie den Datentyp.
- MODIFY COLUMN ändert den Datentyp.

```
ALTER TABLE old_name  
    DROP Column spalte;
```

- Es wird die Spalte und alle darin enthaltenen Daten gelöscht.

- Als Ausgangspunkt wird eine Datensammlung in Tabellenform genutzt.
- Mit Hilfe der Normalisierung werden große Tabellen in viele kleine Tabellen aufgeteilt.
- Durch die Normalisierung
 - ... ist jedes Datenfeld nur einmal vorhanden.
 - ... bestehen zwischen den verschiedenen Spalten keine Abhängigkeiten.
 - ... ist jeder Datensatz eindeutig mit Hilfe von bestimmten Schlüsseln identifizierbar.
 - ... werden Redundanzen entfernt.
 - bleibt die Datenbank konsistent. Durch das Löschen oder Ändern von Datensätzen werden keine fehlerbehafteten oder widersprüchlichen Daten erzeugt.



Autor	Titel	ISBN	AuslieferungAls	Preis
Kai Meyer	Seide und Schwert		Hardcover	



Autor	Titel	ISBN	AuslieferungAls	Preis
Kai Meyer	Seide und Schwert		Hardcover	
Kai Meyer	Lanze und Licht		Taschenbuch	
Kai Meyer	Lanze und Licht		Hörbuch	

ID	Buch	Autor
B1	Seide und Schwert	1
B2	Lanze und Licht	1

ID	Vorname	Nachname
1	Kai	Meyer
2	Tad	Williams

Buch	Auslieferung
B1	Hard
B2	Taschen
B2	Hoer

ID	
Hard	Hardcover
Taschen	Taschenbuch
Hoer	Hörbuch

- Alle Informationen zu einer CD, Buch etc. sind in einer Tabelle aufgelistet.
- Einige Attribute enthalten Wertelisten.
- Nachteile:
 - Das Bearbeiten der Daten ist nur mit einem hohen Aufwand möglich.
 - Die Felder können nicht in Abhängigkeit von bestimmten Kriterien durchsucht werden.
 - Es muss Speicherplatz für eine unbekannte Menge an Daten freigehalten werden.
 - Daten werden doppelt abgelegt. Aus diesem Grund ist es schwierig Änderungen vorzunehmen.

Ernst Müller
Schmiedestr. 4
31157 Sarstedt

Firma Sonnenschein
Flugweg 1
30159 Hannover

Bestelldatum

Bestellnummer

Artikelnummer	Artikel	Einheit	Preis	Menge	Gesamt
---------------	---------	---------	-------	-------	--------

Netto

Mehrwertsteuer

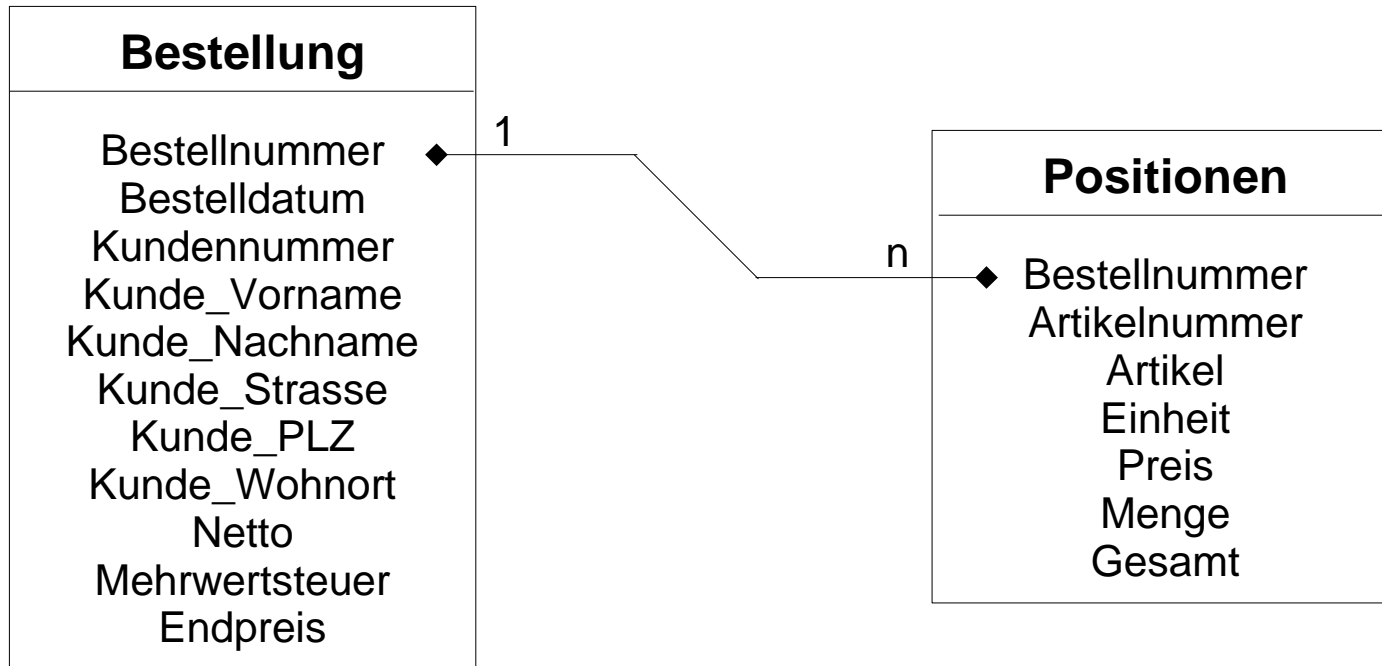
Endpreis

Folgende Datenfelder werden benötigt

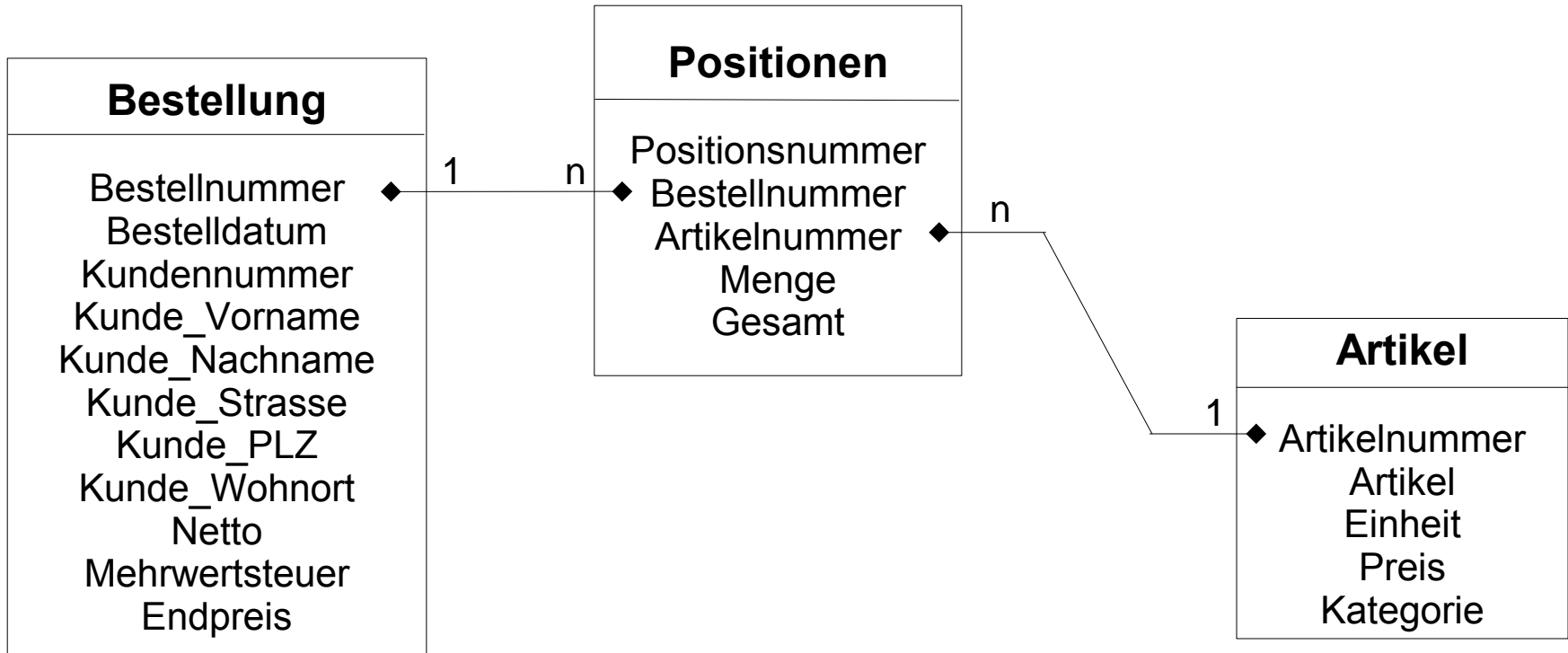
Bestellung

Bestellnummer
Bestelldatum
Kundennummer
Kundenname
Kundenadresse
Artikelnummer01
Artikel01
Einheit01
Preis01
Menge01
Gesamt01
Artikelnummer02
Artikel02
Einheit02
Preis02
Menge02
Gesamt02
Netto
Mehrwertsteuer
Endpreis

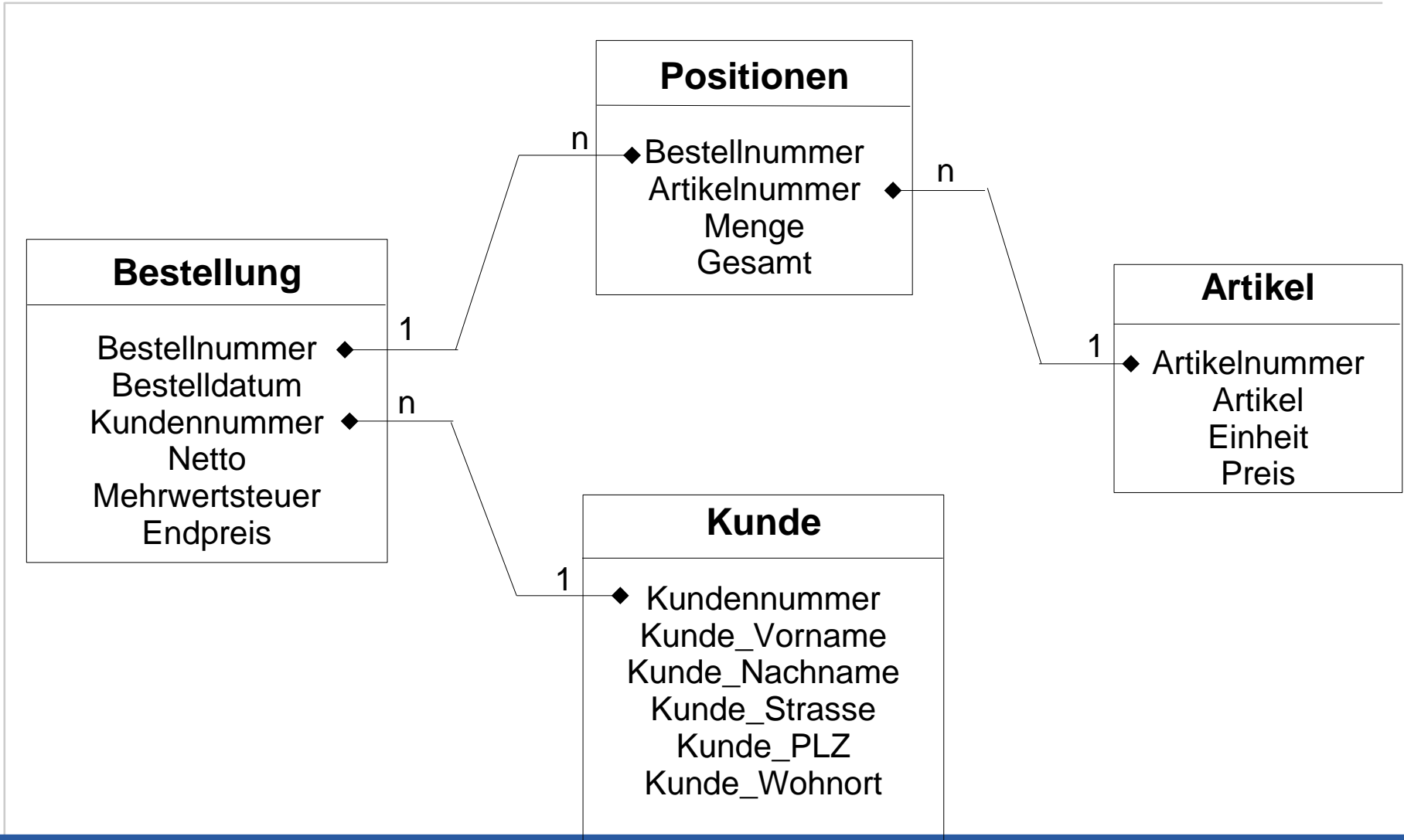
- Die Datensammlung wird in Zeilen und Spalten strukturiert abgelegt.
- Jeder Datensatz beschreibt ein Objekt. Das heißt: Jeder Datensatz unterscheidet sich von allen anderen Felder in mindestens einem Attributwert.
- Jedes Attribut kommt nur einmal in einer Tabelle vor.
- Für jedes Attribut gibt es einen Wert. Es gibt keine Wertlisten.
 - Die Mitarbeiter werden einer bestimmten Abteilung zugeordnet.
 - Die verschiedenen Spalten werden nicht als Spalten abgelegt, sondern in einer Spalte.
 - In dieser Spalte steht ein Schlüsselwert aus der Tabelle "Abteilung".
- Jedes Feld ist atomar. Der Feldinhalt kann nicht weiter untergliedert werden. Zum Beispiel:
 - Die Postleitzahl und der Ort werden in verschiedenen Spalten gespeichert.
 - Vor- und Nachname einer Person werden in verschiedenen Spalten gespeichert.



- Die 1. Normalform ist erfüllt.
- Jedes Nicht-Schlüsselfeld ist nur vom gesamten Primärschlüssel abhängig.
- Es bestehen keine funktionalen Abhängigkeiten zwischen Nicht-Schlüsselfeldern.
- In einem Datensatz kommen nur Informationen vor, die sich auf das Subjekt beziehen, welches durch den Schlüsselwert repräsentiert wird.
- Vorteile:
 - Es wird nicht unnötig Speicherplatz verbraucht.
 - Schreibfehler können leichter behoben werden.
 - Es können zusätzliche Informationen gespeichert werden.



- Die Datenbank erfüllt die 1. und 2. Normalform.
- Jeder Datensatz behandelt nur ein Thema.
- Es gibt keine funktionalen Abhängigkeiten. Zum Beispiel:
 - Ein Kunde kann nur angelegt werden, wenn er Ware bestellt.
 - Ein Lieferant kann nur angelegt werden, wenn er das erste Mal Ware liefert.
 - Die Postleitzahl ist abhängig vom Ort.
- Alle Nicht-Schlüsselfeld sind nur vom Primärschlüssel abhängig.
- Es dürfen keine indirekten Abhängigkeiten vorhanden sein.



- Gegenstands-Beziehungsmodell
- ... bildet die Gegenstände (Entity) der Wirklichkeit vereinfacht ab.
- ... stellt Objekte und deren Beziehungen dar.
- ... ist eine graphische Darstellung von Tabellen und deren Beziehungen (Relation).

- ... ist ein Gegenstand.
- ... spiegeln ein Objekt aus der realen Welt wieder. Zum Beispiel das Buch, der Kunde, der Lieferant etc. sind Entities.
- ... wird in dem ER-Modell als Rechteck dargestellt.
- ... hat eine eindeutige Kennung, die als Primärschlüssel implementiert wird.

- Die Aufgabenstellung oder das Pflichtenheft werden nach Substantiven durchsucht.
- Nach der Textanalyse werden die gefundenen Begriffe nach unnötigen oder redundanten Objekten durchgesehen.
- Um Objekte zu finden, sollte ein Text nach
 - ... existierenden Strukturen durchsucht werden. Beispiel: Personaldaten eines Kunden.
 - ... anderen Systemen und Geräten durchsucht werden. Beispiel: In einem Bestellwesen der Buchungsrechner oder das Thermostat, um die Heizung zu regeln.
 - ... Rollen von Menschen und System durchsucht werden. Beispiel: Der Besteller einer Ware, der Kfz-Halter etc.
 - ... organisatorischen Einheiten durchsucht werden. Beispiel: Bundesland, Verkaufsgebiet.

- Voraussetzung: Der *Besucher* hat einen *Bibliotheksausweis*.
- Der Besucher sucht im *Katalog* der *Bibliothek* die gewünschten *Bücher* heraus.
- Arbeitsvorgang "Buch ist im Lesesaal":
 - Das *Buch* befindet sich im *Lesesaal*. Der *Besucher* geht zu dem, im Katalog angegebenen *Regal*, Lesesaal und entnimmt das Buch.
 - Mit dem *Buch* geht der *Nutzer* zum Ausleihschalter und leiht das Buch aus.
- Arbeitsvorgang "Buch ist im Magazin":
 - Der *Nutzer* bestellt das *Buch* über den Katalog.
 - Nach einer bestimmten Zeitspanne kann das *Buch* am Ausgabeschalter abgeholt und ausgeliehen werden.
- Arbeitsvorgang "Buch ausgeliehen":
 - Der *Nutzer* reserviert das *Buch* vor.

Ausleiher

Bibliotheksausweis

Ist der Ausweis ein Attribut des Ausleihers? Sind beide Objekte ein Synonym für die gleiche Person?

Katalog

Buch

Ist der Katalog die Gesamtheit aller Bücher? Bietet der Katalog einen Mehrwert an Informationen wie die Beantwortung der Frage "Wo befindet sich das Buch"?

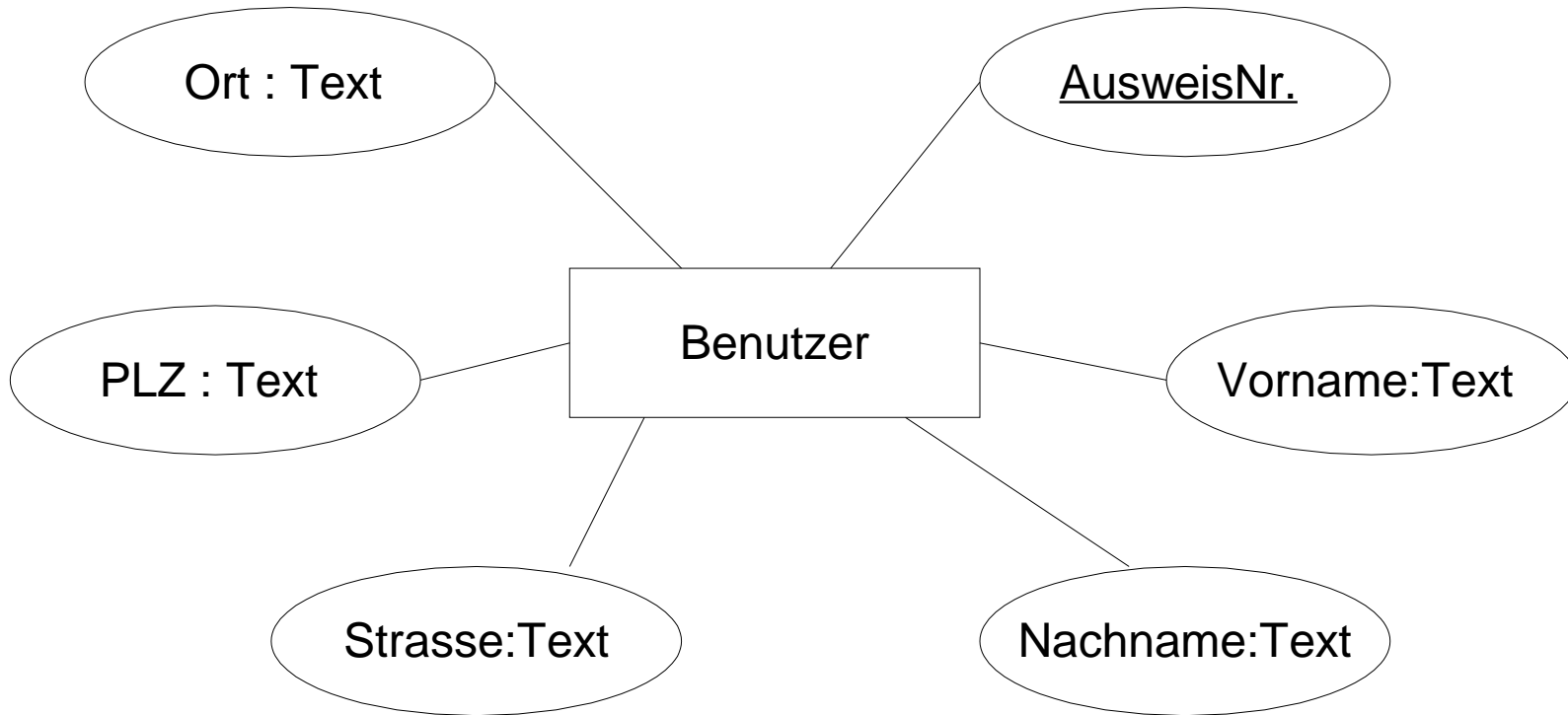
Bibliothek

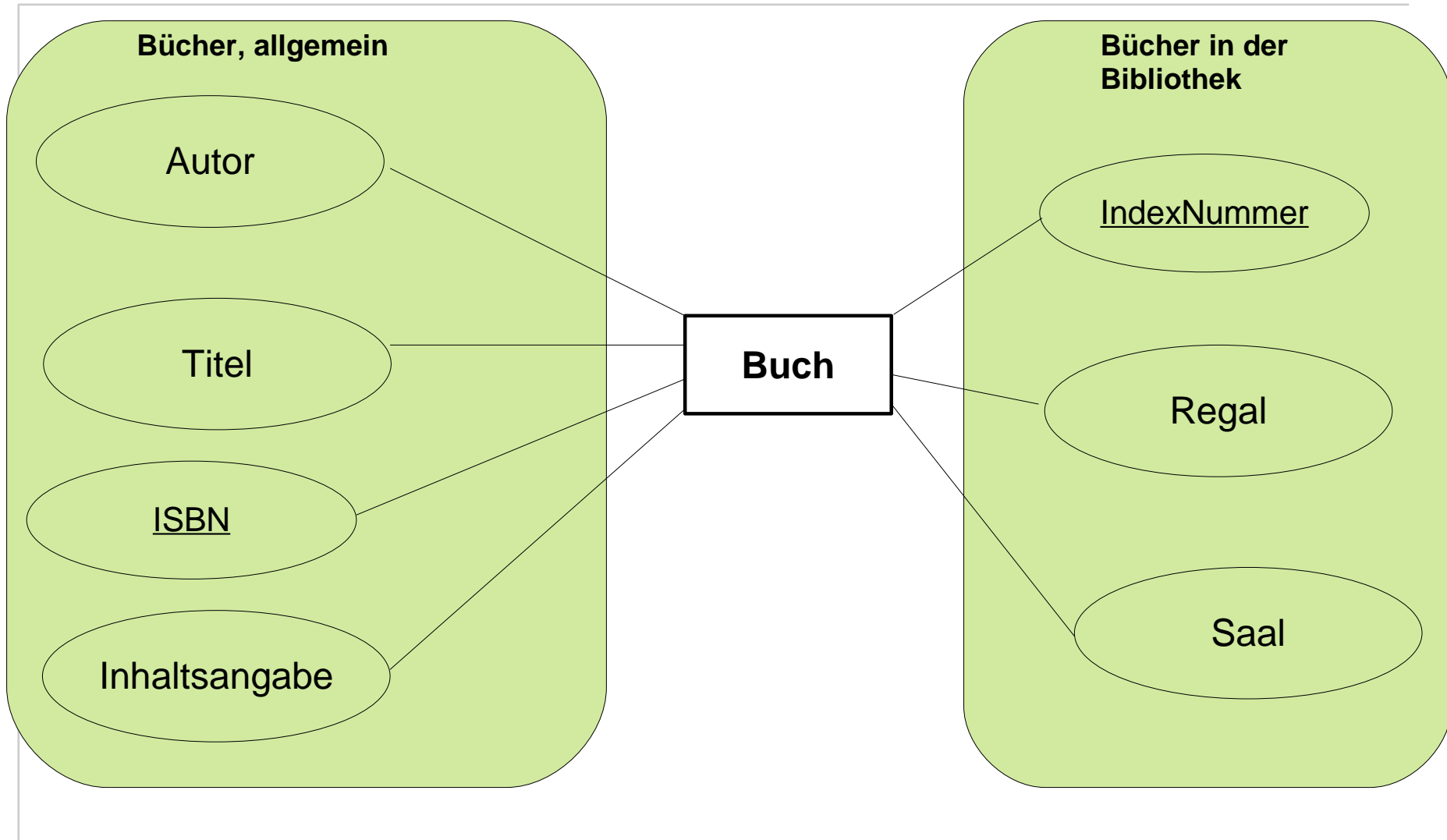
Lesesaal

Regal

Wird die Bibliothek nicht durch die Lesesäle beschrieben? Ein Regal ist eine Detailtabelle eines Lesesaals.

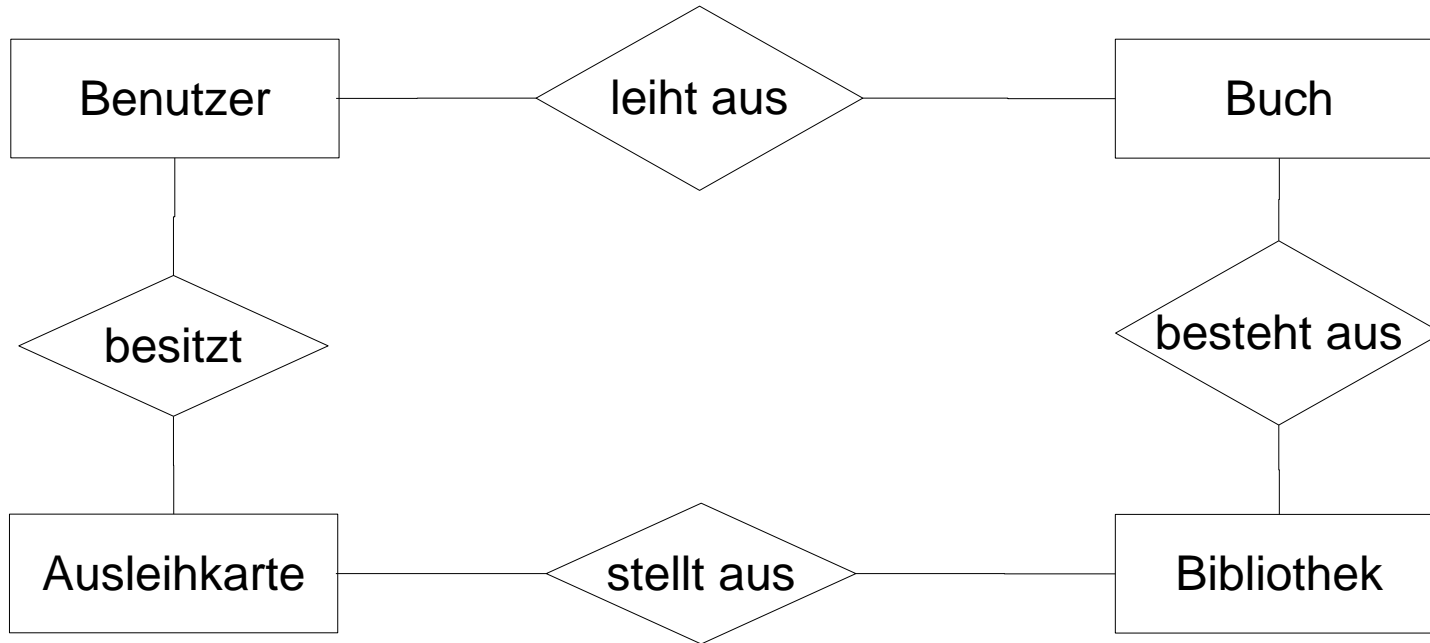
- Jede Entity wird durch Attributwerte beschrieben.
- Mit Hilfe der Attribute werden die Eigenschaften des Objekts definiert.
- Zum Beispiel Buchtitel, Wohnort eines Kunden, Lieferfirma etc. sind Attribute eines bestimmten Objekts.
- ... werden als Ellipsen dargestellt.
 - Der Name eines Schlüsselwertes (eindeutige Kennung) wird unterstrichen.
 - Der Datentyp wird rechts vom Namen angegeben werden. Als Trennzeichen zwischen Namen und Datentyp wird ein Doppelpunkt genutzt.

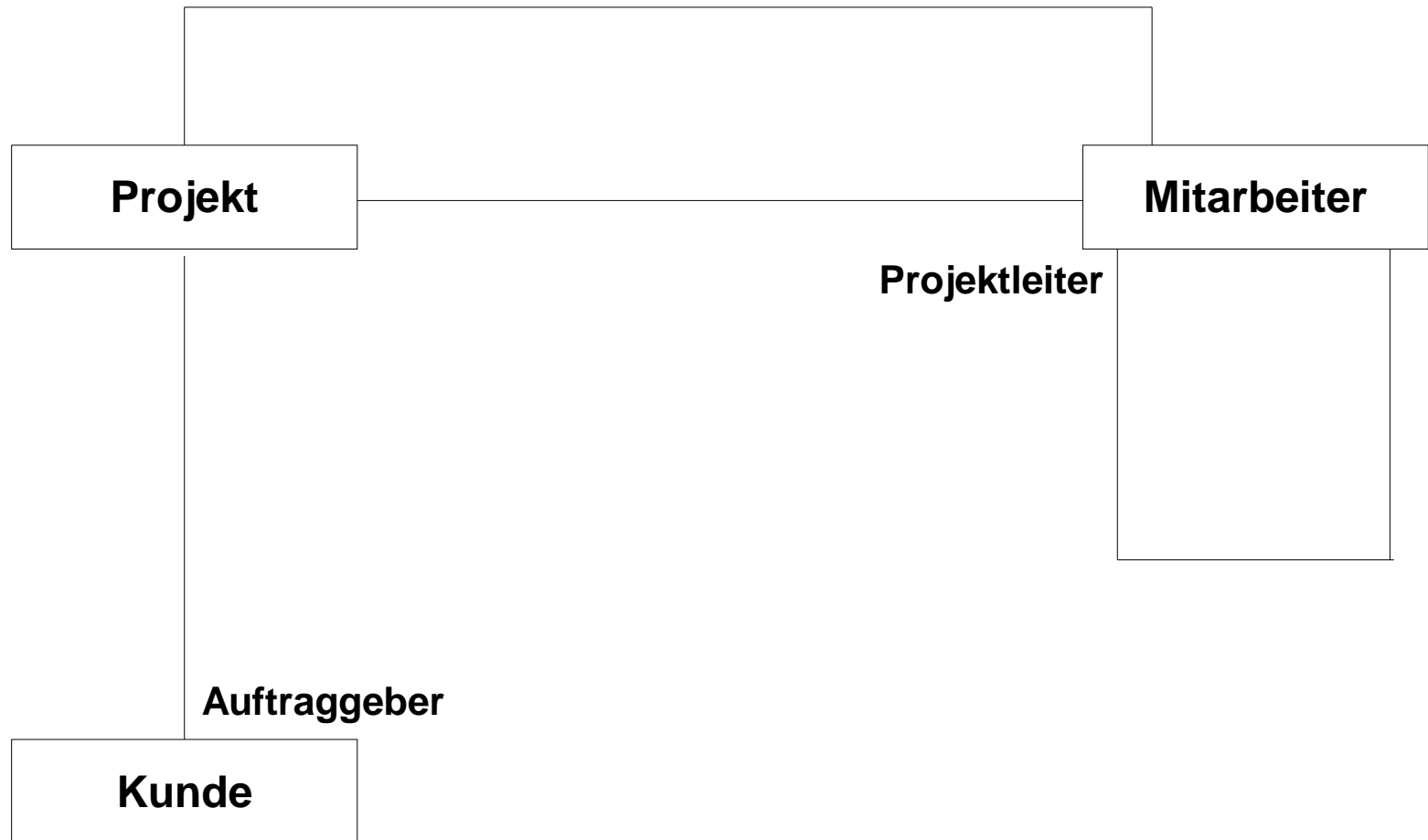




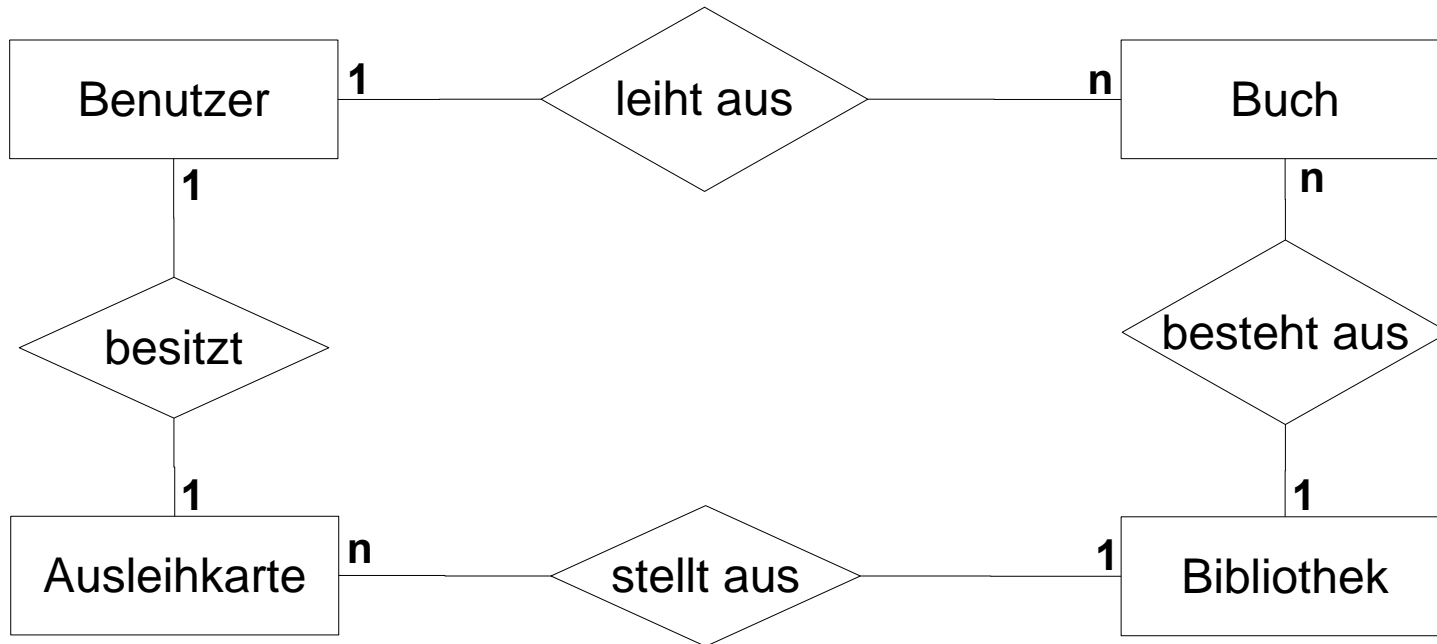
- Die Verknüpfung zwischen zwei oder mehr Entities wird als Relationship bezeichnet.
- Die verknüpften Objekte haben eine Beziehung untereinander. Zum Beispiel:
 - Ein Lieferant liefert Waren.
 - Ein Kunde kauft bestimmte Waren ein.
 - Ein Student leiht sich ein Buch in der Bibliothek aus.

- Verben oder Präpositionen beschreiben häufig Beziehungen.
- Beispiele:
 - Der Kunde *bestellt* mehrere Ware.
 - In einem Lager sind alle *lieferbaren* Waren *vorhanden*.
 - Der Student *beantragt* eine Ausleihkarte *in* der Bibliothek.
 - Der KFZ-Halter *meldet* sein Fahrzeug *an* der KFZ-Stelle an.



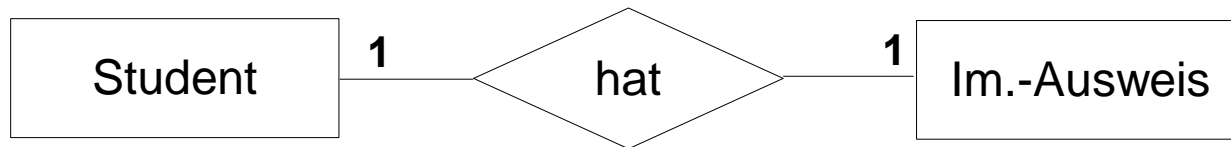


- ... beschreiben den Grad der Beziehung.
- Wie viele Objekte sind mit einem anderen Objekt verbunden?
- Beispiele
 - Der Kunde bestellt *mehrere* Ware.
 - Ein Kunde hat seit Oktober *mehrere* Bestellungen aufgegeben.
 - In einem Lager sind *alle* lieferbaren Waren vorhanden.
 - Der Nutzer beantragt *eine* Ausleihkarte in der Bibliothek.
 - Der KFZ-Halter meldet *sein* Fahrzeug an der KFZ-Stelle an.

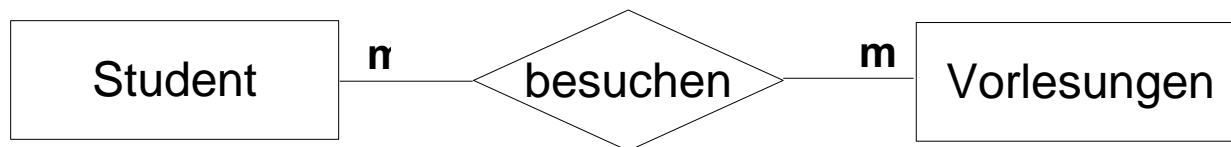




Eine Bibliothek stellt beliebig viele Ausleihausweise aus.



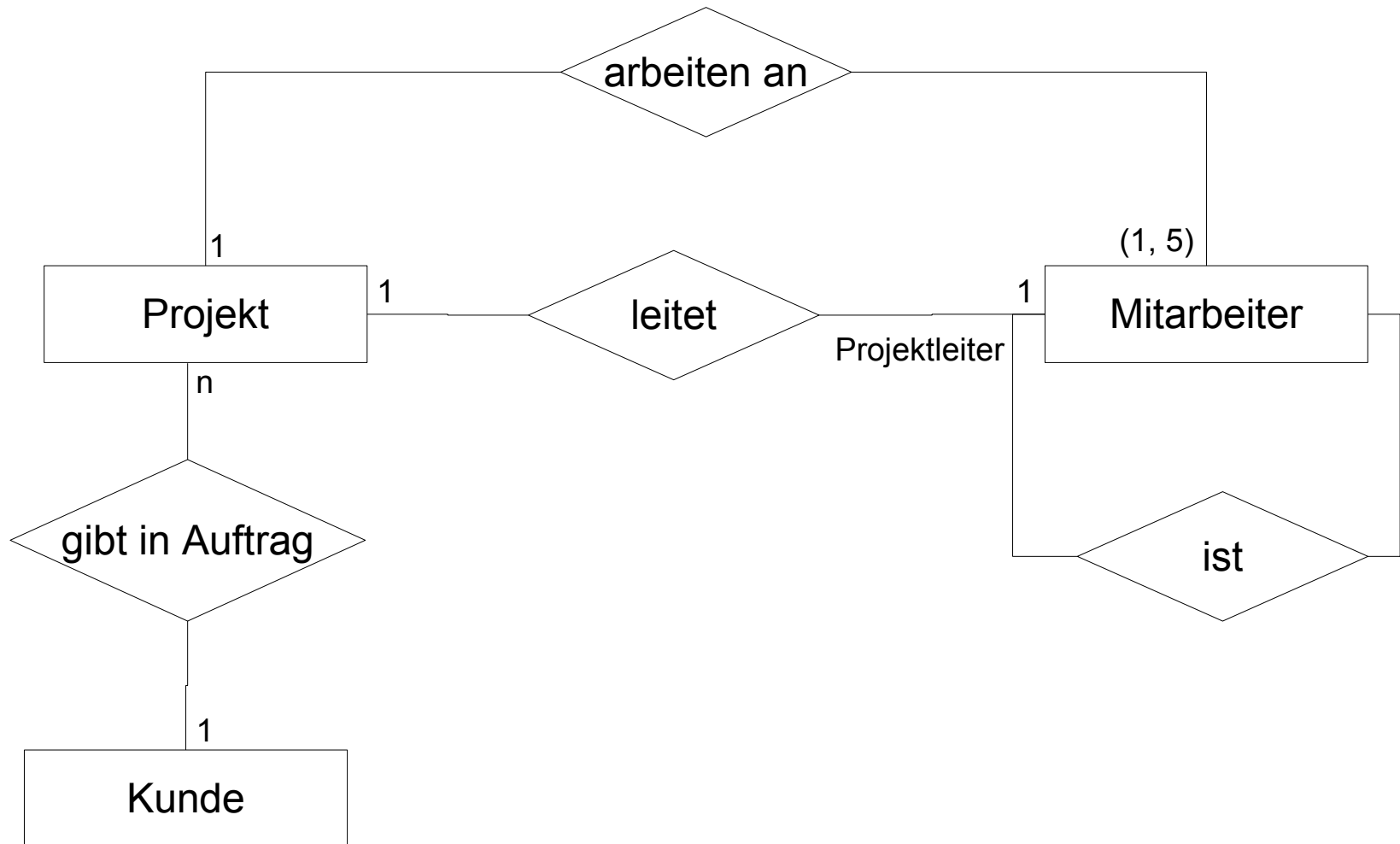
Ein Student hat einen Immatrikulationsausweis.



Viele Studenten besuchen viele Vorlesungen.



An einem Projekt arbeiten ein bis fünf Mitarbeiter



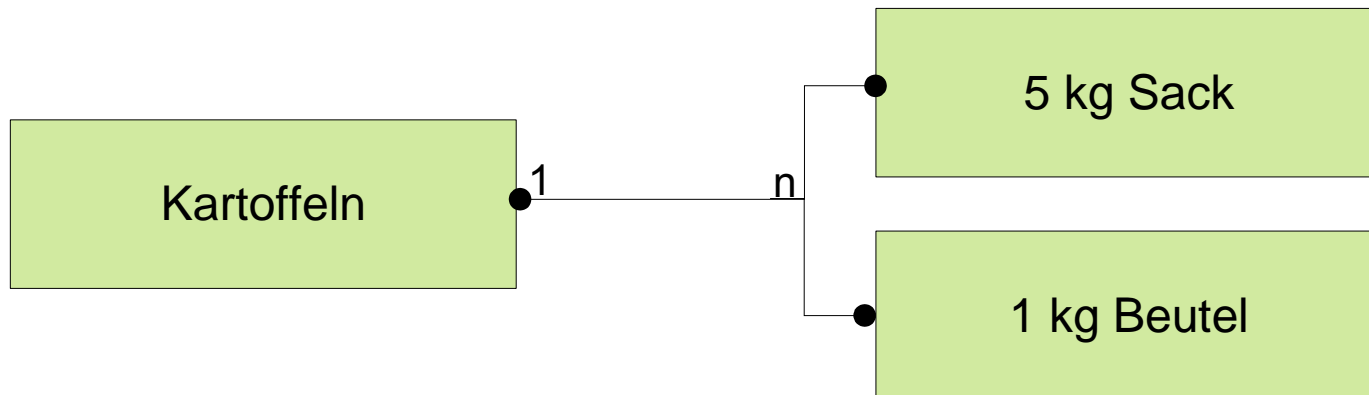
- Der Datensatz einer Mastertabelle ist mit einem Datensatz in der Detailtabelle verbunden.
- Jedem Objekt A wird exakt ein Objekt B zugeordnet.
- Diese Art der Beziehung wird häufig in einer Tabelle abgebildet.
- Sinnvoll ist diese Beziehung,
 - ... wenn statt einer Tabelle mit vielen Spalten kleine Tabellen mit wenigen Spalten genutzt werden.
 - ... falls Informationen nur einer bestimmten Personengruppe zur Verfügung stehen.
 - ... wenn sehr große Datenmengen verwaltet werden sollen.

- Ein Student besitzt einen Immatrikulationsausweis.
- Die Bibliothek stellt pro Nutzer einen Ausleihausweis aus.
- Ein Projektleiter leitet ein Projekt.
- Die Ware wird von einem Lieferanten geliefert.
- Das Bürgerbüro ordnet jedem Bürger einen Personalausweis zu.
- Einem Auto kann ein Fahrzeughalter zugeordnet werden.



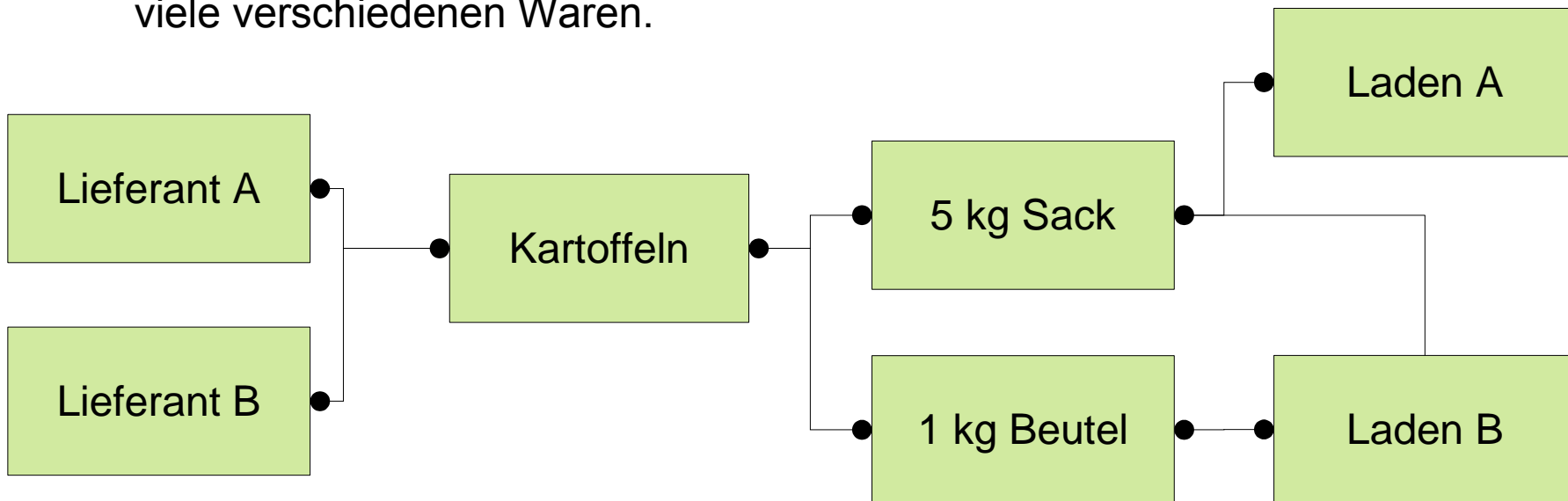
- Jedem Datensatz in der Mastertabelle sind mehrere Datensätze in der Detailtabelle zugeordnet.
- Jedem Objekt A können mehrere Objekte B zugeordnet werden.
- Häufigste Datenbeziehung.

- Die Ware kann in verschiedenen Verpackungseinheiten geliefert werden.
- In einem Kinosaal werden in Abhängigkeit der Zeit verschiedenen Filme gezeigt.
- Der Nutzer einer Bibliothek leitet sich eine bestimmte Anzahl von Büchern aus.
- Eine Firma besteht aus einer Anzahl von Mitarbeitern.



- Mehrere Datensätze lassen sich mehreren anderen Datensätzen zuordnen.
- Diese Art der Beziehung wird über eine sogenannte Verbindungstabelle realisiert.
 - Zuerst werden die Tabellen erstellt, die die gewünschten Objekte beschreiben.
 - In der Verbindungstabelle werden die Schlüsselfelder der Objekte aufgelistet, die eine Beziehung zueinander haben.
- Eine Beziehung wird immer auf 1:n-Beziehungen heruntergebrochen.

- Ein Student besucht mehrere Vorlesungen. Die Vorlesungen werden aber von mehreren Studenten besucht.
- Eine Firma besteht aus mehreren Mitarbeitern. Die Fax-Nummer der Firma kann für mehrere Mitarbeiter gelten.
- Eine Ware kann an mehrere Läden geliefert werden. Jeder Lieferant liefert viele verschiedenen Waren.



- Zwei oder mehr Tabellen werden mit Hilfe von Schlüsselwerten miteinander verbunden.
- ... liefern eine neue virtuelle Tabelle zurück.
- ... unterscheiden sich in der Verknüpfung der Tabellen.

- ... verbindet alle Zeilen der ersten Tabelle mit allen Zeilen der zweiten Tabelle.
- ... bildet das kartesische Produkt zweier Tabellen.
- ... ist meist nutzlos.

```
SELECT projektliste.projektname, mitarbeiter.nachname  
FROM projektliste CROSS JOIN mitarbeiter;
```

```
mysql> SELECT projektliste.projektname AS projekt, mitarbeiter.nachname FROM pro  
jektliste CROSS JOIN mitarbeiter;  
+-----+-----+  
| projekt | nachname |  
+-----+-----+  
| projekt01 | Schneider |  
| projekt02 | Schneider |  
| projekt03 | Schneider |  
| projekt04 | Schneider |  
| projekt01 | Schmitt |  
| projekt02 | Schmitt |  
| projekt03 | Schmitt |  
| projekt04 | Schmitt |  
| projekt01 | Schmidt |  
| projekt02 | Schmidt |  
| projekt03 | Schmidt |  
| projekt04 | Schmidt |  
| projekt01 | Müller |  
| projekt02 | Müller |  
| projekt03 | Müller |  
| projekt04 | Müller |  
+-----+-----+  
16 rows in set (0.00 sec)
```

- `tabelle.feld`
 - ... wird benötigt, wenn Felder aus verschiedenen Tabellen genutzt werden.
 - Zuerst wird die Tabelle angegeben, in der das Feld definiert ist.
 - Dann wird mit Hilfe des Punkts ein Feld mit dem Tabellennamen verbunden. Der Feldname muss in der Tabelle definiert sein.

- `SELECT tabelle.feld AS alias`
 - Für ein Datenfeld wird ein Alias vergeben.
 - Der Aliasname wird in der virtuellen Tabelle verwandt.
 - Mit Hilfe des Alias wird die Lesbarkeit erhöht.
 - ... werden nur im Ergebnis angezeigt, ändern aber nicht die Bezeichnung im Entwurf einer Tabelle.
- `SELECT alias.feld FROM tabelle AS alias`
 - Für eine Tabelle kann ein Alias vergeben werden.
 - Falls Datenfelder aus der Tabelle genutzt werden, muss der Alias-Name statt des Ursprungstabellennamen angegeben werden.
 - Mit Hilfe des Alias wird die Lesbarkeit bei sehr langen Bezeichnungen erhöht.

- ... oder Exklusionsverknüpfung, Equival-Join.
- ... verknüpft über ein oder mehrere gemeinsame Felder.
- ... sucht passend zu einem Fremdschlüssel in einer Detailtabelle einen Primärschlüssel in einer Mastertabelle.
- Es wird eine 1 : n – Beziehung nachgebildet.
- Es werden nur die Datensätze angezeigt, deren verknüpfte Werte in beiden Tabellen vorkommen.
- ... wird am häufigsten verwendet.

Europa	Frankreich
Europa	Spanien
Asien	China
Asien	Indien
	Antarktis
Afrika	Südafrika
	Arktis

Europa
Asien
Amerika
Australien
Afrika

*Inner
Join*

Europa	Frankreich
Europa	Spanien
Asien	China
Asien	Indien
Afrika	Südafrika

```
SELECT lstprojekt.projektname, mitarbeiter.nachname
FROM projektliste AS lstprojekt
INNER JOIN mitarbeiter
ON lstprojekt.projektleiter = mitarbeiter.mitarbeiter_ID ;
```

Welche Tabelle wird mit welcher verknüpft?

Bedingung der Verknüpfung

```
mysql> SELECT projektliste.projektname AS projekt, mitarbeiter.nachname FROM projektliste INNER JOIN mitarbeiter ON projektliste.projektleite = mitarbeiter.mitarbeiter_ID;
+-----+-----+
| projekt | nachname |
+-----+-----+
| projekt01 | Schneider |
| projekt02 | Schneider |
| projekt03 | Schmitt |
| projekt04 | Schmitt |
+-----+-----+
4 rows in set (0.01 sec)
```

Doppelte Datensätze
werden nicht angezeigt.



```
SELECT DISTINCT
    lstprojekt.projektname, mitarbeiter.nachname
FROM projektliste AS lstprojekt
INNER JOIN mitarbeiter
ON lstprojekt.projektleiter = mitarbeiter.mitarbeiter_ID ;
```

- ... nutzt die Operatoren $>$, $<$, $>=$, $<=$ oder $<>$ in der Bedingung.
- ... liefert alle Zeilen zurück, die nicht die gleichen Werte besitzen.

```
SELECT lstprojekt.projektname, mitarbeiter.nachname
FROM projektliste AS lstprojekt
INNER JOIN mitarbeiter
ON
  lstprojekt.projektleiter <> mitarbeiter.mitarbeiter_ID;
```

- ... stellt die Verknüpfung über Spalten mit der gleichen Bezeichnung in beiden Tabellen her.
- ... überprüft nicht die Werte der Datenfelder auf Gleichheit, sondern deren Eigenschaften.

```
SELECT lstprojekt.projektname, mitarbeiter.nachname  
FROM projektliste AS lstprojekt  
NATURAL JOIN mitarbeiter;
```

- ... zeigt alle Datensätze einer Tabelle an und alle dazugehörigen Datensätze aus einer anderen Tabelle.
- ... kann als Left-Outer-Join oder Right-Outer-Join implementiert werden.
- ... beantwortet Fragen wie zum Beispiel:
 - Welcher Kunden hat noch keine Bestellung getätigt.
 - Welcher Lieferant hat keine Waren in einem bestimmten Monat geliefert?
 - Welcher Student hat an keiner Prüfung teilgenommen?

Europa	Frankreich
Europa	Spanien
Asien	China
Asien	Indien
	Antarktis
Afrika	Südafrika
	Arktis

*Left
Join*

Europa
Asien
Amerika
Australien
Afrika

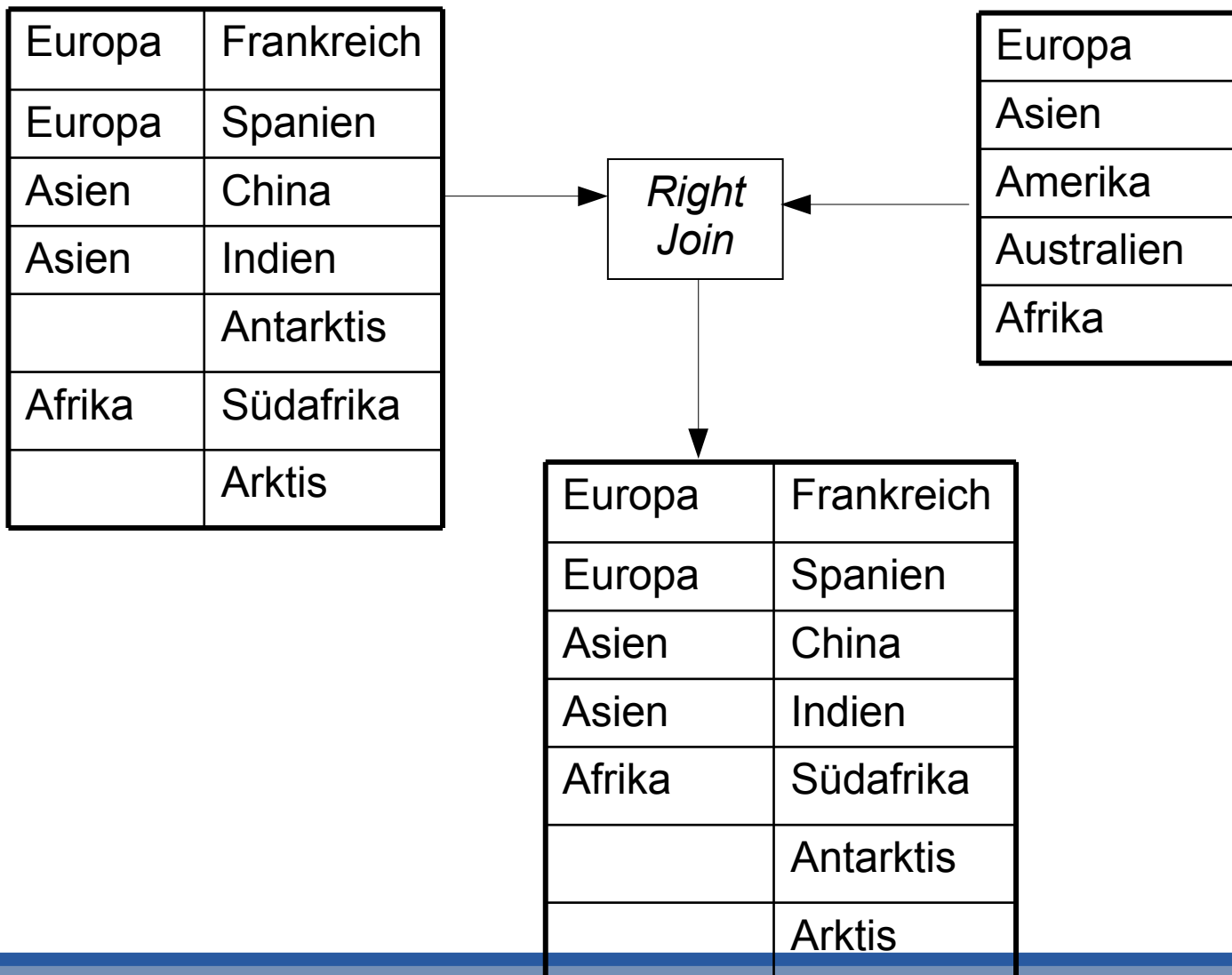
Europa	Frankreich
Europa	Spanien
Asien	China
Asien	Indien
Afrika	Südafrika
Amerika	
Australien	

```
SELECT lstprojekt.projektname, mitarbeiter.nachname  
FROM projektliste AS lstprojekt  
LEFT OUTER JOIN mitarbeiter  
ON lstprojekt.projektleiter = mitarbeiter.mitarbeiter_ID ;
```

Alle Datensätze der Tabelle links werden angezeigt plus die dazugehörigen aus der rechten Tabelle.

Bedingung der Verknüpfung

```
mysql> SELECT mitarbeiter.nachname, lstprojekt.projektname FROM mitarbeiter LEFT  
OUTER JOIN projektliste AS lstprojekt ON lstprojekt.projektleite = mitarbeiter.  
mitarbeiter_ID;  
+-----+-----+  
| nachname | projektname |  
+-----+-----+  
| Schneider | projekt01 |  
| Schneider | projekt02 |  
| Schmitt   | projekt03 |  
| Schmitt   | projekt04 |  
| Schmidt   | NULL      |  
| Müller    | NULL      |  
+-----+-----+  
6 rows in set (0.00 sec)
```



```
SELECT abteilung.abtname, mitarbeiter.nachname  
FROM abteilung  
RIGHT OUTER JOIN mitarbeiter  
ON abteilung.abteilung_ID = mitarbeiter.abteilung;
```

Alle Datensätze der Tabelle rechts werden angezeigt plus die dazugehörigen aus der linken Tabelle.

Bedingung der Verknüpfung

```
mysql> SELECT mitarbeiter.nachname, lstprojekt.projektname FROM mitarbeiter LEFT  
OUTER JOIN projektliste AS lstprojekt ON lstprojekt.projektleite = mitarbeiter.  
mitarbeiter_ID;  
+-----+-----+  
| nachname | projektname |  
+-----+-----+  
| Schneider | projekt01 |  
| Schneider | projekt02 |  
| Schmitt   | projekt03 |  
| Schmitt   | projekt04 |  
| Schmidt   | NULL      |  
| Müller    | NULL      |  
+-----+-----+  
6 rows in set (0.00 sec)
```

```
SELECT bestellung.produkt,  
       artikel.artikelname,  
       lagerbestand.menge
```

Welche Daten
werden angezeigt?

```
FROM Bestellung
```

Woher kommen die Hauptdaten?

```
INNER JOIN artikel  
       ON artikel.artikel_ID = bestellung.produkt
```

Es werden nur die Daten angezeigt,
die ein Pendant in artikel besitzen

```
LEFT JOIN lagerbestand  
       ON bestellung.produkt = lagerbestand.produkt;
```

Zusätzlich werden noch die Daten für
die Bestellung aus dem Lager angezeigt.

bestellung	
produkt	menge
1	4
2	5
3	2
1	1
4	5

FROM Bestellung

temp	
produkt	name
1	Fernseher
2	MG3-Player
3	CD-Player
1	Fernseher
4	DVD-Recorder

ON

artikel.artikel_ID =
bestellung.produkt

INNER JOIN artikel

artikel	
nummer	name
1	Fernseher
2	MG3-Player
3	CD-Player
4	DVD-Recorder

temp	
produkt	name
1	Fernseher
2	MG3-Player
3	CD-Player
1	Fernseher
4	DVD-Recorder

temp		
produkt	name	menge
1	Fernseher	50
2	MG3-Player	20
3	CD-Player	120
1	Fernseher	1
4	DVD-Recorder	

lagerbestand	
produkt	menge
1	50
2	20
3	120

LEFT JOIN
lagerbestand

ON bestellung.produkt =
lagerbestand.produkt

- ... sind Abfragen in einer Abfrage.
- ... werden genutzt, wenn das Ergebnis einer Abfrage für eine andere Abfrage als Kriterium genutzt wird.
- ... besitzen folgende Kennzeichen:
 - Unterabfragen sind immer SELECT-Anweisungen.
 - Unterabfragen müssen immer geklammert werden. Die Klammerung erhöht die Lesbarkeit. Die innere Abfrage wird von der äußeren Abfrage abgegrenzt.
 - Das Ende einer Unterabfrage wird nicht durch ein Semikolon angezeigt.
- ... werden
 - ... anstelle der Namen von Datenfeldern in SELECT-Anweisungen genutzt.
 - ... in der WHERE-Anweisung einer SELECT-, UPDATE- oder DELETE-Anweisung genutzt.
 - ... als Wert (`SET ...`) in einer UPDATE-Anweisung genutzt.
 - ... als Wert (`VALUES ...`) in einer INSERT-Anweisung genutzt.

Es werden alle Artikel und deren Preise angezeigt.

Durch die Bedingung wird nur der Artikel angezeigt, der den höchsten Preis besitzt.

```
SELECT artikel, preis FROM LAGER  
WHERE preis =  
(SELECT MAX(artikelpreis) AS maxpreis FROM lager);
```

Die Abfrage gibt den höchsten Preis zurück. Es wird exakt ein Wert zurückgegeben. Die Abfrage wird als innere Abfrage bezeichnet.

```
SELECT artikel, preis FROM LAGER  
  
ORDER BY preis DESC  
  
LIMIT 1;
```

- Wann liefert diese Abfrage kein korrektes Ergebnis?

Es werden alle Informationen, die in der Tabelle bestellung enthalten sind, angezeigt.

Durch die Bedingung werden nur Artikel angezeigt, deren Artikelnummer mit einem Wert aus der Liste übereinstimmen.

```
SELECT * FROM bestellung
WHERE produkt IN
(SELECT artikel_ID FROM artikel
WHERE artikelname LIKE 'B%');
```

Die Abfrage liefert alle Artikelnummern von Artikeln, die mit dem Buchstaben B beginnen. Es wird nur eine Spalte zurückgeliefert. Die Spalte enthält die Werte für die Liste.

Existiert die Untermenge in der äußeren Abfrage? R | R | Z | N |

Es werden alle Informationen, die in der Tabelle artikel enthalten sind, angezeigt.

Es werden alle Artikel angezeigt, die auch in der Tabelle bestellung, existieren.

```
SELECT * FROM artikel
```

```
WHERE EXISTS
```

```
(SELECT * FROM bestellung  
WHERE bestellung.produkt = artikel-artikel_ID);
```

Hier wird ein Filter geschrieben, der alle Bestellungen herausucht, deren Artikel auch in der Artikel-Tabelle vorhanden sind.

Existiert die Untermenge in der äußeren Abfrage? R | R | Z | N |

Es werden alle Informationen, die in der Tabelle artikel enthalten sind, angezeigt.

Es werden alle Artikel angezeigt, die nicht in der Tabelle bestellung, existieren.

```
SELECT * FROM artikel
```

```
WHERE NOT EXISTS
```

```
(SELECT * FROM bestellung  
WHERE bestellung.produkt = artikel-artikel_ID);
```

Hier wird ein Filter geschrieben, der alle Bestellungen herausucht, deren Artikel auch in der Artikel-Tabelle vorhanden sind.