

Einführung in Datenbanken

Kapitel 16: Einführung in das Entity-Relationship-Modell

Prof. Dr. Stefan Brass

Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2022/23

<http://www.informatik.uni-halle.de/~brass/db22/>

Begriffe des ER-Modells (4)

Relationship:

- Beziehung zwischen Paaren von Entities.

Solche Relationships werden auch „binäre Relationships“ genannt, um sie von solchen zu unterscheiden, bei denen mehr als zwei Entities beteiligt sind. Manche Autoren (aber wenige Tools) erlauben beliebige Relationships (z.B. ternäre Relationships). Aus meiner Erfahrung führt das oft zu Fehlern (Studenten „verschmelzen“ zwei binäre Relationships zu einer ternären, das verletzt Normalformen).

- Z.B. Ich (ein Dozent) halte „Einführung in Datenbanken“ (eine Vorlesung).
- Das Wort „Relationship“ wird auch als Abkürzung für „Relationship-Typ“ verwendet (siehe unten).

Es sollte vom Kontext her klar sein, was gemeint ist.

Begriffe des ER-Modells (5)

Entity-Typ:

- Menge ähnlicher Entities (in Bezug auf die zu speichernden Informationen), d.h. Entities, die die gleichen Attribute haben.
- Z.B. alle Dozenten dieser Universität.

Relationship-Typ:

- Menge ähnlicher Relationships.
- Z.B. „Dozent X hält Vorlesung Y“.

Attribute: Notation (3)

- Im neueren Werkzeug „Oracle SQL Developer Data Modeler“ gibt es zwei Spalten von Symbolen:
 - In der ersten Spalte steht „#“ um die Primärschlüsselattribute zu markieren, bzw. nichts, wenn das Attribut nicht zum Primärschlüssel gehört.

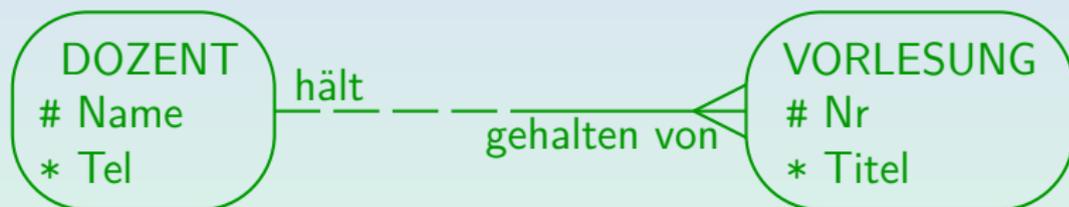
Man kann Alternativschlüssel abspeichern, aber sie werden nicht graphisch dargestellt.
 - In der zweiten Spalte steht „*“ für Attribute, die nicht Null sein können, bzw. „○“, falls das Attribut Nullwerte erlaubt.

Die theoretisch mögliche Kombination „# ○“ ist ausgeschlossen.



Relationships: Notation (1)

- Ein Relationship zwischen zwei Entity-Typen wird durch eine Verbindungslinie zwischen den Entity-Typen notiert:



- Strichelung (hier links) und Aufspaltung (Krähenfuß, hier rechts) können zunächst ignoriert werden.

Sie drücken Kardinalitäten aus, eine spezielle Art von Integritätsbedingungen für Relationships. Diese werden ab Folie 44 erläutert.

Relationships: Notation (2)

- Ein Relationship hat immer zwei Namen, einen für jede Richtung.

Man schreibt immer den Namen für die Beziehung von A nach B in die Nähe von A , und den Namen für die Beziehung von B nach A in die Nähe von B . Bei horizontalen Verbindungen ist es üblich, den Beziehungsnamen links über die Verbindungslinie zu schreiben, und rechts unter die Verbindungslinie. Das ist aber keine Vorschrift.

- Rekursive Relationships (Beziehungen von einem Entity-Typ zum gleichen Entity-Typ) sind zulässig.
 - Beispiel: Die „ist Voraussetzung für“-Beziehung zwischen zwei Vorlesungen.

Eindeutigkeits-Bedingungen

- Die Namen von Entity-Typen im Diagramm müssen eindeutig sein (d.h. jeder Kasten muss einen verschiedenen Namen enthalten).
- Ein Entity-Typ darf keine zwei Attribute mit gleichem Namen haben.

Verschiedene Entity-Typen können gleich benannte Attribute haben.

- Relationships werden über die beiden beteiligten Entity-Typen und die beiden Namen identifiziert.

Man kann also Relationship-Namen mehrfach verwenden. Es darf nur nicht zwischen den gleichen beiden Entity-Typen zwei Relationships geben, die in beiden Namen übereinstimmen.

Geeignete Namen (1)

Entity-Typen:

- Üblich sind Substantive für Entity-Typen.
- Ich bevorzuge die Singularform (einzelne Entities).

Manche Autoren verwenden Pluralformen. Nach der Übersetzung ins relationale Modell sind diese Namen wohl passender (Wenn man aber an die Deklaration von Tupelvariablen in SQL denkt, passt die Singularform besser.) Oracle Designer nimmt den Singular für Entity-Typen und den Plural für die dazugehörigen Tabellen. Wichtig ist nur, dass man konsistent einen Stil verwendet.
- Keine Namen wie „Kundendaten“ verwenden: Das Ziel ist ein Modell von der realen Welt, nicht von der Datenbank.

Geeignete Namen (2)

Relationships:

- Oft werden Verben für Relationship-Namen verwendet (in der Umkehrrichtung dann im Passiv).

Z.B. „Dozent hält Vorlesung“, „Vorlesung gehalten von Dozent“. Eine Kombination aus Substantiv und Präposition ist auch häufig. Z.B. „Dozent von“ (kurz für „ist Dozent von“), umgekehrt „von“.

- Man sollte vermeiden, in beiden Richtungen nur eine sehr unspezifische Präposition zu haben.

Da in der Barker-Notation Relationships über beide Entity-Typen und die Namen beider Richtungen des Relationships identifiziert werden, wäre unspezifische Namen aber möglich, wenn es zwischen den beiden Entity-Typen nur eine natürliche Beziehung gibt.

Attribut vs. Relationship

- Im Prinzip würden „Fremdschlüssel-Attribute“ die gleiche Information wie ein Relationship enthalten:

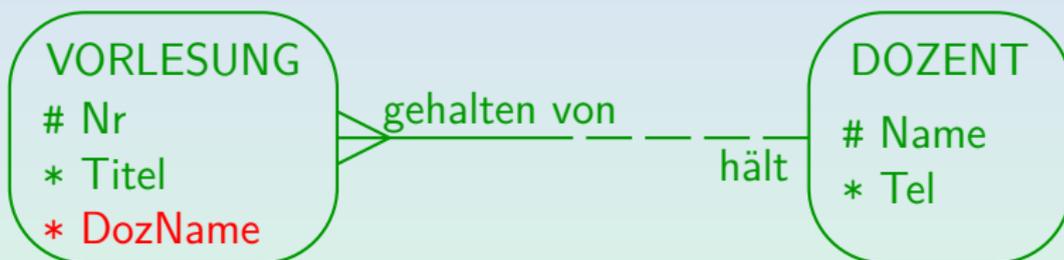


- Das ist aber im ER-Modell schlecht/falsch (Punktabzug!).

Relationships sollten explizit dargestellt werden. Fremdschlüssel werden im ER-Modell nicht speziell unterstützt. Man müsste sie mindestens als Integritätsbedingung dazu schreiben. Man verliert so aber die visuelle Repräsentation und nutzt die ER-Konstrukte schlecht.

Vermeidung von Redundanzen (1)

- Man sollte Attribute, die über ein Relationship erreicht werden können, nicht wiederholen:



- Viele Studenten, die bereits Erfahrung im relationalen DB-Entwurf haben, machen diesen Fehler.

Es wird ein solche Spalte nach der Übersetzung ins relationale Modell geben. Aber das relationale Modell hat keine Relationships. Ein Relationship und ein solches Attribut zu haben, ist redundant.

Vermeidung von Redundanzen (2)

- Zyklen in einem Diagramm sollten genau auf mögliche Redundanzen untersucht werden:



Es ist falsch, die Komposition zweier Relationships wieder als Relationship zu definieren. Natürlich sind Zyklen nicht immer redundant oder schlecht, aber oft existiert mindestens eine Abhängigkeit, so dass man eine Integritätsbedingung formulieren sollte.

Diagramme und Schemata (1)

- In den ER-Diagrammen werden nicht alle Schema-Informationen dargestellt:
 - Attribute haben Datentypen, die in vollständigen ER-Schemata festgelegt werden müssen.
 - Nur ein Schlüssel pro Entity-Typ darstellbar.
 - Große ER-Diagramme werden übersichtlicher, wenn die Attribute nicht angezeigt werden.

Es ist nicht ungewöhnlich, dass ein Entity-Typ 50 Attribute hat.
 - Kommentare/Erklärungen sind nützlich, sehen aber in ER-Diagrammen nicht gut aus.

Diagramme und Schemata (2)

- Es gibt also ein „wirkliches Schema“ (z.B. in Textform oder in einer DB). ER-Diagramme sind nur Auszüge, die zur Illustration genutzt werden.

Es gibt keinen Standard für eine Syntax, um ganze Schemata aufzuschreiben, dagegen gibt es für ER-Diagramme bekannte Notationen.

- Wichtig ist, die graphische Syntax zu lernen (und dass möglicherweise nicht alles dargestellt wird).
- Moderne Entwicklungs-Tools lösen das Problem: Z.B. zeigt ein Klick/Doppelklick auf einen Entity-Typ im Diagramm weitere Informationen in einer Dialog-Box.

ER-Schemata: Semantik (2)

- Dies ist ein Spezialfall der logischen Interpretation, die in Kapitel 5 vorgestellt wurde.

Die logische Interpretation enthält formal auch die Interpretation der Datentypen, diese ist aber fest (gehört nicht zum Zustand).

- Die Haupteinschränkungen des ER-Modells sind:

- Es können nur neue Sorten mit endlichen Domains eingeführt werden (Entity-Typen).

Die Datentypen sind ja vorab fest gegeben (ins DBMS eingebaut).

- Neue Funktionen nur von Entity-Typen zu Datentypen (Attribute).
- Neue Prädikate müssen zwei Entity-Typen als Argumente haben (Relationships).

ER-Schemata: Semantik (3)

Beispiel-Zustand:

- $\mathcal{I}[\text{Dozent}] = \{sb, ms\}$.
- $\mathcal{I}[\text{Name}] = f_1$, mit $f_1(sb) = \text{'Brass'}$, $f_1(ms) = \text{'Spring'}$.
- $\mathcal{I}[\text{Tel}] = f_2$, mit $f_2(sb) = 49404$, $f_2(ms) = 49429$.
- $\mathcal{I}[\text{Vorlesung}] = \{db, ds, dp\}$.
- $\mathcal{I}[\text{Nr}] = g_1$, mit
 $g_1(db) = 20727$, $g_1(ds) = 42232$, $g_1(dp) = 40492$.
- $\mathcal{I}[\text{Titel}] = g_2$, mit
 $g_2(db) = \text{'DB'}$, $g_2(ds) = \text{'Data Str.'}$, $g_2(dp) = \text{'Doc. Proc.'}$.
- $\mathcal{I}[\text{hält}] = \{(sb, db), (sb, ds), (ms, ds), (ms, dp)\}$.

Inhalt

- 1 Einführung
- 2 Grundlegende ER-Konstrukte
- 3 Schlüssel
- 4 Kardinalitäten**
- 5 Schwache Entities

Kardinalitäten: Minimum (1)

- Umgekehrt kann man auch die minimale Anzahl von Entities des anderen Typs angeben, zu dem ein Entity eine Beziehung haben muss.
- Die Barker-Notation unterstützt nur 0 und 1:
 - Wenn Entities des Typs E_1 nicht an der Beziehung teilnehmen müssen, wird die Hälfte der Verbindungslinie auf der Seite von E_1 gestrichelt.
 - Wenn Entities vom Typ E_1 zu mindestens einem Entity des Typs E_2 in Beziehung stehen müssen, wird die Linie auf der E_1 -Seite durchgezogen.

Kardinalitäten: Minimum (2)



- Ein Dozent muss hier nicht unbedingt eine Vorlesung halten.

Die gestrichelte Linie auf seiner Hälfte zeigt, dass die Teilnahme von „DOZENT“-Entities an der Beziehung optional ist.

- Aber eine Vorlesung muss immer eine Beziehung zu einem Dozenten haben.

Die durchgezogene Hälfte auf der Vorlesungs-Seite zeigt, dass die Teilnahme von „VORLESUNG“-Entities an der Beziehung verpflichtend ist (ob das in der Realität immer so ist, ist eine andere Frage).

Kardinalitäten: Minimum (3)

- Selbstverständlich können auch beide Hälften der Verbindungslinie gestrichelt oder beide durchgezogen sein:



- Hier ist nicht verlangt, dass ein Student mindestens eine Vorlesung hört. Es ist auch nicht verlangt, dass eine Vorlesung mindestens einen Teilnehmer hat.

Eine Vorlesung ohne Teilnehmer wird gestrichen, wenn das Semester begonnen hat. Aber wenn sie angekündigt wird, ist völlig normal, dass sie zunächst keine Teilnehmer hat (temporäre Situationen bedenken!).

Kardinalitäten: Allgemein (2)

- Wie oben erläutert, sind in der Barker-Notation nur die häufigsten Fälle darstellbar:
 - Minimum 0 oder 1
 - Maximum 1 oder unbeschränkt
- Wenn man ein E_1 -Entity festhält und die zugehörigen E_2 -Entities zählt, wird
 - das Minimum auf der Seite von E_1 dargestellt (gestrichelt oder durchgezogen),
 - das Maximum auf der Seite von E_2 (aufgefächert mit Krähenfuß oder nicht).

Inhalt

- 1 Einführung
- 2 Grundlegende ER-Konstrukte
- 3 Schlüssel
- 4 Kardinalitäten
- 5 Schwache Entities**

Schwache Entities (1)

- Ein Entity kann eine Art „Detail“ beschreiben, das ohne „Master“-Entity nicht existieren kann.
- Dann gibt es eine eins-zu-viele Beziehung („1:n“) vom Master-Entity-Typ zum Detail-Entity-Typ.

Jedes Detail-Objekt gehört zu genau einem Master-Objekt.

- Außerdem wird der Schlüssel des Master-Entities ein Teil des Schlüssels des Detail-Entities:



Schwache Entities (5)

- Das Entscheidende bei schwachen Entity-Typen ist die spezielle Konstruktion des Schlüssels:
 - Der eigene Schlüssel ist eindeutig nur im Kontext eines anderen Entities.
 - Daher muss man den Schlüssel dieses anderen Entity-Typs übernehmen (und noch erweitern).
- Im Kontext einer gegebenen Rechnung reicht die Positions-Nr (1, 2, ...), global ist sie aber nur zusammen mit einer Rechnungs-Nummer eindeutig.

Genauso: Im Kontext eines gegebenen Gebäudes reicht die Raum-Nummer, global braucht man zusätzlich eine Gebäude-Identifikation.

Schwache Entities (6)

- In der Barker-Notation wird ein schwacher Entity-Typ durch einen Querstrich an der Beziehung zum übergeordneten Entity gekennzeichnet:



- Das Schlüsselattribut „RNr“ des übergeordneten Entity-Typs wird beim schwachen Entity-Typ nicht angegeben.

Über die Beziehung ist aber immer genau eine Rechnung erreichbar, die dann eine bestimmte Rechnungsnummer hat. Es gibt also die gleiche Information wie vorher, nur nicht mehr redundant.

Schwache Entities (9)

- Ein schwaches Entity muss weitere Schlüsselattribute zu den geerbten hinzufügen.

Wenn die Schlüssel der beiden Entity-Typen die gleichen wären, sollte eine Spezialisierung verwendet werden (→ Vorlesung über DB-Entwurf).

Zumindest ist es dann eine „1:1“-Beziehung (nicht „1:n“).

- Manchmal wird das Master-Entity „Eltern-Entity“ und das abhängige schwache Entity „Kind-Entity“ genannt.
- Entities mit eigenem Schlüssel (nicht-schwache Entities) werden reguläre/starke Entities genannt.

Schwache Entities (10)

- Schwache Entities sind normale Entities, außer dass ihr Schlüssel auf spezielle Art gebildet wird.
- Somit können sie auch normale Relationships haben (neben der identifizierenden Beziehung):



- Schwache Entities können selbst Master-Entities für andere schwache Entities sein.

Es kann eine ganze Hierarchie von Master-Detail-Relationships geben.
Aber Zyklen sind verboten.

Schwache Entities (11)

Aufgabe:

- Modellieren Sie Tests für ein E-Learning Angebot im Web (mehrere Multiple-Choice-Tests).
- Jeder Test wird durch einen Titel, jede Frage in einem Test durch eine Zahl und jede Antwort zu einer Frage durch einen Buchstaben identifiziert.

Für jede Frage und Antwort muss der Text gespeichert werden und Antworten müssen in richtige und falsche klassifiziert werden.

Association-Entities (4)

