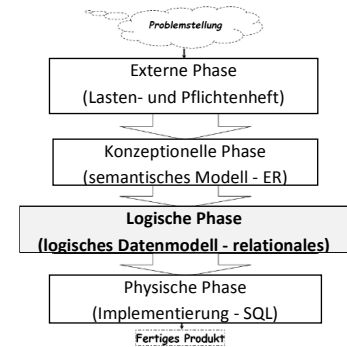


Die im ER-Modell gefundenen Beziehungen und Informationen sollen im relationalen Modell in Tabellenform dargestellt werden. Eine relationale Datenbank kann man sich als eine Sammlung von Tabellen vorstellen, in welchen Datensätze abgespeichert sind. Jede Zeile in einer Tabelle ist ein Datensatz, der aus einer Reihe von Attributen (Spalten der Tabelle) besteht. Die Umwandlung der Entitätsmengen in Tabellen erfolgt nach bestimmten Regeln, die hier am Beispiel Kundenkonten, Kunden, Bestellungen, Artikel dargestellt werden.



### 3.1 Gegenüberstellung von Begrifflichkeiten

| ER-Modell                               | Relationen-Modell<br>(Logisches Modell) | Relationale Datenbank  |
|---|---|------------------------|
| Entität                                 | Tupel                                   | Zeile oder Datensatz   |
| Entitätsmenge                           | Relation                                | Tabelle                |
| funktionale Beziehung<br>(Relationship) | Fremdschlüssel                          | Spaltenüberschrift(en) |
| Attribut                                | Attribut                                | Spaltenüberschrift     |
| Attributwert                            | Attributwert                            | Zelle                  |

Abbildung 1 Gegenüberstellung von Begriffen

### 3.2 Vorgehensweise (Abbildungsregeln)

| Regel | Betrifft                           | Abbildungsregel für das relationale Modell   |
|-------|------------------------------------|--|
| 1     | Entitätsmenge                      | Jede Entitätsmenge ist in eine eigene Relation (Tabellen) zu überführen.   |
| 2     | Primärschlüssel                    | Jedes Datensatz einer Relation (Tupel) muss eindeutig identifizierbar sein.  |
| 3     | 1:N Beziehung<br>(1:C Beziehungen) | Der Primärschlüssel der 1-Relation ist als Fremdschlüssel in der N-Relation zu hinterlegen.  |
| 4     | 1:1 Beziehung                      | Der Primärschlüssel der 1-Relation ist als Fremdschlüssel in der anderen 1-Relation zu hinterlegen. Alternative: In einer Relation zusammenfassen. |
| 5     | M:N Beziehung                      | Jede komplex-komplexe Beziehungsmenge (z.B. m:n) muss als eigenständige Relation definiert werden.   |

Abbildung 2 Abbildungsregeln für die Umwandlung in ein logisches Modell

**Regel 1: Jede Entitätsmenge ist in eine eigene Relation (Tabellen) zu überführen.**

Aus jeder Entitätsmenge wird eine eigenständige Relation (=Tabelle). Die Attribute sind in den Relationen entsprechend einzutragen und entsprechen später den Spalten der Tabelle. Die Darstellung kann sowohl durch ein Tabellenschema als auch durch das Relationenschema (vgl. letzter Abschnitt 3.3) erfolgen. In beiden Fällen wird der Relation ein eindeutiger Name (Plural) gegeben und aufgeschrieben. Die Attribute werden untereinander aufgezählt.

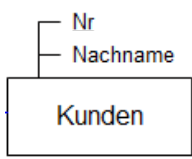
| ER-Diagramm<br>(Entitätsmenge)  | Logisches Modell  | Beispiel-Tabelle in<br>relationaler Datenbank |          |          |  |          |          |     |      |     |      |  |
|---|---|---|----------|----------|--|----------|----------|-----|------|-----|------|--|
|  | <table><tr><td>Kunden</td></tr><tr><td>KundenNr</td></tr><tr><td>Nachname</td></tr></table> | Kunden  | KundenNr | Nachname | <table><tr><td>KundenNr</td><td>Nachname</td></tr><tr><td>101</td><td>Bolt</td></tr><tr><td>102</td><td>Hamp</td></tr></table> | KundenNr | Nachname | 101 | Bolt | 102 | Hamp |  |
| Kunden  |   |   |          |          |  |          |          |     |      |     |      |  |
| KundenNr  |   |   |          |          |  |          |          |     |      |     |      |  |
| Nachname  |   |   |          |          |  |          |          |     |      |     |      |  |
| KundenNr  | Nachname  |   |          |          |  |          |          |     |      |     |      |  |
| 101   | Bolt  |   |          |          |  |          |          |     |      |     |      |  |
| 102   | Hamp  |   |          |          |  |          |          |     |      |     |      |  |

Abbildung 3 Darstellung im Tabellenschema

**Regel 2: Jeder Datensatz einer Relation (Tupel) muss eindeutig identifizierbar sein.**

Für jede Relation muss ein Primärschlüssel („Primary Key“ oder Identifikationsschlüssel) festgelegt werden. Dieser Primärschlüssel, kurz PS, dient der **eindeutigen** und unverwechselbaren Identifizierung eines Tupels (=Zeile). Es gibt verschiedene Arten von Primärschlüssel. Von einem natürlichen PS ist die Rede, wenn Attribute verwendet werden, die bereits durch die Relation vorgegeben werden (z. B. Relation „Auto“ und PS „Fahrgestellnummer“). Von einem künstlichen PS ist die Rede, wenn ein neues Attribut der Relation hinzugefügt wird, das nicht bereits natürlich vorliegt (z. B. Relation „Abteilung“ und PS „Abteilungsnummer“). Ein PS kann auch aus mehreren Attributen gebildet werden. In diesem Fall ist von einem zusammengesetzten PS die Rede. Das Attribut (=Spalte), das dem Primärschlüssel entspricht, wird unterstrichen.

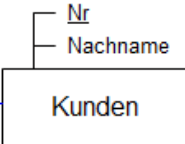
| ER-Diagramm<br>(Entitätsmenge)  | Logisches Modell   | Beispieltabelle |                 |          |   |                 |          |     |      |     |      |  |
|---|--|-----------------|-----------------|----------|---|-----------------|----------|-----|------|-----|------|--|
|  | <table><tr><td>Kunden</td></tr><tr><td><u>KundenNr</u></td></tr><tr><td>Nachname</td></tr></table> | Kunden          | <u>KundenNr</u> | Nachname | <table><tr><td><u>KundenNr</u></td><td>Nachname</td></tr><tr><td>101</td><td>Bolt</td></tr><tr><td>102</td><td>Hamp</td></tr></table> | <u>KundenNr</u> | Nachname | 101 | Bolt | 102 | Hamp |  |
| Kunden  |  |                 |                 |          |   |                 |          |     |      |     |      |  |
| <u>KundenNr</u>   |  |                 |                 |          |   |                 |          |     |      |     |      |  |
| Nachname  |  |                 |                 |          |   |                 |          |     |      |     |      |  |
| <u>KundenNr</u>   | Nachname   |                 |                 |          |   |                 |          |     |      |     |      |  |
| 101   | Bolt   |                 |                 |          |   |                 |          |     |      |     |      |  |
| 102   | Hamp   |                 |                 |          |   |                 |          |     |      |     |      |  |

Abbildung 4 Primärschlüssel werden unterstrichen dargestellt

Wichtige Regeln beim Festlegen eines Primärschlüssels.

- In einer Primärschlüssel-Spalte dürfen keine Attributwerte doppelt vorkommen (einmalig).
- Ein Primärschlüssel muss einen Wert haben.
- Der Attributwert des Primärschlüssels sollte möglichst kurz (effizient) sein.
- Bei einem zusammengesetzten Primärschlüssel sollten nicht mehr Attribute als unbedingt notwendig zur Bildung des Schlüssels kombiniert werden (**Minimalitätsprinzip**)



**Regel 4: Der Primärschlüssel der 1-Relation ist als Fremdschlüssel in der anderen 1-Relation zu hinterlegen.**

### Variante 1

Um eine 1:1 Beziehung darzustellen, wird der Primärschlüssel einer Relation als Fremdschlüssel („Foreign Key“) in die andere Relation eingetragen. Ein Fremdschlüssel ist also immer in einer anderen Relation Primärschlüssel. Dabei ist es egal, welche Relation den Fremdschlüssel erhält. Es sollte aber die Relation gewählt werden, in der möglichst wenig leere Zelleneinträge entstehen.

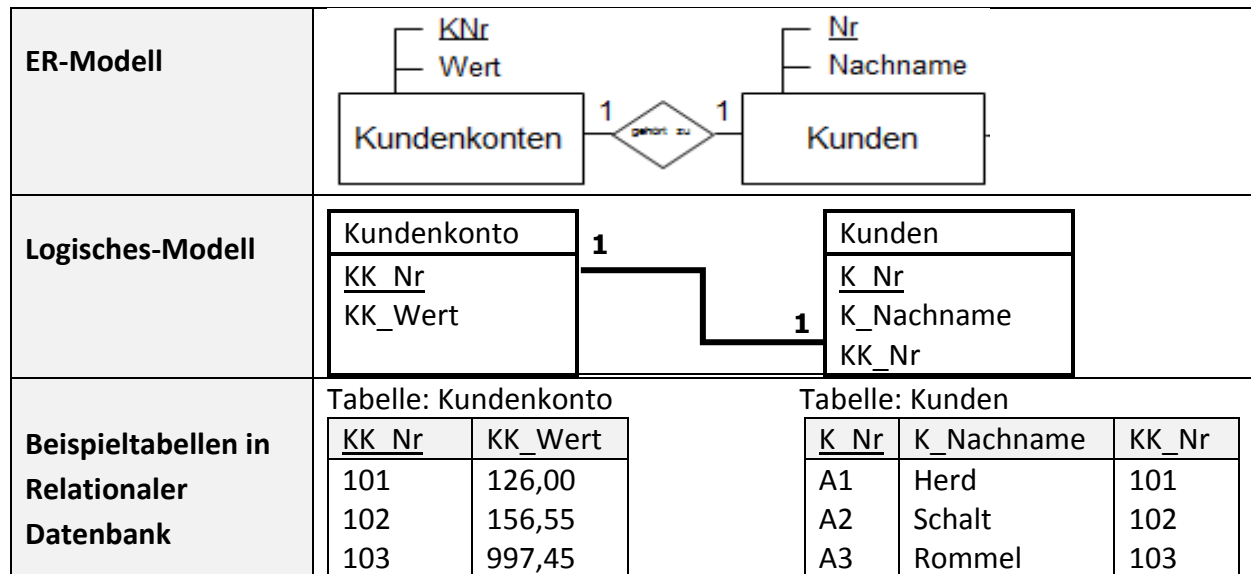


Abbildung 6 Darstellung des logischen Modells Variante 1

### Variante 2

Alternativ erhalten beide Relationen denselben Primärschlüssel. Dadurch ist der Primärschlüssel gleichzeitig auch Fremdschlüssel.

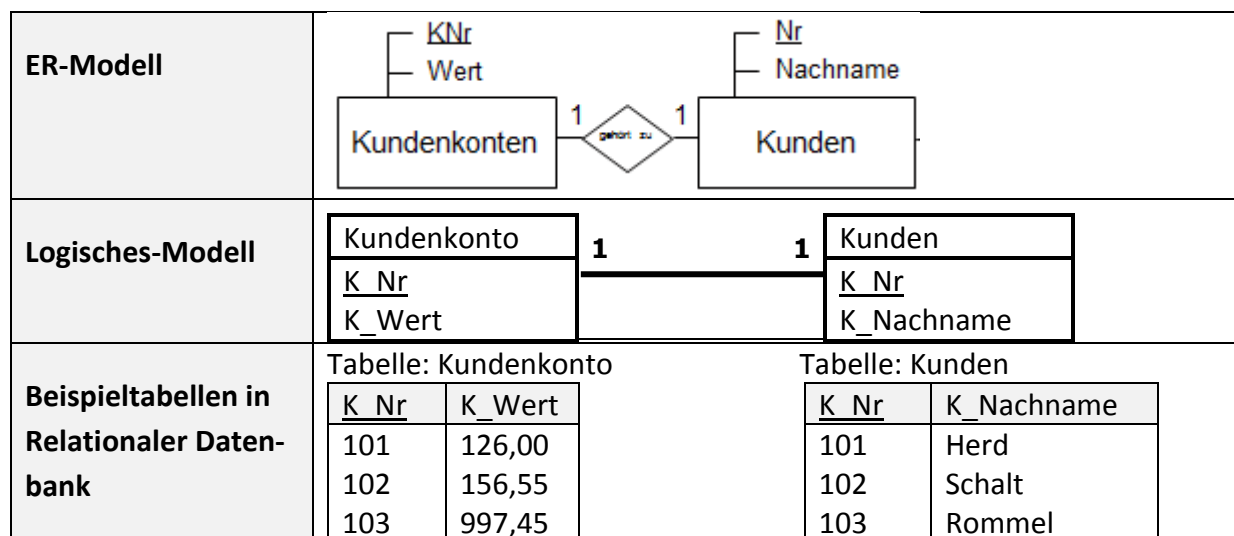


Abbildung 7 Darstellung des logischen Modells Variante 2

### Variante 3

In vielen Fällen können 1:1 Beziehungsmengen auch nur durch eine Relation abgebildet werden. In diesem Fall erhält eine Relation alle Attribute beider Entitätsmengen. Die zweite Relation wird dadurch nicht mehr benötigt.

