

1. Datenmodellierung und Datenbanken

1.1. Wdh. Modellbegriff

Def. Modell

= ist eine abstrakte Beschreibung eines realen oder geplanten Systems. Dabei werden bewusst bestimmte Merkmale vernachlässigt, um für den Modellierer wesentliche Eigenschaften hervorzuheben.

Def. Datenmodell

= Modell zur Veranschaulichung von Datenstrukturen und deren Beziehungen.

1.2. Begriffe bei Datenbanksystemen

Datenbanksystem (DBS) = Einheit aus DB und DBMS

Datenbank (DB) = Sammlung von Daten zu einem Problembereich

Datenbankmanagementsystem (DBMS) = Software zur Eingabe, Verwaltung, Auswertung und Ausgabe von Daten der DB

Bsp.:

DBMS: LibreOffice Base, MS Access, MySQL, Oracle, ...

DB: Bücher-DB, Schüler-DB, Einwohnermelderegister, Lagerverwaltung, Kunden-DB, Patienten-DB, DB der Sparkassen, Verkehrszentralregister, ...

- von Objekten wie eine Person, ein Gegenstand oder ein Ereignis werden in einer DB **Attribute** (Merkmale, Eigenschaften) erfasst
- zur Speicherung dieser Attribute dienen **Datenfelder**
- die Aneinanderreihung der Datenfelder eines Objekts ergibt einen **Datensatz**

Bsp.:

Objekt: eine Person mit den Attributen Name, Vorname, Geb.datum, Wohnort

Müller	Max	12.11.1999	Freiberg
--------	-----	------------	----------

Aufgaben eines DBMS

- a) die Durchführung der Datenspeicherung,
- b) die Verwaltung der Daten auf Basis des zugrunde gelegten Datenmodells
- c) Widerspruchsfreie Speicherung, Synchronisation bei Mehrbenutzerbetrieb
- d) Datenschutz, Zugriffsbeschränkungen (Passwort)
- e) Effizientes Verarbeiten großer Datenmengen, Schutz vor Datenverlust

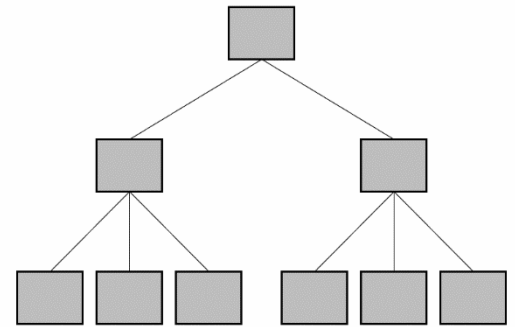
In welcher Struktur könnten folgende Datensammlungen organisiert werden?

- a) Daten von Angehörigen einer Familie → Stammbaum
- b) Verwaltung von Stücklisten für Bauteile und Baugruppen von Maschinen → Netzstruktur
- c) Daten von Lesern und Büchern einer Bibliothek → Tabelle

1.3. Arten von Datenmodellen

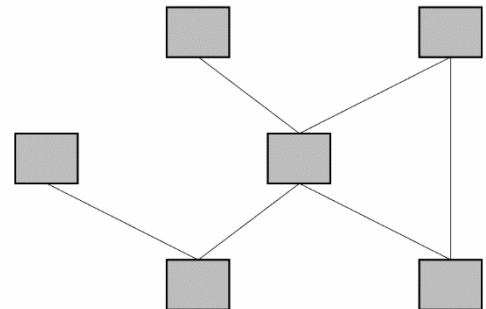
- **Hierarchisches Datenbankmodell**

- ältestes Datenbankmodell
- bildet die reale Welt durch eine hierarchische Baumstruktur ab
- jeder Satz (Record) hat *genau* einen Vorgänger, mit Ausnahme *genau eines* Satzes, nämlich der *Wurzel*
- Baumstruktur lässt nur 1:1 und 1:n-Beziehungen zu
- Verknüpfungen zwischen den Datensatzabbildern werden als Eltern-Kind-Beziehungen (Parent-Child Relationships, PCR) realisiert
- **Nachteil:** Verknüpfungen zwischen verschiedenen Bäumen oder über mehrere Ebenen innerhalb eines Baumes sind nicht möglich
- hierarchische Modell ist im Bereich der Datenbanksysteme heute weitgehend von anderen Datenbankmodellen abgelöst
- Dateisysteme vieler BS sind hierarchisch aufgebaut



- **Netzwerkdatenbankmodell**

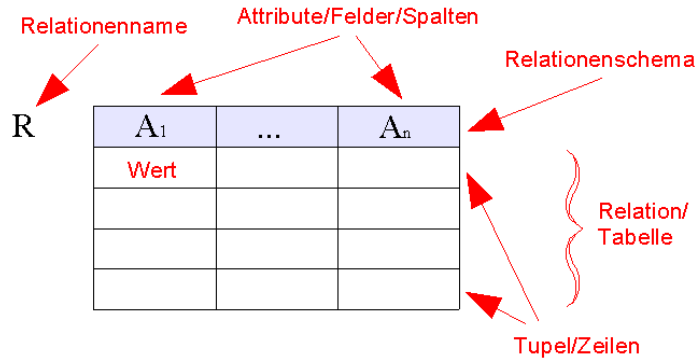
- fordert keine strenge Hierarchie sondern kann auch m:n-Beziehungen abbilden
- d.h. ein Datensatz kann mehrere Vorgänger haben
- es können mehrere Datensätze an oberster Stelle stehen
- es existieren meist unterschiedliche Suchwege, um zu einem bestimmten Datensatz zu kommen



- heute hauptsächlich auf Großrechnern eingesetzt
- **Nachteile:** Wartung ist schwierig und aufwändig, komplizierte Modellierung, hoher Speicherbedarf

- **Relationales Datenbankmodell (= Tabellenmodell)**

- wurde 1970 von Edgar F. Codd erstmals vorgeschlagen und ist bis heute, ein etablierter Standard für Datenbanken
- Grundlage des Konzeptes relationaler Datenbanken ist die Relation (= Beschreibung einer Tabelle)
- Relationale DB = Sammlung von Tabellen
- Zeilen sind die Datensätze
- Spalten sind die Datenfelder
- Relationenschema legt die Anzahl und den Typ der Attribute fest
- Manipulation der Daten erfolgt über die Datenbanksprache SQL
- sind vergleichsweise einfach und flexibel zu handhaben
- **Nachteil:** Werte eines Datenfeldes müssen immer vom gleichen Datentyp sein (entweder Text, oder Grafik oder Datum)



1.4. Relationale Datenmodellierung

LB DUDEN SII S. 155 – 157

Ausarbeitung zu ERM/ ERD und Kardinalitäten

Begriffe klären:

- Entity-Relationship-Modell (ERM)
dient der Modellierung großer Datenmengen (z.B. in Worten)
- Entity-Relationship-Diagramm (ERD)
ist die grafische Darstellung des ERM
- Entität
ist ein eindeutig zu bestimmendes Objekt
- Entityklasse (EK)
Sammlung von Datenobjekten mit gemeinsamen Eigenschaften (Attribut, Merkmal)
- Attribut
ist ein Merkmal/Eigenschaften eines (eindeutigen) Objektes
- Schlüsselattribut
dient zur eindeutigen Identifizierung von Datenobjekten/Entitäten (Kennzeichen, ISBN, IBAN, Handynummer, Kundennummern)
- Relationship
Beziehung zwischen den Entityklassen (EK)
- Arten von Kardinalitäten (siehe M1)
 - 1:1 zu einem Objekt aus EK1 gehört genau ein Objekt aus EK2
 - 1:n zu einem Objekt aus EK1 kann man beliebig viele Objekte aus EK2 zuordnen
 - n:m beliebig viele Objekte aus EK1 kann man beliebig viele Objekte aus EK2 zuordnen

Wenden Sie die Erkenntnisse auf folgendes Beispiel an!

Klassen – Schüler – Arbeitsgemeinschaften

1 : n n : m

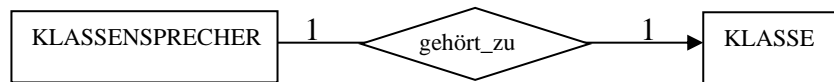
EK: Klassen Schüler Arbeitsgemeinschaften
 Entität: Namen, Schüler
 Attribut: SName...
 Schlüsselat: SNr
 Relationsh.: geht in besucht

M1 Beziehungs-Kardinalitäten

a) 1:1 – Kardinalität

→ zu einem Objekt der Entityklasse 1 (E1) gehört genau ein Objekt aus E2

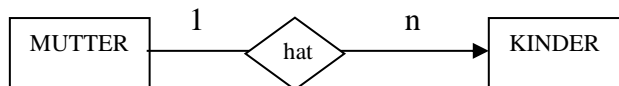
Bsp.:



b) 1: n - Kardinalität

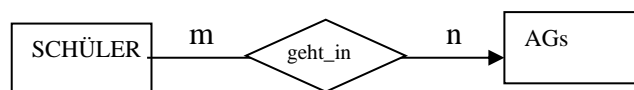
→ zu einem Objekt der Entityklasse 1 (E1) gehören mehrere Objekte aus E2

Bsp.:



c) n : m - Kardinalität

→ zu mehreren Objekten der Entityklasse 1 (E1) gehören mehrere Objekte aus E2



(theoretisch in jede Richtung 1:n → wird zu n:m)

(beliebig viele Schüler können in beliebig viele AG's gehen)

Entwickeln Sie ein vollständiges ERD für folgende gegebene Entityklassen und Relationships:

FAKULTÄTEN (z.B. der TU Dresden wie Informatik, Medizin, Philosophie) (FNr, FName, FOrt)

INSTITUTE (z.B. Angewandte Informatik oder Künstliche Intelligenz) (INr, IName)

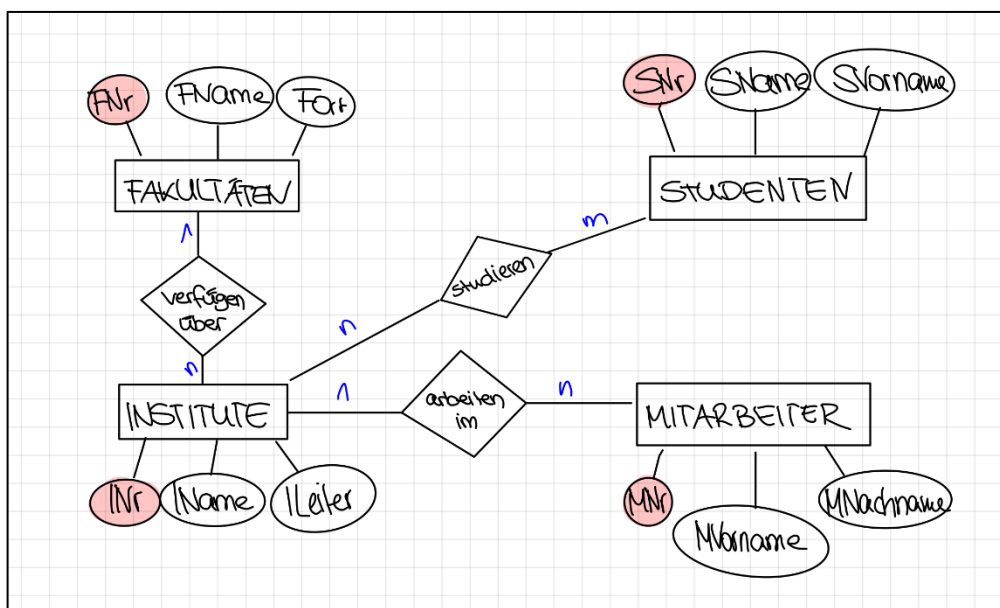
MITARBEITER (MNr, MName, MVorname)

STUDENTEN (SNr, SName, SVorname)

verfügt_über

arbeiten_im (Mitarbeiter im Institut)

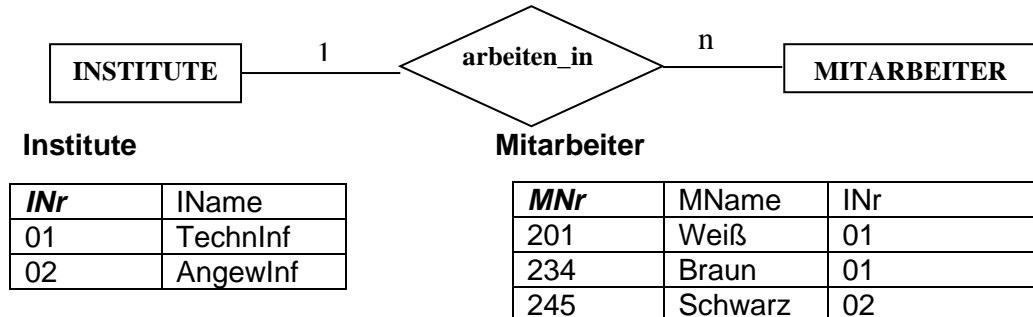
studieren (Studenten an unterschiedlichen Instituten)



Transformationsregeln

1. jede Entityklasse wird ohne Änderung in eine Relation überführt
2. zwei Entityklassen in 1:1-Beziehung werden zu einer Relation zusammengefasst
3. zwei Entityklassen in 1:n Beziehung werden in zwei Relationen überführt; der Primärschlüssel der 1-Seite wird als Fremdschlüssel der n-Seite eingefügt

Bsp.:



4. für zwei Entityklassen in n:m-Beziehung wird eine zusätzliche Relation modelliert, die die beiden Primärschlüssel als Fremdschlüssel enthält.

Bsp.:

