

2 Database Design: das ER - Modell

Konzeptueller Design: ab jetzt betrachten wir das Entity-Relationship-Modell

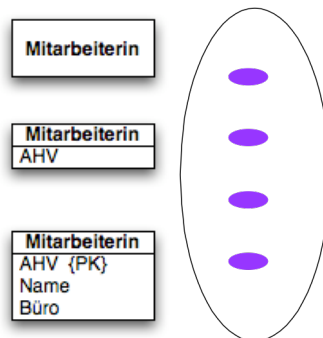
- Was sind Entitäten, was sind Relationen in ein Unternehmen?
- Welche Informationen über diese Entitäten und Relationen sollen wir in der Datenbank speichern?
- Welche Integritätseinschränkungen sollen überprüft werden?

Mit dem ER-Modell kann ein Datenbankschema graphisch dargestellt werden (ER-Diagramme, UML)

ER-Diagramme können dann in einem relationalen Schema abgebildet werden

2.1 Elemente des ER-Modells

2.1.1 Entitäten



Figur 2: Entitäten in UML und ihre Mengendarstellung

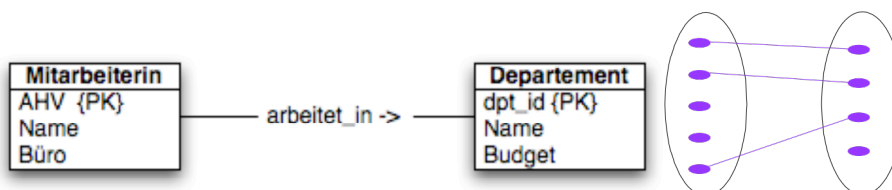
Entität: Ein reelles Objekt, das von anderen Objekte unterschieden werden kann. Eine Entität ist in der DB als eine Menge von Attributen beschrieben.

- Beispiel: ein Mitarbeiter, die Abteilung des Mitarbeiters, der Abteilungsleiter, die Adresse des Abteilungsleiters,

Entitätsmenge: Eine Menge von ähnlichen Entitäten, z. B. alle Mitarbeiter.

- Alle Entitäten in einer Entitätsmenge haben die gleiche Attributsmenge (allerdings nur bis wir die ISA-Hierarchien betrachten)
- Jede Entitätsmenge besitzt eine Schlüssel *[key]*.
- Jedes Attribut hat ein Wertebereich *[domain]*.

2.1.2 Relationen



Figur 3: Darstellung einer Relation in UML und als Menge

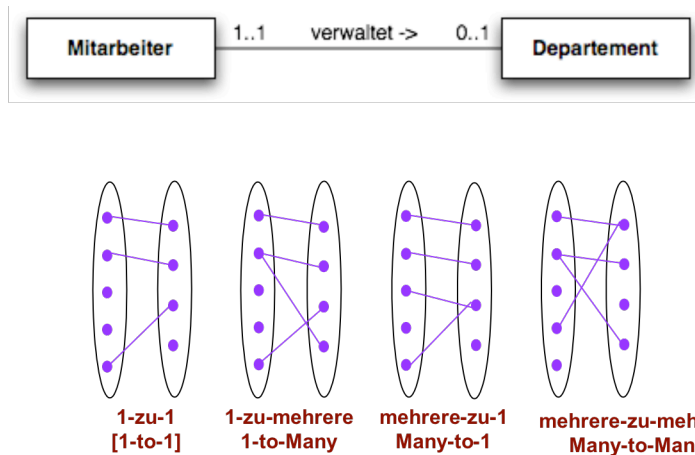
Relation: Assoziation zwischen zwei oder mehr Entitäten, z.B. Peter arbeitet im Informatikdepartement.

Relationsmenge: Menge von ähnlichen Relationen

-Eine n-fache Relationsmenge stellt n Entitätsmengen E1 ... En in Beziehung. Jede Relation in R involviert die Entitäten e1 von E1, ..., en von En.

-Eine gleiche Entitätsmenge kann in eine Relationsmenge mehrmals vorkommen, oder verschiedene Rollen in der gleichen Menge übernehmen.

2.1.3 Schlüsseleinschränkungen [key constraints]

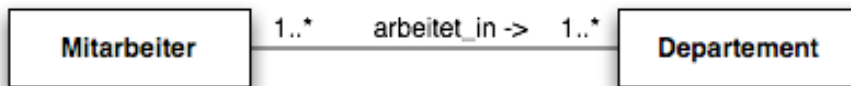


Figur 4: Schlüsseleinschränkungen in UML

-Man betrachtet eine Relation „arbeitet in“: Ein Mitarbeiter/in kann in verschiedene Departemente arbeiten, ein Departement kann mehrere Mitarbeiter haben.

-Andererseits kann jedes Departement höchstens ein Manager haben. Dies wird mit der Schlüsseleinschränkung auf die Relation „verwaltet“ bezogen.

2.1.4 Teilnahmeeinschränkungen [participation constraints]



Figur 5: Teilnahmeeinschränkung in UML

Muss jedes Departement ein Manager oder eine Managerin haben?

Falls ja, ist es eine Teilnahmeeinschränkung: die Teilnahme von „Departement“ in „verwaltet“ ist *total* genannt (sonst *partiell*). Diese Einschränkung ist in Fig. 5 abgebildet.

Jeder *did*-Wert in der Departement-Tabelle muss in ein Datensatz der „verwaltet“-Beziehung vorkommen

Weitere mögliche Einschränkung: Jeder Mitarbeiter muss mindestens in einem Departement arbeiten

2.1.5 Schwache Entitäten [Weak Entities]

Eine *schwache Entität* kann nur durch den Hauptschlüssel von einer anderen Entität (sogenannte Besitzerin [owner]) identifiziert werden.

Besitzerin-Entität und schwache Entität müssen durch eine 1-zu-mehrere Relationsmenge verbunden sein (eine Besitzerin, mehrere schwache Entitäten)

Die schwache Entität muss eine totale Teilnahme in diese Relationsmenge [identifying]

2.1.6 ISA ('is a') Hierarchien

Überlappungseinschränkungen [overlapping constraint]: Kann Sabine eine Stunden_Mitarbeiterin und eine Vertrags_Mitarbeiterin sein? (*Erlaubt/nicht erlaubt*)

Überdeckungseinschränkung: Müssen alle Mitarbeiter auch Stunden_Mitarbeiter und Vertrags_Mitarbeiter sein? (*Ja/nein*)

Gründe zur Benutzung von ISA Hierarchien:

- Um beschreibende Attribute, die spezifisch zu einer sub-klasse sind, zu addieren
- Um Entitäten, die in einer Relation teilnehmen, zu identifizieren

2.1.7 Aggregation

Wird benutzt, wenn wir eine Relation modellieren wollen, wo eine Gruppe von Beziehungen und Entitäten vorkommen.

Aggregation erlaubt uns, diese Gruppe von Relationen und Entitäten als Einheit zu betrachten, die in anderen (Gruppen von) Relationen teilnehmen.

2.2 Konzeptueller Design mit dem ER-Modell

Entscheidungen für den Design:

- Soll ein Konzept mit einer Entität oder mit einem Attribut modelliert werden?
- Soll ein Konzept mit einer Entität oder mit einer Relation modelliert werden?
- Wie sollen Relationen indentifiziert werden? Binär? Ternär?

Einschränkungen im ER-Modell:

- Möglichst viele semantische Kriterien sollen (und müssen) im ER-Modell einbezogen werden
- Einige Einschränkungen können aber nicht mit ER-Diagramme (oder UML Diagramme) beschrieben werden

2.2.1 Entitäten oder Attribute?

Soll „Adresse“ ein Attribut von „Mitarbeiter“ oder eine Entität für sich selbst sein?

- Entität ist dann in Verbindung mit „Mitarbeiter“ durch eine Relation
- Es hängt ab, wie man die Information über die Adresse benutzen will, also von der Semantik der Daten
- Falls wir mehrere Adressen per Mitarbeiter haben wollen, muss „Adresse“ eine Entität sein, weil Attribute nicht zu einer Menge gehören können
- Falls die Struktur der Adresse (Stadt, Strasse, ...) wichtig ist, z.B. falls wir die Namen aller Mitarbeiter in einer bestimmten Stadt zugreifen können wollen, muss Adresse mit einer Entität modelliert werden, weil Attributswerte atomar [atomic] sind

Für die Entität „Mitarbeiter“ haben wir ein ähnliches Problem wie für „Adresse:“ wir wollen die Möglichkeit haben, *mehrere Attributswerte für jede Instanz dieser Beziehung* aufzunehmen. Dies ist gelöst, indem wir die Entität „Dauer“ einführen

2.2.2 Entität oder Relation?

-Das erste ER-Diagramm ist OK, falls ein Leiter oder eine Leiterin ein separates Budget für jedes Departement bekommt

-Was passiert, falls ein Leiter oder eine Leiterin ein Budget für alle Departemente in seiner oder ihrer Verwaltung bekommt?

-Das Attribut „Budget“ wird redundant, d.h. für jedes Departement wird es erneut gespeichert

-Verwirrend: es weist darauf hin, dass dBudget mit dem verwalteten Departement verbunden ist