

Zusammenfassung Modul 162

ERD und ERM

Ein ERD ist immer ein einzelnes Diagramm, wobei ein ERM eine Sammlung von Diagramme sein kann.

ERD: "Entity-Relationship Diagram"

ERM: "Entity-Relationship Model"

Ein ERD besteht aus Entitäten und Beziehungen (Assoziationen)

Entität: Ein Objekt welches Daten enthält, die im Detail spezifiziert werden können. Die Entität "Lehrperson" kann zum Beispiel die Attribute "Vorname", "Nachname", "Alter" etc haben.

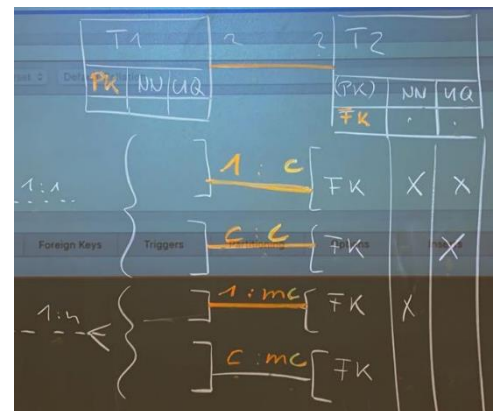
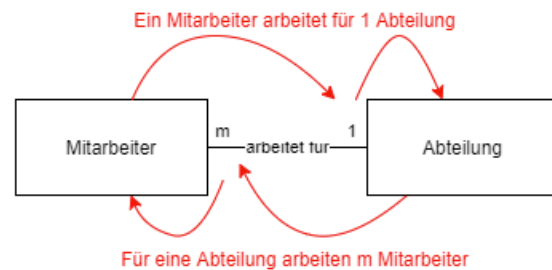
Beziehung / Assoziation: Eine Beziehung zwischen Entitäten wird durch einen Strich gekennzeichnet. Im Beispiel sehen wir, dass alle Entitäten eine Assoziation zueinander besitzen. Dies ist aber keinesfalls immer der Fall (keine, mehrere). Auch Assoziationen können weiter spezifiziert werden.

Abkürzung	Erklärung
1	Genau ein Datensatz
c	Kein oder genau ein Datensatz
m	Mindestens ein Datensatz. Kann auch unendlich viele haben.
mc	Beliebig viele Datensätze. Also kein, ein oder unendlich viele

Kardinalität: Die Kardinalität ist eine Mengenangabe und sagt aus, wie viele Entitäten des einen Typs mit den Entitäten des anderen Typs in Beziehungen stehen können. Daraus folgt auch, dass Assoziationen in beide Richtungen bestehen (aber nur ein Strich gezeichnet wird).

- Hierarchischen Beziehungen - in der Abbildung dunkelgrau
- Konditionellen Beziehungen - in der Abbildung hellgrau
- Netzwerkförmigen Beziehungen - in der Abbildung weiss

	1	c	m	mc
1	1-1	1-c	1-m	1-mc
c	c-1	c-c	c-m	c-mc
m	m-1	m-c	m-m	m-mc
mc	mc-1	mc-c	mc-m	mc-mc

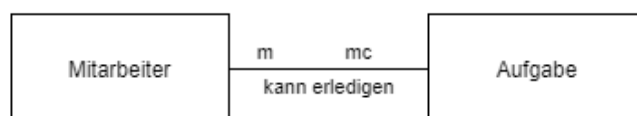


Datenmodell

Konzeptionell Datenmodell

- Entitäten werden oft ohne Attribute dargestellt, können aber Attribute enthalten
- Entitäten können netzwerkförmige Beziehungstypen haben, also m(c):m(c).
- Verwenden sie draw.io oder andere einfache Tools

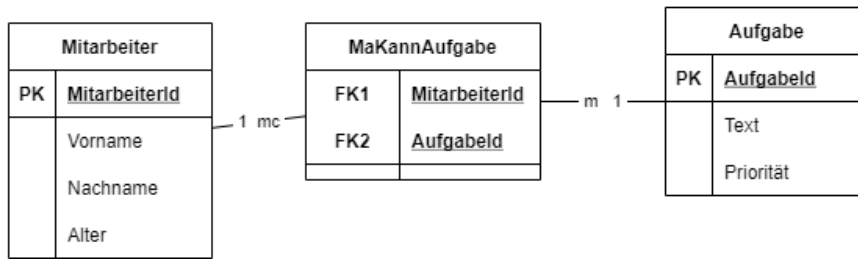
Beispiel:



Logisches Datenmodell

- Attribute sind definiert.
- Alle netzwerkförmigen Beziehungstypen sind aufgelöst
- Datentypen können definiert sein, aber werden allgemein gehalten und es werden nicht DBMS-spezifische Typen verwendet (z.B. String und nicht varchar(x)).
- Verwenden sie draw.io oder andere einfache Tools

Beispiel:



Physisches Datenmodell:

- Alle Regeln eines logischen Schemas werden eingehalten
- Datentypen und andere Eigenschaften eines spezifischen DBMS werden konkret verwendet
- Verwenden sie ein DBMS spezifisches Tool

Beispiel:

Tabellename **Mitarbeiter** Attribut, Spalte, Column

MitarbeiterId	Vorname	Nachname	Alter	Abteilungskürzel
1	Gustav	Meier	22	MA
2	Ida	Kuhler	34	MA
3	Sylvia	Brendel	28	IT
4	Walter	Morani	55	IT
5	Beat	Merz	18	IT

Datensatz, Record, Row, Tupel

Das Gesamte wird als Tabelle bezeichnet Feld, Field, Wert, Value

Keys

Primärschlüssel

Jede Entität benötigt ein Attribut oder eine Kombination von Attributen, die ein Tupel eindeutig identifiziert. Dieses Attribut wird **Primärschlüssel (PS) oder Primary Key (PK)** genannt.

Fremdschlüssel

Alle hierarchischen und konditionellen Beziehungen benötigen einen **Fremdschlüssel (FS) oder Foreign Key (FK)**, der die Beziehung definiert und auf den PK der anderen Entität zeigt. Im folgenden Beispiel ist dies das Attribut *Abteilungskürzel*. **Der FK wird immer auf der Seite der Beziehung eingeführt, die die m oder mc Beziehung hat.** \$

Auflösung von netzwerkförmigen Beziehungen m(c) : m(c)

- ✓ Die Beziehung der ursprünglichen Entitäten fällt weg
- ✓ Es wird eine neue Entität hinzugefügt (Transformationsentität)
- ✓ Es werden neue Beziehungen zwischen den ursprünglichen Entitäten und der neu erstellten Transformationsentität hinzugefügt.
- ✓ Die Kardinalitäten werden verschoben in der

Position.

