

**Datenmodellierung
mittels E/R-Diagrammen**



Was ist ein E/R-Modell?

- Das Entity-Relationship-Modell dient dazu, im Rahmen der semantischen Datenmodellierung einen in einem gegebenen Kontext, z. B. einem Projekt zur Erstellung eines Informationssystems, relevanten Ausschnitt der realen Welt zu beschreiben.
- Das E/R-Modell besteht aus einer Grafik, dem E/R-Diagramm (ERD) und einer Beschreibung der darin verwendeten Elemente, wobei deren Struktur und deren Bedeutung, die Semantik, dargestellt wird.
- Ein E/R-Modell dient
 - sowohl in der konzeptionellen Phase der Anwendungsentwicklung der Verständigung zwischen Anwendern und Entwicklern,
 - als auch in der Implementierungsphase als Grundlage für das Design der meist relationalen Datenbank.
- Dabei wird nur das „Was“ behandelt, also die fachlich-sachlichen Gegebenheiten, nicht aber das „Wie“, die Technik.

Was ist ein E/R-Modell?

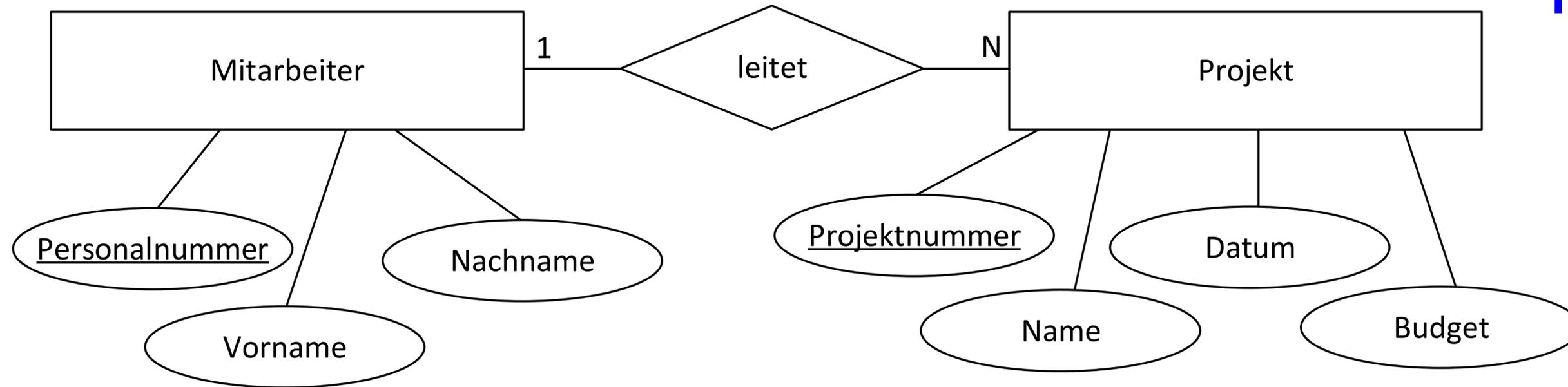
- Der Einsatz von E/R-Modellen ist der De-facto-Standard für die Datenmodellierung, auch wenn es unterschiedliche grafische Darstellungsformen für Datenmodelle gibt.
- Das ursprüngliche E/R-Modell wurde 1976 von Peter Chen vorgestellt, man spricht hier auch von der Chen-Notation.
 - Die Beschreibungsmittel für Generalisierung und Aggregation wurden 1977 von Smith eingeführt.
 - Danach gab es noch mehrere Weiterentwicklungen, so Ende der 1980er Jahre durch Wong und Katz.
- Von der Mächtigkeit her sind die E/R-Modelle vergleichbar mit den Klassendiagrammen der UML, wobei die E/R-Modelle nur die Daten darstellen, nicht jedoch die Funktionen/Methoden.

E/R-Modellierung mit Chen-Notation



- Ein Entitätstyp bzw. eine einzelne Datenbanktabelle entspricht einer Klasse von individuell identifizierbaren Objekten.
 - Ein Entitätstyp nennt man auch Entitätsklasse.
 - Eine Entität bzw. eine konkrete Tabellenzeile ist ein individuell identifizierbares Objekt.
 - Eine Entität nennt man auch Tupel oder Datensatz.
- Eine Beziehung ist eine Verknüpfung bzw. ein Zusammenhang zwischen zwei oder mehreren Entitäten.
- Ein Attribut ist eine Eigenschaft, die im Kontext zu einer Entität steht. Dies ist ein vertikaler Spaltenindex einer Tabelle.

E/R-Modellierung mit Chen-Notation: Erstes Beispiel

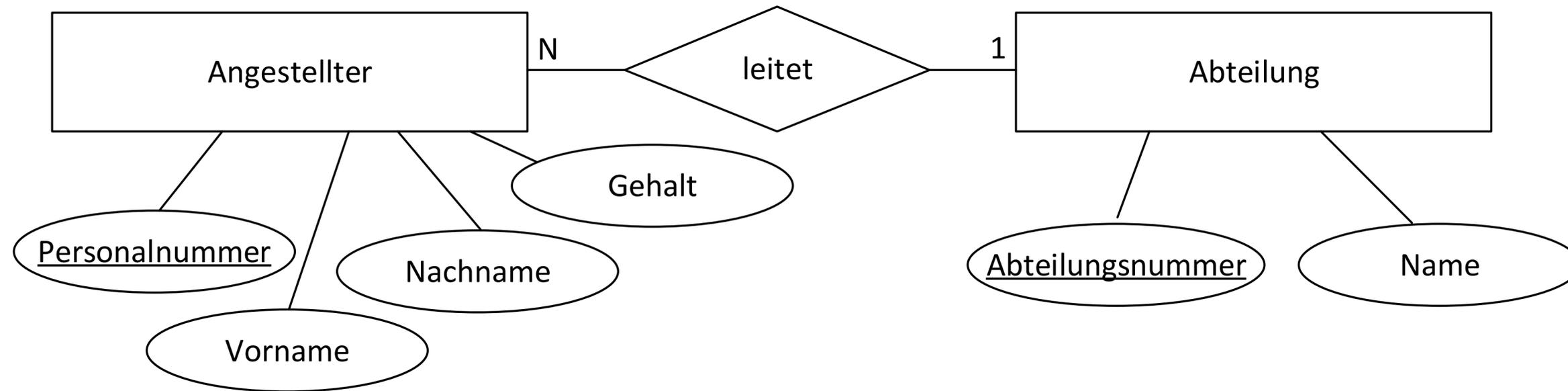


- Ein Mitarbeiter hat
 - eine Personalnummer und
 - einen Vor- und Nachnamen.
- Ein Projekt hat
 - eine Projektnummer
 - einen Namen,
 - ein Datum und
 - ein Budget.
- Ein Mitarbeiter kann mehrere Projekte leiten, aber nur ein Projekt kann von genau einem Mitarbeiter geleitet werden.

Primärschlüssel

- Ein Schlüssel ist ein eindeutig identifizierbares Attribut, bzw. eine identifizierbare Attributkombination die zudem noch minimal ist.
- Generell kann es mehrere Schlüssel und Schlüsselkombinationen geben.
- Allerdings legt man sich auf einen einzigen Schlüssel fest, den man Primärschlüssel nennt.
- Üblicherweise ist der Primärschlüssel ein sehr einfaches, oft auch "künstliches" Attribut, wie beispielsweise die
 - Mitarbeiternummer,
 - Abteilungsnummer,
 - Matrikelnummer oder einfach nur eine
 - ID.

Primärschlüssel



Primärschlüssel: Ein einzelnes Attribut

<i>ISBN</i>	<i>Autor</i>	<i>Buchtitel</i>	...
0001	Hans	V	...
0002	Lutz	W	...
0003	Peter	W	...
0004	Peter	X	...
0005	Ralf	Y	...
...

Primärschlüssel: Aus mehreren Attributen

<i>Name</i>	<i>Geburtstag</i>	<i>Wohnort</i>	...
Heinz Hoffmann	01.08.1966	Norden, BBS	...
Alf Appel	08.11.1957	Mömlingen	...
Sebastian Sonnenschein	04.08.1979	Hamburg	...
Klaus Kleber	15.04.1970	Frankfurt	...
Barbara Bachmann	17.10.1940	Kirchheim	...
...

- ...dies ist keine schöne Modellierung, selbst hier sind noch Dopplungen möglich.
- Sinnvoller wäre das Einfügen einer ID, Personenummer o.ä., die dann in jedem Fall eindeutig ist.

Schlüsselkandidat

- Als Schlüsselkandidat wird ein Attribut oder eine Attributkombination bezeichnet, die einen Datensatz eindeutig identifizieren.
- Ein Schlüsselkandidat ist also prinzipiell in der Lage, einen Primärschlüssel zu bilden.
 - Öfters hat man in einer Tabelle mehrere gültige Schlüsselkandidaten, von denen man dann einen zum Primärschlüssel wählt.

Kardinalität

- Die Kardinalität von Beziehungen definiert, wie viele Entitäten eines Entitätstyps mit genau einer Entität des anderen am Beziehungstyp beteiligten Entitätstyps und umgekehrt in Relation, also in Beziehung, stehen können oder müssen.
- Die Kardinalität von Beziehungen ist in relationalen Datenbanken in folgenden Formen vorhanden:
 - 1:1 Beziehung,
 - 1:N Beziehung und
 - N:M Beziehung.

Kardinalität 1:1

- In einer 1:1-Beziehung in relationalen Datenbanken ist jeder Datensatz in Tabelle A genau einem Datensatz in Tabelle B zugeordnet und umgekehrt.
- Diese Art von Beziehung sollte in der Modellierung vermieden werden, weil die meisten Informationen, die auf diese Weise in Beziehung stehen, sich in einer Tabelle befinden können.
- Eine 1:1-Beziehung verwendet man nur, um eine Tabelle aufgrund ihrer Komplexität zu teilen oder um einen Teil der Tabelle aus Gründen der Zugriffsrechte (ggf. Datenschutz) zu isolieren.

Kardinalität 1:N

- Eine 1:n-Beziehung relationalen Datenbanken ist der häufigste Beziehungstyp in einer Datenbank.
- In einer 1:n-Beziehung können einem Datensatz in Tabelle A mehrere passende Datensätze in Tabelle B zugeordnet sein, aber einem Datensatz in Tabelle B ist nie mehr als ein Datensatz in Tabelle A zugeordnet.

Kardinalität N:M

- Bei n:m-Beziehung in relationalen Datenbanken können jedem Datensatz in Tabelle A mehrere passende Datensätze in Tabelle B zugeordnet sein und umgekehrt.
- Diese Beziehungen können nur über eine dritte Tabelle, eine Verbindungstabelle C, realisiert werden.
- Die Verbindungstabelle C enthält minimal nur die Fremdschlüssel der beiden anderen Tabellen A und B.
- Der Primärschlüssel der Verbindungstabelle wird aus diesen beiden Fremdschlüsseln gebildet.
- Daraus folgt, dass eine N:M Beziehung in Wirklichkeit zwei 1:N Beziehungen sind.

Fremdschlüssel 1:1

- Bei einer 1:1-Beziehung ist
 - einem jeden Datensatz aus R maximal 1 Datensatz aus S zugeordnet;
 - einem jeden Datensatz aus S ist maximal 1 Datensatz aus R zugeordnet.
- Eine 1:1-Beziehung wird realisiert, indem S den Primärschlüssel aus R als Fremdschlüssel enthält.
Dies wird dann auch der Primärschlüssel.

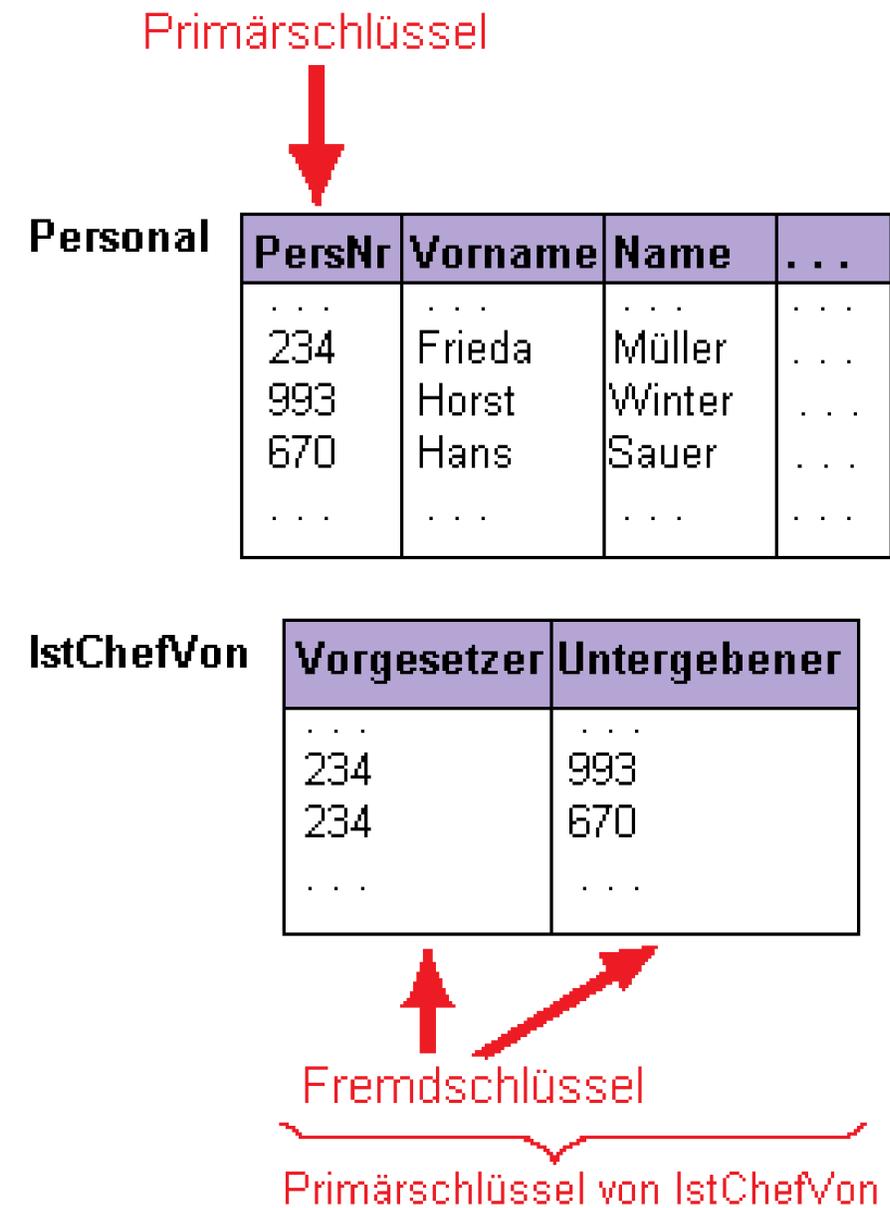
Fremdschlüssel 1:N

- Bei einer 1:N-Beziehung ist
 - einem jeden Datensatz aus R kein Datensatz, ein Datensatz oder mehrere Datensätze aus S zugeordnet;
 - einem jeden Datensatz aus S maximal 1 Datensatz aus R zugeordnet.
- Eine 1:N-Beziehung wird realisiert, indem S den Primärschlüssel aus R als Fremdschlüssel enthält. S hat jedoch einen eigenen Primärschlüssel.

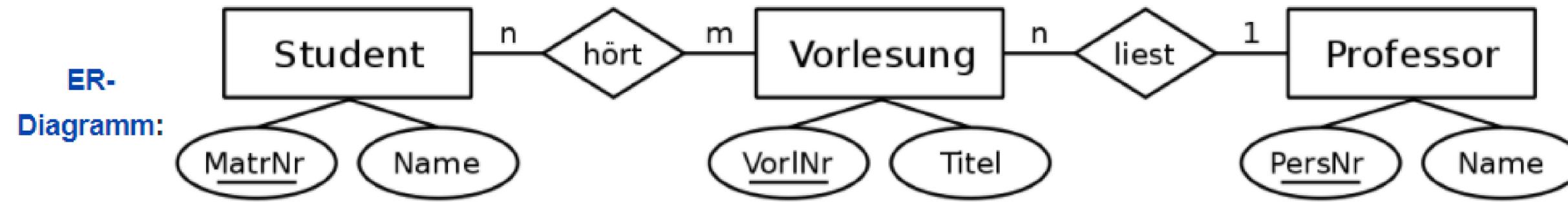
Fremdschlüssel N:M

- Bei einer n:m-Beziehung ist
 - einem jeden Datensatz aus R keiner, einer oder mehrere Datensätze aus S zugeordnet;
 - einem jeden Datensatz aus S keiner, einer oder mehrere Datensätze aus R zugeordnet.
- Eine n:m-Beziehung wird realisiert, indem man eine separate Relation (Tabelle) aufbaut, welche die beiden Primärschlüssel aus R und S als Fremdschlüssel enthält.
- Beide Attributmengen zusammen sind dann der Primärschlüssel dieser Verknüpfungsrelation.

Sonderfall: Reflexive N:M-Beziehung



Ein paar Beziehungen und Tabellen...



Relationen:

Student		hört		Vorlesung			Professor	
<u>MatrNr</u>	Name	<u>MatrNr</u>	<u>VorINr</u>	<u>VorINr</u>	Titel	<u>PersNr</u>	<u>PersNr</u>	Name
26120	Fichte	25403	5001	5001	ET	15	12	Wirth
25403	Jonas	26120	5001	5022	IT	12	15	Tesla
27103	Fauler	26120	5045	5045	DB	12	20	Urlauber

Starke und schwache Entitätstypen

**Starker
Entitätstyp**

**Schwacher
Entitätstyp**

- Schwache Entitätstypen können nicht alleine durch deren Attribute identifiziert werden.
 - Deshalb müssen Fremdschlüssel verwendet werden, um gemeinsam mit den restlichen Attributen einen Primärschlüssel zu bilden.
- Bei dem Fremdschlüssel handelt es sich hierbei üblicherweise um den Primärschlüssel des starken Entitätstyps, welche des schwachen Entitätstyps übergeordnet ist, beziehungsweise von welcher der schwache Entitätstyp abhängig ist.
- Der schwache Entitätstyp kann demzufolge nicht ohne den zugehörigen starken Entitätstyp existieren, was stark
 - an eine Komposition und
 - an eine Vererbungin der Objektorientierung erinnert.

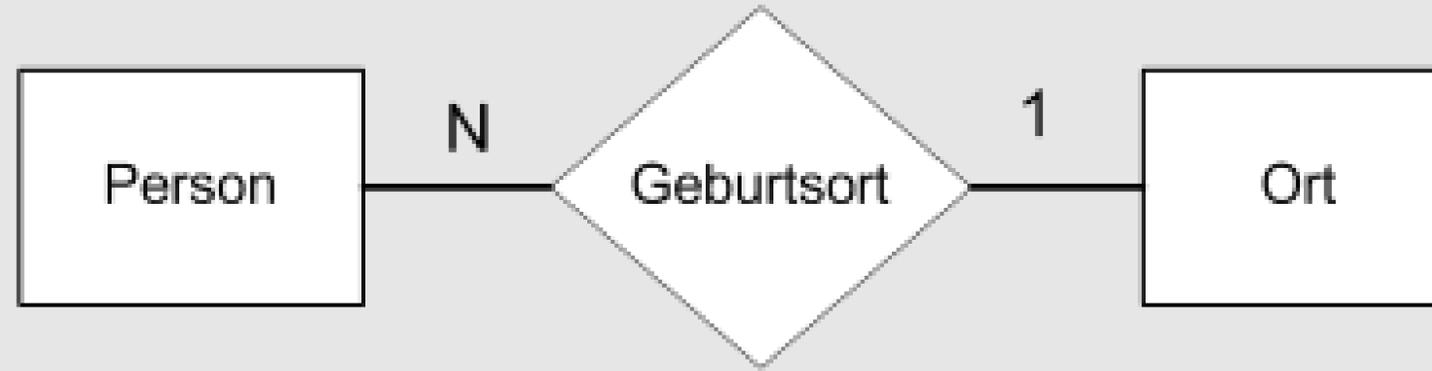
Weitere E/R-Notationen

Alternativen zur Chen-Notation

- Neben der Chen-Notation sind noch einige andere Notationen üblich...
 - UML-Klassendiagramme als Standard, den selbst die ISO in eigenen Normen als Ersatz für E/R-Diagramme verwendet.
 - Die Martin-Notation als weit verbreitete Werkzeug-Diagramm-Sprache zum Information Engineering. Diese ist auch bekannt als die Krähenfuß-Notation.
 - Die IDEF1X als langjähriger De-Facto-Standard bei US-amerikanischen Behörden.
 - Die Bachman-Notation von Charles Bachman als weit verbreitete Werkzeug-Diagramm-Sprache.
 - Die ISO-genormte (min, max)-Notation von Jean-Raymond Abrial von 1974.

Derselbe Sachverhalt in verschiedenen Notationen...

Chen



UML



Martin / IE /
Krähenfuß



Derselbe Sachverhalt in verschiedenen Notationen...

IDEF1X



Bachman

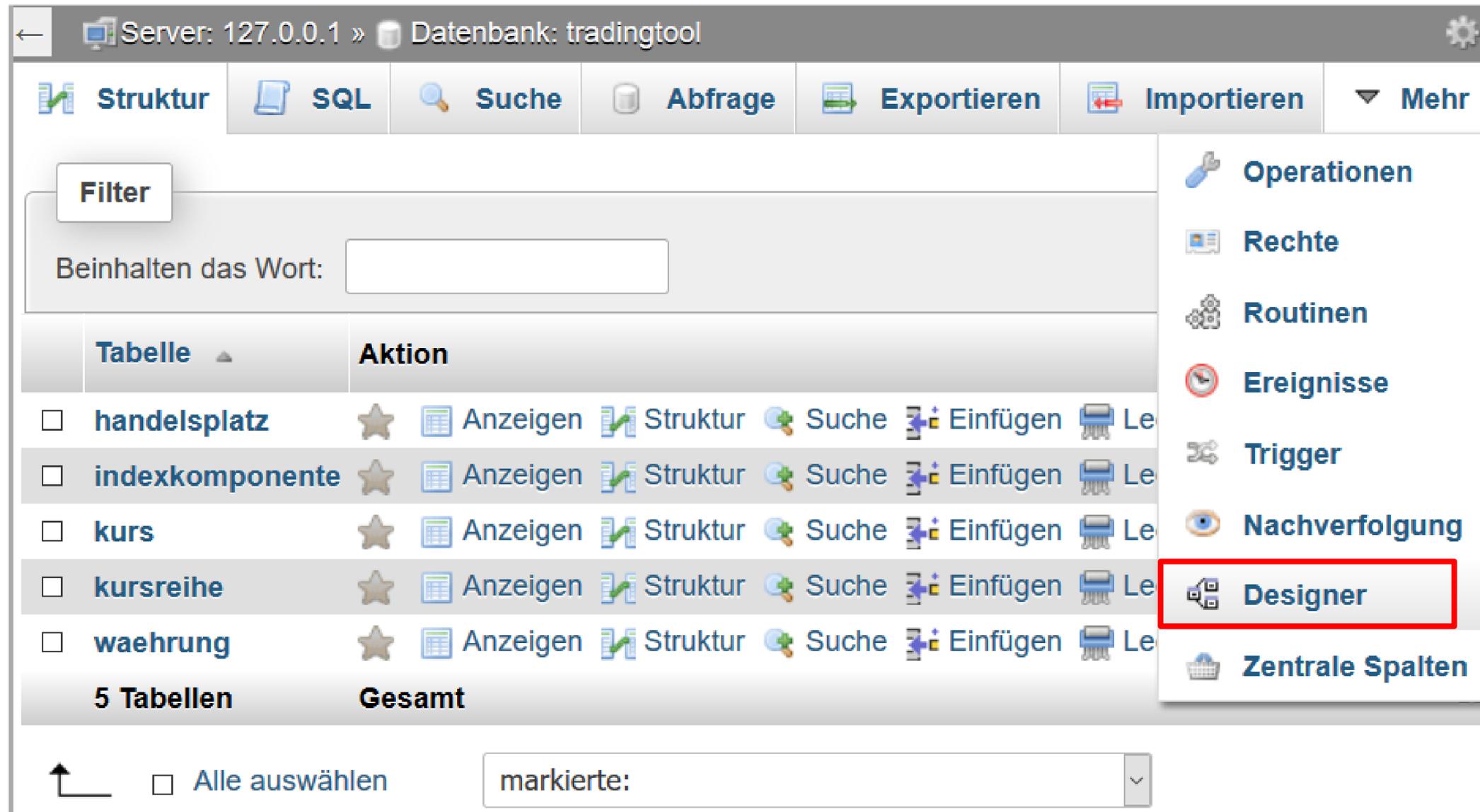


Min-Max / ISO

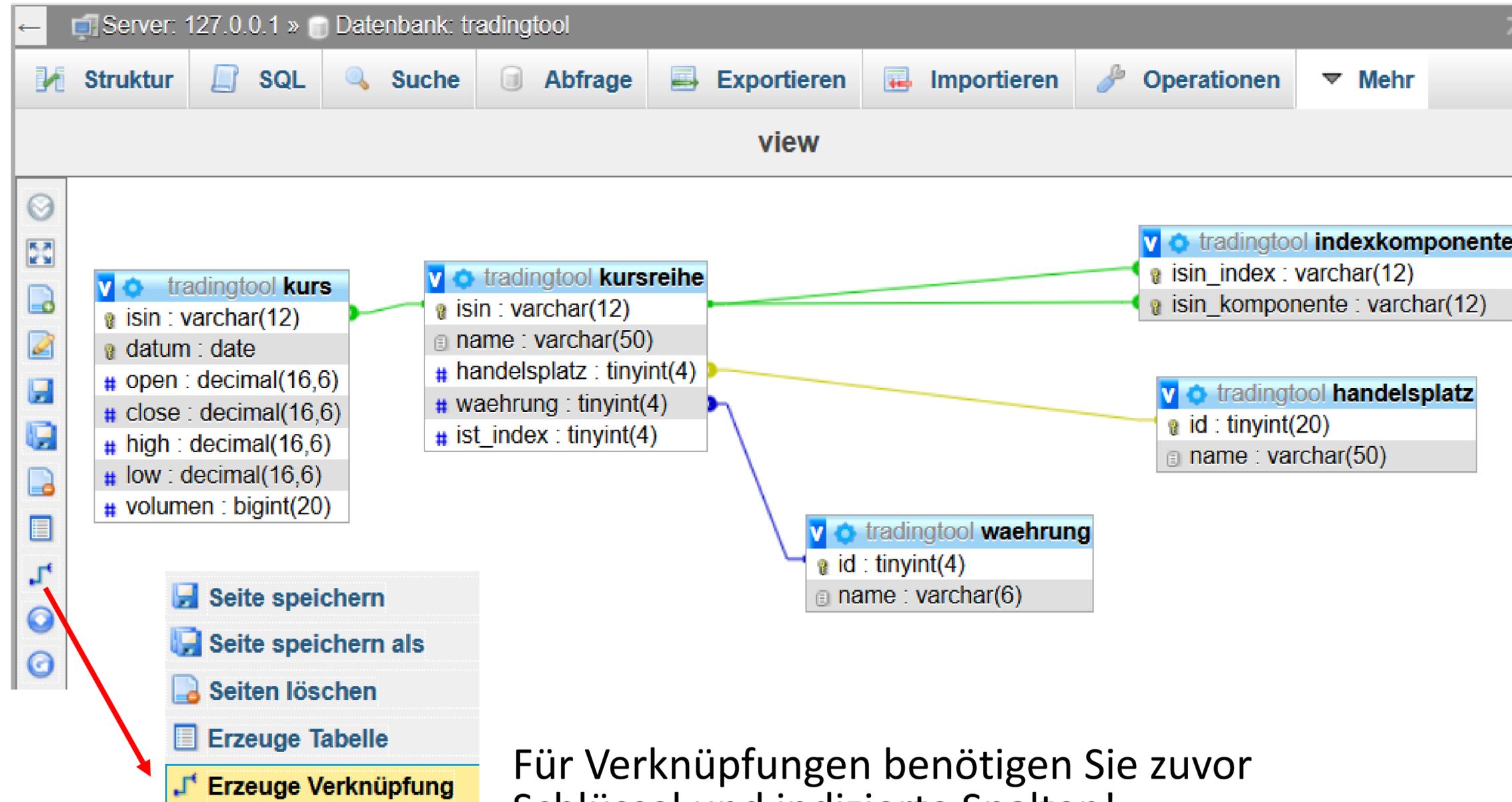


E/R-Modellierung im Designer von MariaDB

Wo findet man den Designer?



Verknüpfungen im Designer



Für Verknüpfungen benötigen Sie zuvor Schlüssel und indizierte Spalten!

Was kann man da einstellen?

- Man kann nun das Verhalten bei Fremdschlüsseln beeinflussen!
- Dabei existieren zwei Varianten, ON DELETE und ON UPDATE:
 - ON DELETE gibt an, was passieren soll, wenn ein Eintrag in der referenzierten Eltern-Tabelle gelöscht wird, auf den sich Fremdschlüssel in der Kind-Tabelle beziehen.
 - ON UPDATE legt fest, was passieren soll, wenn sich der Wert eines Primärschlüssels in der Eltern-Tabelle ändert.
- Für beide Varianten gibt es dann die dazu gehörigen Aktionen:
 - CASCADE
 - SET NULL
 - NO ACTION
 - RESTRICT

CASCADE

- ON DELETE CASCADE
 - Wenn man in der Elterntabelle einen Datensatz löscht, so werden ebenfalls alle Datensätze aus der Kindtabelle in die ewigen Jagdgründe befördert, die über den Fremdschlüssel damit verbunden sind.
- ON UPDATE CASCADE
 - Wenn man in der Elterntabelle den Inhalt eines Feldes ändert, dass in der Kindtabelle als Fremdschlüssel definiert wurde, so werden in Letzterer die Fremdschlüsselwerte ebenfalls angepasst.

SET NULL

- Hier gibt es keinen Unterschied bei ON DELETE und ON UPDATE.
- Wird also der entsprechende Eintrag in der Elterntabelle gelöscht oder geändert, so setzt MariaDB in der Kindtabelle den Wert der Fremdschlüsselspalte automatisch auf NULL.
- Allerdings darf die Fremdschlüsselspalte nicht mit NOT NULL festgelegt sein, sonst gibt es eine Fehlermeldung!

NO ACTION und RESTRICT

- NO ACTION und RESTRICT beziehen sich ausschließlich auf Primärschlüssel, die in anderen Tabellen als Fremdschlüssel benutzt werden.
 - Dies kann man übersetzen mit "mach nix":
 - bei NO ACTION wird die Gegenstelle einfach nicht berücksichtigt
 - bei RESTRICT wird beispielsweise ein Löschen verweigert