



Vorlesung  
Integrierte Modellierung komplexer Systeme  
Datenbanken und Datenmodellierung

Bachelor Studiengang Informatik  
WS 2017-2018

Hugo Colceag | MHP

© 2018 MHP Management- und IT-Beratung GmbH

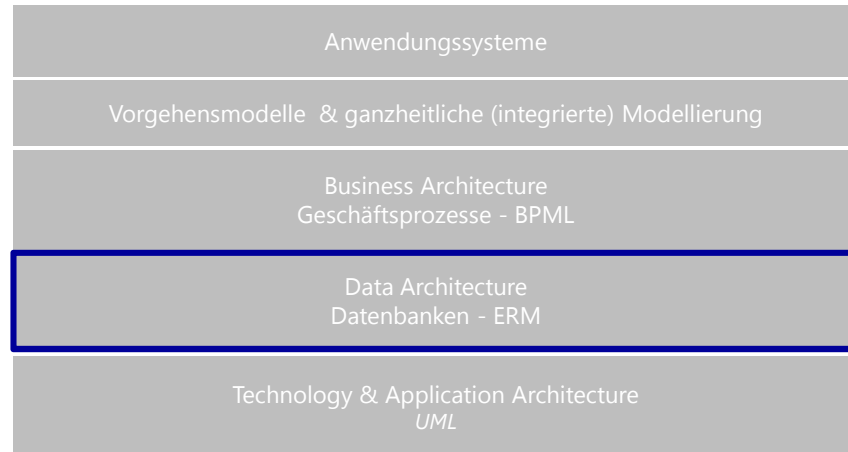


UNIVERSITATEA  
BABEŞ-BOLYAI

Einleitung - Integrierte Modellierung komplexer Systeme



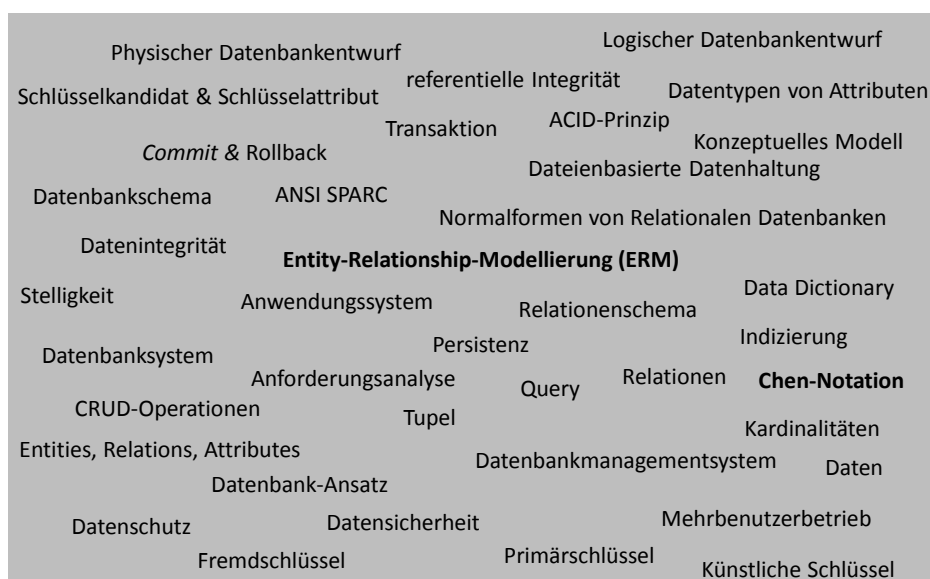
### 3 Vorlesungsinhalte und Aufbau



© 2018 MHP Management- und IT-Beratung GmbH

3


### Mindmap - Datenbanken und Datenmodellierung



© 2018 MHP Management- und IT-Beratung GmbH

4

Agenda


  
A PORSCHE COMPANY

- 1 Daten und Datenbanken
- 2 Datenbankmanagementsysteme
- 3 Das relationale Datenmodell
- 4 Datenbankentwurf

© 2017 MHP Management- und IT-Beratung GmbH

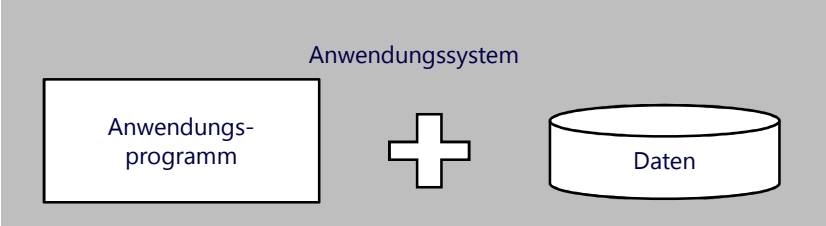
5

1 Daten und Datenbanken

  
A PORSCHE COMPANY

### 1.1 Daten

- Daten (nach Din 44300) Daten sind Zeichen oder kontinuierliche Funktionen, die aufgrund von bekannten oder unterstellten Abmachungen oder vorrangig zum Zwecke der Verarbeitung Informationen darstellen.
- Daten nach ISO/IEC 2382-1 „a reinterpretable representation of information in a formalized manner, suitable for communication, interpretation or processing“
- Anwendungssystem = Daten plus Anwendungsprogramm, das die Daten verwendet und bearbeitet



```
graph LR; subgraph Anwendungssystem; direction LR; A[Anwendungsprogramm] --- B((Daten)); end
```

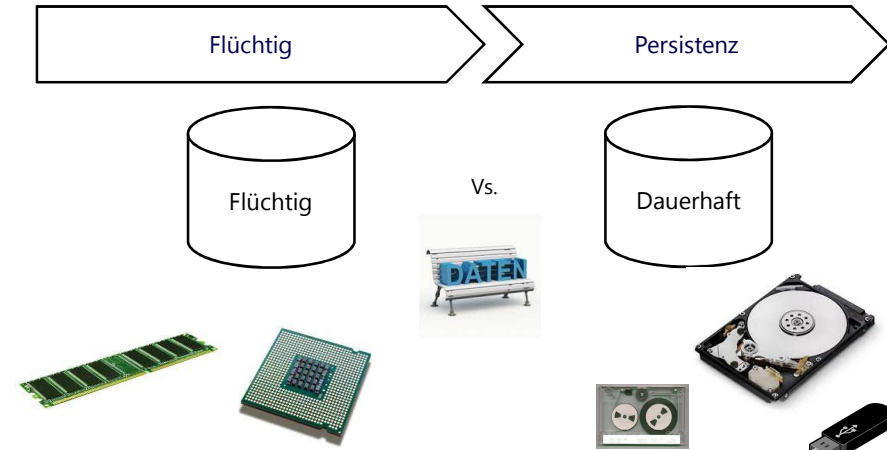
© 2018 MHP Management- und IT-Beratung GmbH

6

## 1 Daten und Datenbanken

## 1.2 Dauerhafte Datenspeicherung – Persistenz

- Ablageort in datenverarbeitende Anwendungsprogramme



© 2018 MHP Management- und IT-Beratung GmbH

7

## 1 Daten und Datenbanken

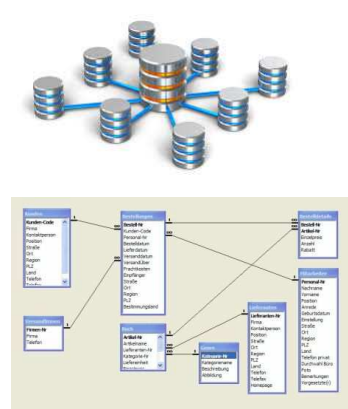
## 1.2.1 Datenhaltung

- „dateienbasierte Datenhaltung“



- „Datenbank-Ansatz“

Vs.



© 2018 MHP Management- und IT-Beratung GmbH

8

## 1 Daten und Datenbanken



## 1.2.1 Dateienbasierte Datenhaltung

- Beispiel 1.1 Dateienbasierte Datenhaltung für die Schulungsplanung

Die Hoske GmbH bietet häufig Schulungen zu ihren Produkten in verschiedenen Städten an. Herr Mayer ist dafür zuständig, die Räumlichkeiten und das Catering in den Pausen zu organisieren. Dazu hat er ein Excel-Tool entwickelt, in dem er Listen mit Anbietern von Veranstaltungsräumen und von Catering-Unternehmen führt. Auch die geplanten Schulungstermine sind hinterlegt. Herr Mayer nutzt das Tool, um die Anfragen und Zusagen der Anbieter für die einzelnen Schulungstermine zu verwalten. Frau Singers Aufgabe ist es, Dozenten für die Schulungen zu organisieren. Sie hat dafür auch ein Excel-Tool, das Informationen über Dozenten, ihre Kompetenzbereiche und Gagen enthält. Sie nutzt ihr Tool, um die Anfragen und Zusagen der Dozenten für die einzelnen Schulungstermine zu verwalten. Linda erfasst die Anmeldungen zu den Schulungen. Eine Schulung findet statt ab 5 Teilnehmern. Pro Termin können maximal 20 Personen teilnehmen.

## 1 Daten und Datenbanken



## 1.2.2 Der Datenbank-Ansatz zur Datenhaltung

- Datenbank ist eine Sammlung von strukturierten, inhaltlich zusammenhängenden Daten.
- Die Struktur nennt man auch „Datenmodell“.
- Eine Datenbank beschreibt einen wohldefinierten Weltausschnitt (Domain)
- Eine zufällige Datensammlung wird nicht als Datenbank bezeichnet.
- alle Daten, die zu einem bestimmten Anwendungsbereich gehören
  - Zentral
  - Für alle Benutzer und Anwendungen,
  - gemeinsame Datenspeicherung
  - Abruf aus der gemeinsamen Datenspeicherung
- Weltausschnitt vollständig abgebildet
  - alle Daten enthalten, die für den Anwendungsbereich relevant sind



## 1 Daten und Datenbanken



## 1.3 Übungen

- Beispiel 1.2 Die Miniwelt der Schulungsplanung

Die Schulungen der Hoske GmbH erfreuen sich reger Nachfrage und sind profitabel. Um dem wachsenden Aufwand gerecht zu werden und um Fehlplanungen zu vermeiden, lässt die Hoske die Planungstools verbessern. Alle Daten, die Schulungen betreffen, wie Räumlichkeiten, Termine, Dozenten, Catering-Anbieter, aber auch Teilnehmer, Schulungsunterlagen, Kugelschreiber, Schreibblöcke und Werbegeschenke und alle Anfragen und Zusagen sind jetzt **in einer gemeinsamen Datenbank** abgelegt.

## 1 Daten und Datenbanken



## 1.3 Übungen

- Aufgabe 1.1

Für die Aufgaben der Schulungsplanung, die in Beispiel 1.1 und Beispiel 1.2 beschrieben sind:

**Welche Vorteile bringt der Datenbank-Ansatz gegenüber der dateibasierten Datenhaltung mit sich?**

Geben Sie Beispiele von Vorgängen oder Situationen, die die Mitarbeiter der Hoske dank des datenbankbasierten Ansatzes besser bearbeiten können als mit ihren eigenentwickelten Tools.

**Hat der Datenbank-Ansatz auch Nachteile? Nennen Sie sie.**

## 1 Daten und Datenbanken



## 1.3 Übungen

- Beispiel 1.3 Datenhaltung im Customer Relationship Management

Dateienbasierte Datenhaltung in der Vertriebsabteilung des Unternehmens Hoske: jeder Vertriebler pflegt seine eigenen Notizen in eigenen Dateien, die er so aufbaut und organisiert, wie es ihm oder ihr am sinnvollsten scheint. Manche nutzen dazu Word-Dokumente mit speziellen Vorlagen, in denen sie die Informationen erfassen, und legen die Dokumente in ihrer eigenen, für sie sinnvollen Verzeichnisstruktur ab. Andere pflegen die Informationen strukturiert in selbst entwickelte Excel-Tabellen ein. Einige mobile Außendienstmitarbeiter verwenden eine Notizbuch-App auf ihrem Smartphone mit Datenspeicher in der Cloud.

Datenbank-Ansatz: es gibt eine gemeinsame Datenbank, in die jeder Vertriebler die Infos zu seinen Kundenkontakten einstellt. Jeder Vertriebsmitarbeiter hat darauf Zugriff und kann im Falle von Urlaub, Krankheit oder Kundenübernahmen alle Informationen abrufen, die sie oder er benötigt. Speziell entwickelte Software stellt dazu Bedienoberflächen zur Verfügung.

## 1 Daten und Datenbanken



## 1.3 Übungen

- Aufgabe 1.2

Im Vertrieb ist es wichtig, seine Kunden genau zu kennen und zu wissen, was sie wollen. Vertriebsmitarbeiter verfassen daher Notizen über ihre Kundenkontakte, sogenannte Besuchszettel, Messeberichte oder einfache Gesprächsnotizen.

Zu den wichtigen Informationen über einen Kundenkontakt gehören zum Beispiel der Name des Kunden, den Anlass des Kontakts (Messe, Neuauftrag, Reklamation etc.), das Medium (persönlich, telefonisch, Email etc.), den Initiator (Kunde oder Vertriebsmitarbeiter), den Ort und die Zeit, die Gesprächspartner und die besprochenen Themen und Übereinkünfte.

Die Vertriebsmitarbeiter des Unternehmens Hoske im Innendienst, insgesamt 5 Personen, lösen die Aufgabe, die Informationen zu Kundenkontakten zu verwalten, mithilfe einer Excel-Arbeitsmappe „Kundenkontakte“.

## 1 Daten und Datenbanken



## 1.3 Übungen

## ▪ Aufgabe 1.2

- a) Wie könnte die Excel-Mappe „Kundenkontakte“ konkret aussehen?
- b) Welche Daten benötigt und erzeugt die Anwendung?
- c) Bitte entwickeln Sie zu zweit oder zu dritt ein Konzept für eine solche Excel-Arbeitsmappe (direkt in Excel oder als Skizze).
- d) Beschreiben Sie in Worten detailnachvollziehbar detailliert die Standardprozesse, wie Mitarbeiter Daten in die Mappe einpflegen und wie sie Informationen abrufen.

## 1 Daten und Datenbanken



## 1.3 Übungen

## Aufgabe 1.2

- a) Ist die Lösung prinzipiell gut geeignet, die Standardprozesse zu unterstützen?
- b) Welche Vorteile und welche Schwächen weist die Lösung auf?
- c) Welche Verbesserungs- und Erweiterungsmöglichkeiten gibt es? Beschreiben Sie weitere mögliche Anwendungsfälle für das System, also Ziele, die Mitarbeiter mithilfe eines Systems wie „Kundenkontakte“ möglicherweise erreichen wollen.
- d) Schlagen Sie aufbauend auf c) einige zusätzliche Funktionen vor, die für die Vertriebsmitarbeiter nützlich sein könnten. Wie gut ist dieses Excel-basierte System geeignet, um diese zusätzlichen Funktionen umzusetzen?
- e) Handelt es sich bei dieser Lösung um eine **dateienbasierten** Datenhaltung oder um den **Datenbank-Ansatz**? Bitte diskutieren Sie.



## Agenda



- 1 Daten und Datenbanken
- 2 Datenbankmanagementsysteme
- 3 Das relationale Datenmodell
- 4 Datenbankentwurf

## 2 Datenbankmanagementsysteme



### 2.1 Grundfunktionen

- Ein Datenbankmanagementsystem (DBMS) ist ein allgemein einsetzbares Programmpaket, das die Definition, Erzeugung, Manipulation und gemeinsame Nutzung von Datenbanken durch mehrere Benutzer und Anwendungen unterstützt. (Elmasri & Navathe, Fundamentals of Database Systems, 2011, S. 5)

1. **Allgemein einsetzbar:** prinzipiell unabhängig vom Anwendungsbereich
2. **Definition:**
  - a) Datentypen, Strukturen und Einschränkungen für die Daten festzulegen
  - b) Definition und Beschreibung der Datenbank in einem **Datenkatalog** oder **Data Dictionary**
  - c) Informationen: Meta-Daten (das bedeutet „Daten über Daten“, also „Daten, die Daten beschreiben“)
3. **Erzeugung:** Daten gemäß der Datendefinition auf einem Speichermedium ablegen, das unter Kontrolle des DBMS steht
4. **Manipulation:** Operationen auf Daten ausführen → *Query*

## 2 Datenbankmanagementsysteme

## 2.1 Grundfunktionen

4. **Manipulation:** Operationen auf Daten ausführen → *Query*

- *Query:* „Anfragen an die Datenbank stellen“ / *querying a database*

- CRUD-Operationen = Create-Read-Update-Delete:

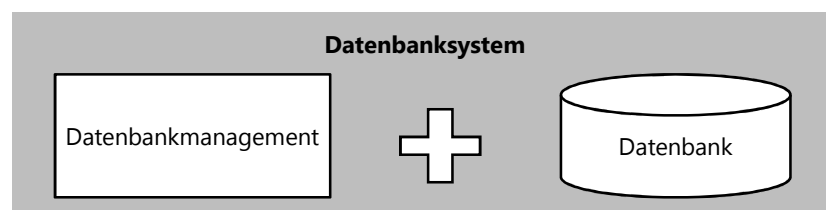
- **Create:** Daten eintragen
- **Read:** Daten abrufen
- **Update:** Daten ändern
- **Delete:** Daten löschen

5. **Gemeinsame Nutzung:** mehrere Benutzer und Softwareanwendungen können Datenbank gleichzeitig nutzen (DBMS koordiniert die Zugriffe)

## 2 Datenbankmanagementsysteme

## 2.2 Datenbanksystem

- Ein System bestehend aus einem Datenbankmanagement plus einer Datenbank nennt man ein Datenbanksystem.



## 2 Datenbankmanagementsysteme



### 2.2 Datenbanksystem

#### ▪ Beispiel 2.1 Excel als Datenbankmanagementsystem?

Eine Datenbank kann auch in Excel oder einer anderen Tabellenkalkulationssoftware verwaltet werden:

Die **Definition** der Datenbank besteht darin festzulegen, wie die Daten auf einem oder mehreren Arbeitsblättern in welchen Zellen in welchen Zahlenformaten abgelegt werden sollen

**Erzeugung** der Datenbank bedeutet, die Strukturen in der Excel-Arbeitsmappe anzulegen und die Arbeitsmappe dauerhaft zu speichern.

Die Benutzer können nun Daten in der Datenbank suchen und lesen, neue Daten eintragen und bestehende **Daten ändern** oder löschen.

**Gemeinsame Nutzung**, auch durch andere Softwareanwendungen, ist ebenfalls möglich, erfordert aber viel Programmierung oder organisatorischen Aufwand bei der Koordination der Zugriffe. Hier, aber auch an anderen Stellen, stößt eine Tabellenkalkulationssoftware als Datenbankmanagementsystem an Grenzen, wie im Folgenden noch deutlich werden wird.

## 2 Datenbankmanagementsysteme



### 2.3 Einsatz eines Datenbankmanagementsystem

- Wie wird ein DBMS eingesetzt?
- Wer benutzt und bedient es wie?
- Wie wirkt es mit anderen Anwendungen zusammen?

## 2 Datenbankmanagementsysteme



### 2.3.1 Zugriff auf die Daten nur über das DBMS

- Funktion des Datenbankmanagementsystems [DBMS]
  - Daten
    - dauerhaft speichern (indem es sie in Dateien auf geeignete Speichermedien schreibt)
    - manipulierbar machen (legt die Daten in speziellen Strukturen ab)
    - bereitstellen
  - Die Dateien mit den Daten schreibt und liest deshalb ausschließlich das Datenbankmanagementsystem

## 2 Datenbankmanagementsysteme



### 2.3.2 Verwaltung von mehreren Datenbanken

Ein Datenbankmanagementsystem kann bei Bedarf auch mehrere Datenbanken verwalten (vergleiche auch 2.3.3).

## 2 Datenbankmanagementsysteme

## 2.3.3 Datenbanksystem als Dienstprogramm für andere Anwendungen

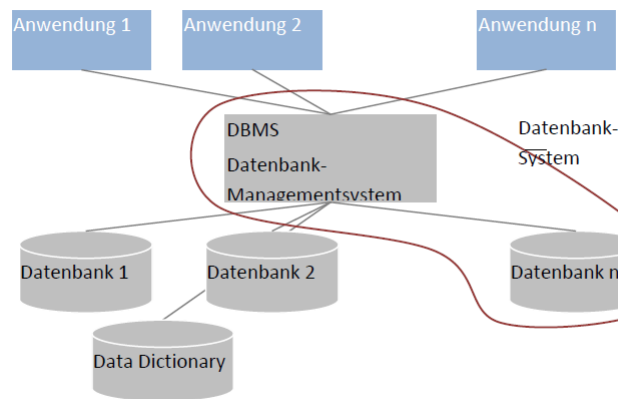


Abbildung 2-1 Aufbau und Einsatz eines Datenbanksystems

## 2 Datenbankmanagementsysteme

## 2.3.4 Trennung von Daten und Anwendungsprogrammen

- Anwendungsprogramme und Datenbank weitgehend voneinander entkoppelt
- Datenbank: speichert Informationen als **anwendungsunabhängige „Miniwelt“**
- Anwendungssoftware und Datenbank können getrennt voneinander entwickelt werden
- **Datenbank:**
  - Aufgabe: erforderliche Informationen bereitstellen
    - Struktur ergibt sich jedoch aus der Logik der abgebildeten Miniwelt
    - !!!! **NICHT** aus den Anforderungen der Anwendungssoftware
  - unabhängig von einer bestimmten Anwendungssoftware
  - kann von verschiedenen Anwendungsprogrammen genutzt werden
  - liefert auch ohne Anwendungsprogramm verständliche Informationen

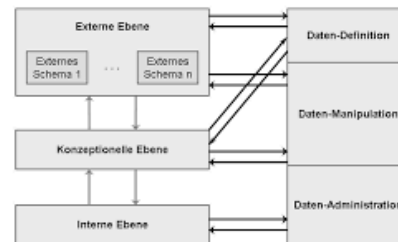
## 2 Datenbankmanagementsysteme

## 2.3.5 3-Ebenen-Architektur nach ANSI SPARC

Die Drei-Ebenen-Architektur wurde 1975 vom Standards Planning and Requirements Committee (**SPARC**) des American National Standards Institute (**ANSI**) entwickelt.

Die **Vorteile** sind:

- Logische Datenunabhängigkeit
- Physikalische Datenunabhängigkeit



Externe Ebene	Benutzersicht
Logische/Konzeptionelle Ebene	Datenmodellierung, Miniwelt
Interne Ebene	Physikalische Speicherung, Indizes

## 2 Datenbankmanagementsysteme

## 2.4 Weitere Funktionen von Datenbankmanagementsystemen

Da Datenbankmanagementsysteme mittlerweile seit einem halben Jahrhundert im Einsatz sind und immer weiterentwickelt wurden, handelt es sich dabei um **hochspezialisierte** und **mächtige Systeme**, ohne die viele Anwendungen gar nicht möglich wären. Man stelle sich zum Beispiel die **Produktkataloge** von Amazon vor, die Menge der darin enthaltenen Daten und die **Anzahl der Zugriffe** voraus der ganzen Welt.

1. Indizierung und Anfrageoptimierung
2. Mehrbenutzerbetrieb / Sperrkonzepte
3. Datenintegrität
4. Transaktionsverwaltung
5. Datensicherheit
6. Datenschutz

## 2 Datenbankmanagementsysteme

## 2.4.1 Effizienz [Indizierung und Anfrageoptimierung]

**Indizierung:**

- Verzeichnis, in dem das DBMS nachschlägt, an welcher **Position** im Datenspeicher ein Datensatz steht, von dem nur ein bestimmter Wert bekannt ist
- oft als Baum aufgebaut
- Vorteil: Schneller Zugriff auf den gesuchten Datensatz
- Nachteile: Speicherplatz, Zeit für Erstellung und Aktualisierung

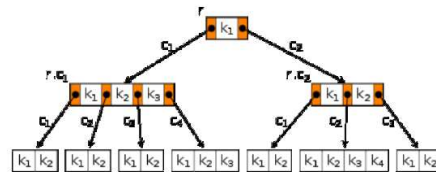


Abbildung 2-3 Beispiel für einen Index

**Automatische Anfrageoptimierung:**

DBMS analysiert Anfragen und formt sie so um, dass der Zugriff auf die Daten möglichst schnell von statten geht.

## 2 Datenbankmanagementsysteme

## 2.4.2 Mehrbenutzerbetrieb

- mehrere Benutzer oder Anwendungen greifen **gleichzeitig** auf die Daten zu
- **Probleme:**
  - gleichzeitige Änderungen von Daten
  - Gleichzeitige Änderung und Anzeige von Daten
  - Gleichzeitige Anzeige und Löschung von Daten
  - ... weitere **gleichzeitige Ausführung von CRUD Operationen**
- **Lösung:**
  - Unterschiedliche Sperrkonzepte

## 2 Datenbankmanagementsysteme

## 2.4.2 Mehrbenutzerbetrieb

- Beispiel 2.2 **Lost Update** durch **konkurrierenden Zugriff mehrerer Benutzer**

Beim Lost Update verändern zwei Benutzer fast gleichzeitig denselben Datenwert. Das Beispiel zeigt, wie dabei die Änderung eines der Benutzer verloren gehen kann.

Im Beispiel dient eine Datenbank dazu, Kontostände zu speichern. Auf dem Konto 511 liegen 100€. Der Benutzer möchte 10€ auf das Konto 511 einzahlen, der Benutzer B ebenfalls.

Zeit	A	B	kstand
	read(kstand)		100€
		read(kstand)	100€
	kstand ← kstand + 10		100€
		kstand ← kstand + 10	100€
	update(kstand)		110€
		update(kstand)	110€

## 2 Datenbankmanagementsysteme

## 2.4.3 Datenintegrität - Fehler und Inkonsistenzen von vornherein verhindern

- Beispiel 2.3 Integritätsbedingungen (*integrity constraint*) in einer Hochschuldatenbank

- I1: Jeder Student muss **eine Matrikelnummer** haben
- I2: Jede Matrikelnummer darf **nur einmal vergeben** werden
- I3: Jeder Student **muss in einem Studiengang** eingeschrieben sein.
- I4: Als Studiengang eines Studenten darf nur ein in der Datenbank **existierender Studiengang** eingegeben werden.
- I5: Der Datensatz eines Studiengangs **darf nicht** aus der Datenbank **gelöscht** werden, solange noch **Studierende im Studiengang eingeschrieben** sind.

Datenbankmanagementsysteme verfügen über Mechanismen und gewährleisten so in gewissen Grenzen Datenintegrität.

**Ziel:** Vorkehrungen treffen, um Fehler und Inkonsistenzen von **vornherein** zu verhindern!



## 2 Datenbankmanagementsysteme



### 2.4.4 Transaktionsverwaltung

#### Transaktion:

Eine Transaktion ist eine **Folge von DB-Operationen**, die als **atomare (unteilbare) Einheit** betrachtet wird.

- zentrales Konzept zur Steuerung des Mehrbenutzerbetriebs
- Erhalt der Datenintegrität
- Transaktion besteht aus mehreren elementaren DB-Operationen
- Elementare DB-Operationen sind die bekannten CRUD-Operationen
  
- **DBMS** verhält sich so, als wäre die **Transaktion eine unteilbare Einheit**
- **Bei Fehler in einzelnen sequenzielle ausgeführten Operationen** stellt das DBMS den Ausgangszustand wieder her (**Rollback**)
- Persistierung durch DBMS erfolgt nur nach kompletter fehlerfreier Ausführung der Transaktion (**Commit**)

## 2 Datenbankmanagementsysteme



### 2.4.4 Transaktionsverwaltung – ACID Prinzip

#### Transaktion:

Eine Transaktion ist eine **Folge von DB-Operationen**, die als **atomare (unteilbare) Einheit** betrachtet wird.

- **ACID-Prinzip:**
  - **Atomarität:** vollständig oder gar nicht
  - **Consistenz:** Integritätsbedingungen eingehalten
  - **Isolation:** jede Transaktion ist unabhängig von anderen Transaktionen
  - **Dauerhaftigkeit:** Eine abgeschlossene Transaktion überdauert nachfolgende Fehler

## 2 Datenbankmanagementsysteme



### 2.4.5 Datensicherheit

- DBMS bieten Mechanismen, um Daten vor Verlust zu sichern
- Beispiele:
  - Backups: Erstellen von Datensicherungskopien
  - Recovery: Wiederherstellen nach Störungen
  - Inkrementelle Datensicherung:
    - jede Änderung auf den Daten wird protokolliert
    - Änderungen auf der jeweils letzten Sicherungskopie wiederherstellen
  - Replikation:
    - ständig 2 oder mehrere Datenbankmanagementsysteme auf **separater Hardware parallel**
    - Replikation der Operationen des ersten, produktiven Systems auf zweites
    - Ersatzsystem
    - Szenarios bezeichnet man auch als **Replikations-Cluster**, das Prinzip heißt **Failover**

## 2 Datenbankmanagementsysteme



### 2.4.6 Datenschutz

- Datenschutz ist der Schutz der Daten vor unberechtigten Zugriffen.
- Datenbankmanagementsysteme unterstützen durch Mechanismen
- Definition von Benutzer und Benutzergruppen
- Authentifizierung
  - Anmeldung am Datenbankmanagementsystem
- Autorisierung
  - individuelle Berechtigungen
  - Leserechte oder Schreibrechte auf bestimmten Datenbereichen
  - Rechte Datenbanken oder Strukturen in bestimmten Datenbanken anzulegen oder zu löschen
  - ...

## 2 Datenbankmanagementsysteme



## 2.5 Übungen

## ▪ Aufgabe 2.1

Die Vertriebsmitarbeiter des Unternehmen Hoske sammeln Informationen über Kundenkontakte in einem Tool, das auf Basis des Tabellenkalkulationsprogramms MS Excel realisiert ist. Dieses Tool erfüllt mehrere Funktion:

Die Mitarbeiter können Informationen zu Kundenkontakten einpflegen und abrufen. Das Tool speichert die Daten dauerhaft. Im Laufe der Zeit sind einige Komfort-Funktionen dazugekommen wie zum Beispiel Auswertungen und Unterstützung bei der Planung von Kundenkontakten.

Welche der in Abschnitt 2.4 genannten Funktionen eines Datenbankmanagementsystems kann MS Excel in diesem Anwendungsszenario mit wenig Aufwand abdecken?

- a) Welche Funktionen, die ein DBMS bietet, wären in diesem Anwendungsszenario nützlich, aber mit Excel nur mit besonderem Aufwand zu realisieren?
- b) Welche Vorteile bietet Excel in diesem Anwendungsszenario gegenüber einem DBMS? (5 Min)

## 2 Datenbankmanagementsysteme



## 2.5 Übungen

## ▪ Aufgabe 2.2

Ein Unternehmen hat Tausende Kunden weltweit. Täglich kommen viele Anfragen von Kunden per Email und telefonisch und werden im „Kundenkontaktcenter“ bearbeitet. Dort arbeiten bis zu 15 Mitarbeiter gleichzeitig. Auch diese Mitarbeiter erfassen Informationen über die Kundenkontakte in einem CRM-System.

Erläutern Sie, welche Funktionen eines DBMS erforderlich sind, um die Anforderungen dieses Kundenkontaktcenters an das CRM-System abzudecken.

## Agenda

- 1 Daten und Datenbanken
- 2 Datenbankmanagementsysteme
- 3 Das relationale Datenmodell
- 4 Datenbankentwurf

## 3 Das relationale Datenmodell

## 3.1 Relationen

- Das abstrakte Datenmodell ist die Art, wie Daten betrachtet werden: als Tabellen, als Objekte, also Schlüssel-Wert-Paare und andere mehr.
- derzeit verbreitetste Datenmodellen ist das relationale Datenmodell
- Es wurde in den 1970er Jahren entwickelt (Codd, 1970), (Date C. , 1975).
- Das relationale Datenmodell repräsentiert Daten als Relationen. Relationen lassen sich darstellen als 2-dimensionale Tabellen

Eine Relation ist eine Menge von Tupeln  $(w_1, w_2, \dots, w_n) \in D_1 \times D_2 \times \dots \times D_n$   
 Die  $w_i$  heißen Werte, die  $D_i$  sind Wertebereiche von Attribute.

- Beispiel 3.1 Relation

Student

Matrikelnr	Name	Vorname	Geburtsdatum	Email
1315	Müller	Andi	23.01.1989	mueller8113@yahoo.com
1319	Meyer	Uli	08.08.1988	meyer2221@gmail.de
1321	Braun	Ute	20.05.1990	schlumpf@gmx.de

## 3 Das relationale Datenmodell

## 3.1 Relationen

- Übertragen auf die Darstellung als Tabelle:
  - Die **Attribute** entsprechen den **Spalten**.
  - Die Anzahl der Spalten/Attribute ist fest.
  - In jeder **Zeile** steht ein **Tupel**. Eine Zeile nennt man auch einen **Datensatz**.
  - Die Anzahl der Zeilen ist flexibel.
  - Jedes Tupel/jeder Datensatz kommt **nur einmal in der Relation** vor.
- Beispiel 3.1 Relation

Student

Matrikelnr	Name	Vorname	Geburtsdatum	Email
1315	Müller	Andi	23.01.1989	mueller8113@yahoo.com
1319	Meyer	Uli	08.08.1988	meyer2221@gmail.de
1321	Braun	Ute	20.05.1990	schlumpf@gmx.de

## 3 Das relationale Datenmodell

## 3.1 Relationen

- **Relationenschema**: beschreibt den Aufbau einer Relation
- **Stelligkeit** der Relation: die Anzahl ihrer Attribute
- **Datenbankschema**:
  - besteht aus der Menge der Relationenschemata einer Datenbank
  - beschreibt den Aufbau der Datenbank
- Beispiel 3.2 Relation Kunde

Kunde

Kdnr	Suchbegriff	Name	Branche
115	Mayer-Kempton	Mayer GmbH	MV
116	Mueller-Sonthofen	Franz Mueller und Sohn	MB
117	Mueller-Kempton	MPK Mueller Präzision Kempton	MB

- Relationenschema: **Kunde(Kdnr, Suchbegriff, Name, Branche)**
- Stelligkeit der Relation: **4**

## 3 Das relationale Datenmodell

## 3.2 Schlüssel

- Ein Schlüssel ist eine **Teilmenge** der Attribute, die jedes **Tupel eindeutig** identifizieren
  - Alle Attribute der Relation bilden zusammen also einen Schlüssel.
- Ein **Schlüsselkandidat** ist ein Schlüssel mit **minimaler** Anzahl Attribute
  - Eine Relation kann mehrere Schlüsselkandidaten haben.
- Ein **Schlüsselattribut** ist ein Attribut, das Teil mindestens eines Schlüsselkandidaten ist.
  - Alle anderen Attribute sind Nicht-Schlüsselattribute.

Student		Schlüsselkandidat 2			
Schlüsselattribute	Matrikelnr	Name	Vorname	Geburtsdatum	Email
Tupel	1315	Müller	Andi	23.01.1989	mueller8113@yahoo.com
	1319	Meyer	Uli	08.08.1988	meyer2221@gmail.de
	1321	Braun	Ute	20.05.1990	schlumpf@gmx.de
		Schlüsselkandidat 1			

## 3 Das relationale Datenmodell

## 3.2.1 Primärschlüssel

- Ein **Primärschlüssel** (englisch **primary key**) ist ein beliebig ausgewählter Schlüsselkandidat, der zur **eindeutigen Identifizierung** jeder Zeile benutzt wird.
- Ein Primärschlüssel, der aus mehreren Attributen besteht, ist ein **zusammengesetzter Primärschlüssel**
- Primärschlüssel kennzeichnet man in Relationenschemata und Tabellen durch **Unterstreichen der Attributnamen/Spaltenüberschriften**.

Student					
<u>Matrikelnr</u>	Name	Vorname	Geburtsdatum	Email	
1315	Müller	Andi	23.01.1989	mueller8113@yahoo.com	
1319	Meyer	Uli	08.08.1988	meyer2221@gmail.de	
1321	Braun	Ute	20.05.1990	schlumpf@gmx.de	

## 3 Das relationale Datenmodell

## 3.2.2 Fremdschlüssel und referentielle Integrität

- Fremdschlüssel bilden Beziehungen zwischen Daten in **verschiedenen Relationen** ab
- Eine Relation wird bezeichnet als **Elterntabelle**, oder unabhängige Relation, **Master-Tabelle**, referenzierte Tabelle.
- Die andere Relation heißt **Kindtabelle**, abhängige Relation, **Detail-Tabelle** beziehungsweise **referenzierende Tabelle**
- Ein **Fremdschlüssel** ist ein Attribut in einer Kindtabelle, das sich auf einen Primärschlüssel in der Elterntabelle bezieht
- Ein **Fremdschlüssel** (englisch *foreign key*) ist ein Attribut in einer Tabelle, das einem Primärschlüssel in einer anderen Tabelle entspricht oder eine Menge von Attributen) in einer Tabelle, die einem zusammengesetzten Primärschlüssel in einer anderen Tabelle entspricht.
- Die **referentielle Integrität** (Integrität der Bezüge) ist eine wichtige **Integritätsbedingung**. Referentielle Integrität ist gegeben, wenn alle Werte, die im Fremdschlüssel in der Kindtabelle vorkommen, auch im Primärschlüssel der Elterntabelle enthalten.

## 3 Das relationale Datenmodell

## 3.2.2 Fremdschlüssel und referentielle Integrität

- Beispiel 3.3 Fremdschlüssel

Student				Studiengang		
Matrikelnr	Name	Vorname	Studiengang	Studiengang	Abschluss	Leiter
1315	Müller	Andi	WI	WI	Bachelor	Prof. Ganz
1319	Meyer	Uli	WI	MB	Master	Prof. Schwer
1321	Braun	Ute	MB			

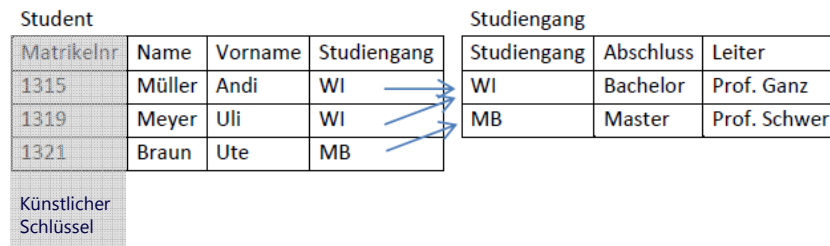
Fremd-schlüssel
Primär-schlüssel

Die **referentielle Integrität ist erfüllt**, denn für alle Studiengänge, in die Studenten laut Kindtabelle eingeschrieben sind, existiert ein entsprechender Eintrag in der Elterntabelle.

## 3 Das relationale Datenmodell

## 3.2.3 Künstliche Schlüssel

- Attribute ein, die an sich keine eigene Information tragen, um sie als Schlüssel zu verwenden nennt man **künstliche Schlüssel**.



## 3 Das relationale Datenmodell

## 3.2.3 Künstliche Schlüssel

- Beispiel 3.4 Künstliche Schlüssel

Beispiele für künstliche Schlüssel sind: Matrikelnummer, Personalnummer, Kundennummer, Auftragsnummer, ISBN, Inventarnummer, Seriennummer.

Die Kombination (Name, Vorname, Geburtstag, Geburtsort) bildet einen natürlichen Schlüssel für Personen, ist aber unübersichtlich und umständlich einzusetzen.

**Wann und wozu verwendet man künstliche Schlüssel?**

- Wenn es keine natürlichen Schlüssel gibt oder wenn sich der natürliche Schlüssel aus sehr vielen Attributen zusammensetzt.
- Weil man den Schlüssel als Verweis in andere Relationen (also als Fremdschlüssel) benutzen möchte.



## 3 Das relationale Datenmodell



## 3.3 Normalformen von Relationalen Datenbanken

- Informationen über einen Anwendungsbereich können unterschiedlich in Relationen abgebildet werden
- Normalformen sind „standardisierte“ Arten, die sich als besonders geeignet erwiesen haben
  - Erste Normalform (1NF)
  - Zweite Normalform (2NF)
  - Dritte Normalform (3NF)

## 3 Das relationale Datenmodell



## 3.4.2 Beispiel zum Normalisierungsprozess

- Beispiel 3.7 Normalisierung einer Datenbank mit Prüfungsanmeldungen

In diesem Beispiel wird eine Datenbank bis zur dritten Normalform normalisiert. Anfangs besteht diese Datenbank aus nur einer Tabelle. Diese dient dazu, Anmeldungen von Studierenden zu Prüfungen im laufenden Semester zu verwalten.

Prüfungsanmeldung

Name	Vorname	Matrikelnr	Studiengang	Abschluss	Leiter	Prüfung
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	PM, Physik, BWL
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz	BWL, PM
Braun	Ute	1321	MB	Master	Prof. Schwer	PM, Mathe

## 3 Das relationale Datenmodell

## 3.4.2.1 Ausgangsdatenbank

Prüfungsanmeldung

Name	Vorname	Matrikelnr	Studiengang	Abschluss	Leiter	Prüfung
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	PM, Physik, BWL
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz	BWL, PM
Braun	Ute	1321	MB	Master	Prof. Schwer	PM, Mathe

## 3 Das relationale Datenmodell

## 3.4.2.2 Datenbank in 1NF

Prüfungsanmeldung

Name	Vorname	Matrikelnr	Studiengang	Abschluss	Leiter	Prüfung
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	PM, Physik, BWL
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz	BWL, PM
Braun	Ute	1321	MB	Master	Prof. Schwer	PM, Mathe



Prüfungsanmeldung

Name	Vorname	Matrikelnr	Studiengang	Abschluss	Leiter	Prüfung
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	PM
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	Physik
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	BWL
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz	BWL
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz	PM
Braun	Ute	1321	MB	Master	Prof. Schwer	PM
Braun	Ute	1321	MB	Master	Prof. Schwer	Mathe

## 3 Das relationale Datenmodell

## 3.4.2.3 Datenbank in 2NF

Prüfungsanmeldung

Matrikelnr	Prüfung
1315	PM
1315	Physik
1315	BWL
1319	BWL
1319	PM
1321	PM
1321	Mathe

Prüfungsanmeldung

Name	Vorname	Matrikelnr	Studiengang	Abschluss	Leiter	Prüfung
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	PM
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	Physik
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	BWL
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz	BWL
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz	PM
Braun	Ute	1321	MB	Master	Prof. Schwer	PM
Braun	Ute	1321	MB	Master	Prof. Schwer	Mathe

Student

Name	Vorname	Matrikelnr	Studiengang	Abschluss	Leiter
Müller	Andi	1315	WI	Bachelor	Prof. Ganz
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz
Braun	Ute	1321	MB	Master	Prof. Schwer

## 3 Das relationale Datenmodell

## 3.4.2.4 Datenbank in 3NF

Prüfungsanmeldung

Matrikelnr	Prüfung
1315	PM
1315	Physik
1315	BWL
1319	BWL
1319	PM
1321	PM
1321	Mathe

Student

Name	Vorname	Matrikelnr	Studiengang	Abschluss	Leiter
Müller	Andi	1315	WI	Bachelor	Prof. Ganz
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz
Braun	Ute	1321	MB	Master	Prof. Schwer

Prüfungsanmeldung

Matrikelnr	Prüfung
1315	PM
1315	Physik
1315	BWL
1319	BWL
1319	PM
1321	PM
1321	Mathe

Studiengang

Studiengang	Abschluss	Leiter
WI	Bachelor	Prof. Ganz
MB	Master	Prof. Schwer

Student

Name	Vorname	Matrikelnr	Studiengang
Müller	Andi	1315	WI
Meyer	Uli	1319	WI
Braun	Ute	1321	MB

## 3 Das relationale Datenmodell

## 3.5 Übungen

Eine relationale Datenbank enthält Daten zu Schulungen. Abbildung 3-1 zeigt einen Ausschnitt.

▪ Aufgabe 3.1 Relationale Datenbanken – Begriffe

- Welche Relationenschemata gehören zu den gezeigten Relationen?
- Was ist die Stelligkeit der Relation Schulung?
- Was ist die Stelligkeit der Relation Dozent?
- Markieren Sie in den Relationenschemata geeignete Primärschlüssel für die Relationen durch Unterstreichen.
- Markieren Sie Fremdschlüssel in den Relationenschemata mit einem Punkt •.
- Sind alle Relationen in 3NF? Wenn nein, welche Relation verletzt welche Normalform?

Schulung				Expertise			
Nr	Titel	Dauer		SchulungNr	DozentNr	Bewertung	Honorar
CA1	CAM basic	1 Tag		CA1	1	4	100
C22	CAM expert	3 Tage		CA1	2	4	100
MT1	Maintenance Expert	2 Tage		C22	1	4	150
				C22	2	4	150
				MT1	3	3	100

Dozent				Location			
Nr	Name	Titel	Vorname	Nr	Name	Ort	Adresse
1	Schneider	Dipl.-Ing.	Franz	4	Regis Business Center	München	Messe 4
2	Sulian	B. Eng.	Sandra	5	Seminaris Hotel	Stuttgart	Bärenstr 12
3	Schlieder	M. Eng.	Steffan				

Termin					
Nr	Terminnr	Termin	Ort	Dozent	Location
CA1	41	23.11.2015	München	2	4
CA1	42	23.11.2015	Stuttgart	1	5
C22	30	24.11.2015	Stuttgart	3	5

Abbildung 3-1 Datenbank der Miniwelt Schulungen (Ausschnitt)

## 3 Das relationale Datenmodell

## 3.5 Übungen

Schulung			Expertise			
Nr	Titel	Dauer	SchulungNr	DozentNr	Bewertung	Honorar
CA1	CAM basic	1 Tag	CA1	1	4	100
C22	CAM expert	3 Tage	CA1	2	4	100
MT1	Maintenance Expert	2 Tage	C22	1	4	150
			C22	2	4	150
			MT1	3	3	100

Dozent			Location		
Nr	Name	Titel	Nr	Name	Adresse
1	Schneider	Dipl.-Ing.	4	Regis Business Center	München Messe 4
2	Sulian	B. Eng.	5	Seminaris Hotel	Stuttgart Bärenstr 12
3	Schlieder	M. Eng.			

Termin					
Nr	Terminnr	Termin	Ort	Dozent	Location
CA1	41	23.11.2015	München	2	4
CA1	42	23.11.2015	Stuttgart	1	5
C22	30	24.11.2015	Stuttgart	3	5

Abbildung 3-1 Datenbank der Miniwelt Schulungen (Ausschnitt)

## 3 Das relationale Datenmodell

## 3.5 Übungen

- Aufgabe 3.2 Autohaus-Daten normalisieren

Ein kleines Autohaus, das seine Daten bisher in Excel erfasst und bearbeitet hat, wird langsam groß und möchte seine Daten jetzt in einer richtigen Datenbank verwalten. Abbildung 3-2 zeigt einen Ausschnitt aus der vorhandenen Excel-Datei.

- Bringen Sie diesen in 1NF.
- Bringen Sie diesen in 2NF.
- Bringen Sie diesen in 3NF.

Name	Adresse	Marke	Typ	Seriennr	Farbe	Verkäufer	VDatum	Eintrittsdatum
Meier	Platanenweg 7	VW	Golf	123456	blau-metallic	Schmid	23.04.2014	01.06.2006
		Opel	Astra	345678	silber	Plüss	07.08.2015	15.10.2007
Müller	Altstadt 12	VW	Golf	55567	silber	Frey	12.04.2015	01.06.2006
Steffen	Gartenstr. 2	VW	Bora	3232323	weiss	Schmid	15.07.2015	01.06.2006
Steffen	Gartenstr. 2	Audi	A6	55454545	schwarz	Frey	13.11.2014	01.06.2006

Abbildung 3-2 Tabelle mit Autohaus-Daten vor der Normalisierung

## 3 Das relationale Datenmodell

## 3.5 Übungen

- Aufgabe 3.3 Lieferungsdaten normalisieren

Im Lager der Hoske müssen die eingetroffenen Lieferungen manuell in einem Excel-Arbeitsblatt erfasst werden. Bringen Sie die Daten in 3NF und markieren Sie die Primärschlüssel.

Legende: AgNr: Artikelgruppennummer LiefNr: Nummer der Lieferung

Lieferungen

LiefNr	Lieferant	Land	22.2.2010	Artikel	Bezeichnung	AgNr	Artikelgruppe	Menge	Einheit
15500	Simmler	CH	22.2.2010	123	SchraubeM4	3	Schrauben	1000	Stk
15500	Simmler	CH	15.3.2010	678	SchraubeM8	3	Schrauben	1300	Stk
15300	Simmler	CH	15.3.2010	123	SchraubeM4	3	Schrauben	1000	Stk
15300	Simmler	CH	22.2.2010	678	SchraubeM8	3	Schrauben	1300	Stk
15060	Schauer	D	22.2.2010	123	SchraubeM4	3	Schrauben	4000	Stk
15060	Schauer	D	23.2.2010	690	SchraubeM3	3	Schrauben	2000	Stk
15570	Leclercq	F	23.2.2010	232	Schlauch12	51	Schläuche	10	m
15600	Miller	D	23.2.2010	500	Schlauch14	51	Schläuche	10	m
15600	Miller	D	22.2.2010	232	Schlauch12	51	Schläuche	5	m

Abbildung 3-3 Lieferungsdaten vor der Normalisierung

## 3 Das relationale Datenmodell

## 3.5 Übungen

- Aufgabe 3.4 Daten zum Lagerbestand normalisieren

Auch die Lagerbestände der Hoske werden manuell in einem Excel-Arbeitsblatt erfasst. Bringen Sie die Daten in 3NF und markieren Sie die Primärschlüssel.

Legende: AgNr: Artikelgruppennummer PM: Produktmanager, zuständig für Artikelgruppen

Lagerbestand

Artikel	Bezeichnung	Abmesung	Artikelgruppe	PM	Einheit	Bestand	Platz	Wert	Zugang
212	Schraube Me	4x12	Schrauben	Simmler	Stk	150	A-17	1,30	02.10.14
214	Schraube Me	4x14	Schrauben	Simmler	Stk	100	A-18	1,00	03.10.14
214	Schraube Me	4x14	Schrauben	Simmler	Stk	50	A-17	1,00	04.10.14
212	Schraube Me	4x12	Schrauben	Simmler	Stk	200	B-01	1,30	05.10.14
216	Schraube Me	4x16	Schrauben	Simmler	Stk	80	A-21	1,50	06.10.14
R15	Rohr Kupfer	15	Rohre	Sepp	m	10	R-11	1,00	07.10.14
R18	Rohr Kupfer	18	Rohre	Sepp	m	20	R-12	2,00	08.10.14

## Agenda

- 1 Daten und Datenbanken
- 2 Datenbankmanagementsysteme
- 3 Das relationale Datenmodell
- 4 Datenbankentwurf

## 4 Datenbankentwurf

## 4.1 Phasen des Datenbankentwurfs

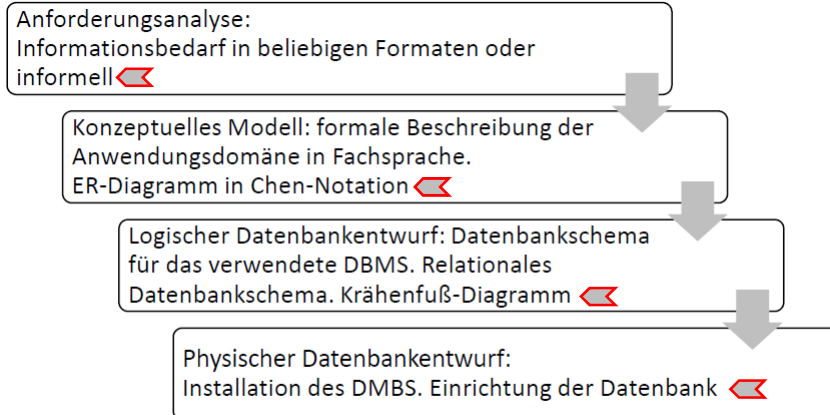
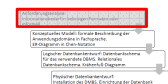


Abbildung 4-1 Überblick über die Phasen des Datenbankentwurfs

## 4 Datenbankentwurf

## 4.1.1 Anforderungsanalyse



- Datenbanksystem die Rolle eines Daten-Dienstleisters für ein oder mehrere Anwendungsprogramme
- Anwendungsprogramme führen Operationen auf diesen Daten aus und realisieren die eigentliche Funktionalität des Gesamtsystem
  - Anforderungen an Anwendungsprogramme bestimmen welche Informationen die Datenbank speichern und bereitstellen muss
- Die Anforderungen an die Anwendung können in verschiedenen, möglicherweise **informellen** Formaten vorliegen

## 4 Datenbankentwurf



## 4.1.2 Konzeptuelles Modell (Miniwelt, Anwendungsdomäne)

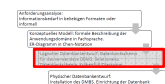


- beschreibt die **Anwendungsdomäne** oder **Miniwelt der Datenbank** auf der **Fachebene**
- Nutzung der **Fachsprache** der zukünftigen Fachanwender und Benutzer, jedoch als **formalisierte** Darstellung
- Ziele:
  - für Fachanwender verständlich
  - Diskussionsgrundlage
  - von technischen Details abstrahiert
  - formalisiert, so dass es eindeutig interpretierbar ist
- Beispielnotation:
  - **Entity-Relationship-Diagramme** in der Notation von **Chen**

## 4 Datenbankentwurf



## 4.1.3 Logischer Datenbankentwurf



- konzeptuelles Modell wird übersetzt in das Datenmodell eines **konkreten DMBS**
  - Beispiel: Übersetzung in ein relationales Modell
- **Eigenschaften:**
  - Detailliert
  - Nahe an der technischen Umsetzung
  - Enthält Schlüssel-Fremdschlüsselbeziehungen, Integritätsbedingungen, Datentypen (physische Umsetzung der Datenbank)



## 4 Datenbankentwurf

## 4.1.4 Physischer Datenbankentwurf



Der physische Entwurf hängt auch ab vom eingesetzten Datenbankmanagementsystem, da die Produkte verschiedener Hersteller hierfür unterschiedliche Optionen bieten.

- **Eigenschaften:**
  - legt technische Details fest
  - Definiert Speicherstrukturen
  - Definiert und richtet Indizes (wie in Abschnitt 2.4.1) ein
  - Auswahl von optimalen Suchalgorithmen
- Für komplexe Anwendungen mit großen Datenmengen braucht es Experten, um die Datenbank auf der Ebene des physischen Entwurfs optimal einzurichten.
- Für kleinere Anwendungen genügen Standardeinstellungen der üblichen DMBS völlig, um ein ausreichend effizientes Datenbanksystem einzurichten.

## 4 Datenbankentwurf

## 4.2.1 Entity-Relationship-Modellierung (ERM)

**Ein Entity-Relationship-Modell beschreibt, wie ein Ausschnitt aus der realen Welt in einer Datenbank abgebildet werden kann.**

- Methode zur Modellierung von Daten
- Entwurf von relationalen Datenbanken
- Analyse und Strukturierung der Informationen aus dem Anwendungsbereich
- Fachwissen der zukünftigen Benutzer und Anwender der Datenbank wichtig
- **ER Diagramme**
  - Anschaulich, leicht erweiterbar, abstraktes Niveau
  - für Nicht-Datenbankspezialisten verständlich
  - Objekte und Strukturen der abzubildenden Anwendungsdomäne
  - Arbeitsgrundlage in Team-Diskussionen
- **Erweiterungen und alternative Notationen (Schreibweisen)**
  1. **Chen Notation:** Methode für den ersten manuellen Entwurf eines konzeptuellen Datenbankmodells
  2. Notation nach Barker / **Krähenfuss Notation:** detaillierte Modelle als mit der Chen-Notation

## 4 Datenbankentwurf



## 4.2.2 Elemente der ER-Modellierung

- **Entities:** Gegenstände, Dinge, auch Personen in der betrachteten Miniwelt
- **Relationships:** Beziehungen, die zwischen Paaren von Entities bestehen
- **Attribute:** relevante Eigenschaften der Entities oder der Relationships
- Deutsche Begriffe:
  - Entity → Entität
  - Relationship → Beziehung

## 4 Datenbankentwurf



## 4.2.2.1 Entities, Attribute und Entity-Typen

- **Entities:**
  - Gegenstände, Dinge, auch Personen in der betrachteten Miniwelt
  - Objekt der realen Welt
  - eigene Identität
  - eindeutig identifizierbar
  - weisen Attribute auf, die einen bestimmten Wert annehmen können
- **Entity-Typ:**
  - Gleichartige Entities
  - dieselben Attribute
- **Identifizierende Attribute**
  - Identifizieren Entitäten eindeutig
  - Entsprechen den Schlüsselattributen in relationalen Datenbanken
  - Beispiel: ISBN-Nummer eines Buches, Inventarnummer eines Bürodruckers

## 4 Datenbankentwurf



## 4.2.2.1 Entities, Attribute und Entity-Typen

- Beispiel 4.1 Entity und Entity-Typ

Ein einzelnes von 10.000 Blatt Druckerpapier ist **kein Entity**. Ein Bürodrucker, der im Netzwerk **identifizierbar** ist oder eine **Inventarnummer** besitzt, ist ein **Entity**.

Alle Bürodrucker sind vom selben Entity-Typ Drucker. Entities des **Entity-Typs Drucker** weisen die **Attribute** Inventarnummer, Art, Anschaffungsdatum und Netzwerkadresse auf.

Ein Entity dieses Entity-Typs hat dann zum Beispiel im Attribut Inventarnummer den Wert „123321“ und im Attribut Art den Wert „Farblaser“.

Im allgemeinen Sprachgebrauch wird oft nicht sauber zwischen Entity und Entity-Typ getrennt, so dass mit „Entity“ manchmal genaugenommen „Entity-Typ“ gemeint ist.

## 4 Datenbankentwurf



## 4.2.2.1 Entities, Attribute und Entity-Typen

Ein „**schwaches**“ Entity ist ein Entity, das nur in Abhängigkeit von einem anderen Entity existieren kann.

- Beispiel 4.2 Starke und schwache Entities in der Miniwelt „Schulungen“

Eine **Schulung** ist ein **starkes Entity**. Sie besitzt Attribute wie Schulungsnummer (als identifizierendes Attribut), Titel, Beschreibung, Umfang, usw.  
Zum Beispiel die Schulung „CAM basic“: Diese Schulung findet mehrmals jährlich statt.

Jeder **Schulungstermin** wird separat geplant. Er hat als Attribute TerminNr, Datum, usw.  
Ein Schulungstermin ist ein **schwaches Entity**, weil er nur in Abhängigkeit zum Entity Schulung „CAM basic“ existieren kann: **ohne Schulung kann es auch keinen Schulungstermin geben.**

#### 4.2.2.2 Relationships und Kardinalitäten von Relationships

- Eine Relationship ist eine Beziehung zwischen Entities.
- Die Kardinalität einer Relationship gibt an, wie viele Entities auf jeder Seite maximal an der Relationship beteiligt sind.
- Relationships können Attribute aufweisen.

#### 4.2.2.2 Relationships und Kardinalitäten von Relationships

- Beispiel 4.3 Relationships und Kardinalitäten von Relationships

„**ist zugelassen auf**“ ist eine Relationship zwischen einem Auto einer Person. Diese Relationship hat das **Attribut** „seit“.

„**fährt**“ ist eine Relationship zwischen einem Auto und einer Person.

„**gehört zu**“ ist eine Relationship zwischen einem Auto und einer Zulassungsbescheinigung.

---

**1:n-Relationship:** ein Auto ist zugelassen auf maximal eine Person. Auf eine Person können mehrere Autos zugelassen sein.

**n:m-Relationship:** jede Person kann mehrere Autos fahren. Jedes Auto kann von mehreren Personen gefahren werden.

**1:1-Relationship:** zu jedem Auto kann es maximal eine Zulassungsbescheinigung geben. Jede Zulassungsbescheinigung gehört zu maximal einem Auto.

## 4 Datenbankentwurf

## 4.3.1 Entity-Relationship-Modellierung mit Chen-Notation

- ER-Modelle sind Diagramme
- **Peter Chen Notation für ER-Diagramme:**
  - Rechteck für Entities
  - Raute für Relationships
  - Ovale für Attribute
  - Durchgezogenen Verbindungslinien verbinden die Elemente eines ER-Diagramms.
  - Die identifizierenden Attribute kann man unterstreichen. (Abbildung 4-3 zeigt ein Beispiel)
- **Kardinalitäten** von Relationships werden an den Verbindungslinien angeschrieben wie in Abbildung 4-4.

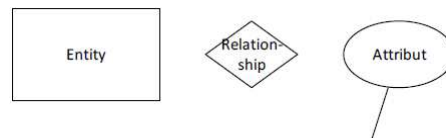


Abbildung 4-2 Symbole der Entity-Relationship-Modellierung

## 4 Datenbankentwurf

## 4.3.1 Chen Notation – Beispiel

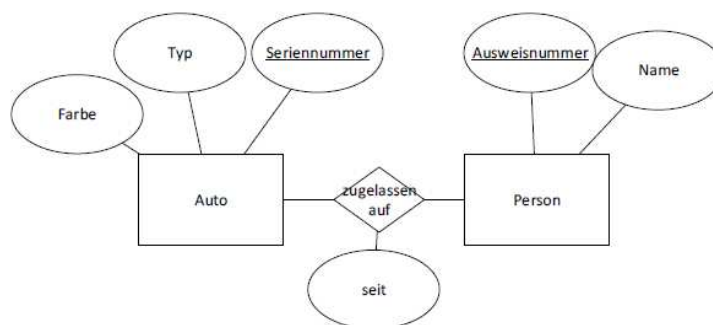


Abbildung 4-3 Beispiel Entity-Relationship-Diagramm

## 4 Datenbankentwurf

## 4.3.1 Chen Notation - Kardinalitäten

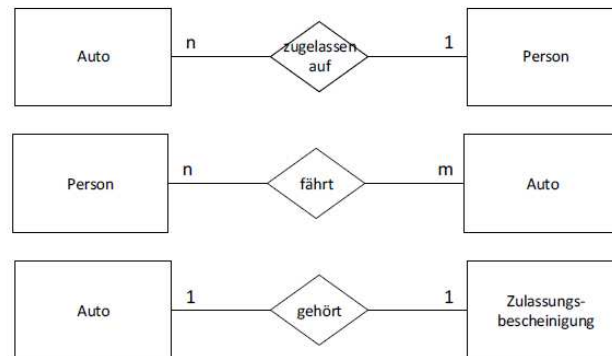


Abbildung 4-4 Kardinalitäten von Relationships

## 4 Datenbankentwurf

## 4.3.2 Beispiel

## ▪ Beispiel 4.4 ER-Modellierung Hochschule

Anmerkung: Das Beispiel ist vereinfacht, in einer echten Anwendung wäre die Datenbank wesentlich komplexer!

Eine Hochschule beginnt, eine Datenbank aufzubauen.

Fahrplan:

1. Welche Entities gibt es?
2. Welche Attribute haben sie? Welche sind identifizierende Attribute?
3. In welchen Beziehungen (Relationship) stehen sie zueinander?
4. Welche Attribute haben die Relationships?
5. Welche Kardinalitäten haben die Relationships?

## 4 Datenbankentwurf

## 4.3.2 Beispiel

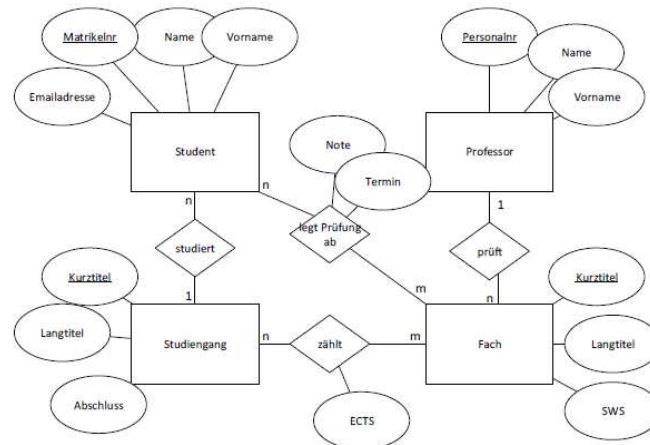


Abbildung 4-5 ER-Diagramm Hochschule in Chen-Notation

## 4 Datenbankentwurf

## 4.3.3 Vom Entity Relationship-Diagramm zum relationalen Datenbankschema

- Ein **Entity-Relationship-Diagramm** kann **eindeutig** in ein **relationales Datenbankschema** **überführt** werden.
- **1. Entities:**
  - a) für jedes Entity eine Tabelle erzeugen
  - b) jedes Attribut des Entity wird zu einer Spalte der Tabelle
  - c) identifizierende Attribute des Entities werden zu Primärschlüsselattributen der Tabelle
- **2. 1:n-Relationships:**
  - a) die identifizierenden beziehungsweise Schlüsselattribute der 1-Seite als Fremdschlüssel in die Tabelle der n-Seite aufnehmen
  - b) Wenn die Relationship eigene Attribute hat: Attribute der Relationship als Attribute in die Tabelle der n-Seite aufnehmen
- **3. n:m- Relationships:**
  - a) für jede n:m-Relationship eine neue Tabelle erzeugen
  - b) identifizierende Attribute beider Seiten als Fremdschlüssel in die neue Tabelle aufnehmen
  - c) daraus den Primärschlüssel der neuen Tabelle bilden
  - d) Wenn die Relationship eigene Attribute hat: Attribute der Relationship als Attribute in die neue Relation aufnehmen
- 4. Prüfen, ob das Datenbankschema die 3. Normalform erfüllt, und falls nötig **normalisieren**.

## 4 Datenbankentwurf

## 4.3.3 Vom Entity Relationship-Diagramm zum relationalen Datenbankschema

## ▪ Beispiel 4.5 Relationales Datenbankschema zum ER-Modell Hochschule

Student					Professor		
Matrikelnr	Name	Vorname	Emailadresse	Studiengang_Kurztitel	Personalnr	Name	Vorname
1315	Müller	Andi	mueller8113@yahoo.com	WI	23	Ganz	Giulia
1319	Meyer	Uli	meyer2221@gmail.de	WI			
1321	Braun	Ute	schlumpf@gmx.de	MMB	24	Schwer	Sabine

Studiengang				Fach	
Kurztitel	Langtitel	Abschluss	Professor_Personalnr	Kurztitel	Langtitel
WI	Wirtschaftsingenieurwesen MB	Bachelor	23	PM	Projektmanagement
MMB	Maschinenbau	Master	24	Phys	Physik
				BWL	Betriebswirtschaftslehre
				Mathe	Mathematik für Ingenieure

ECTS				Prüfungsergebnis			
Fach_Kurztitel	Studiengang_Kurztitel	ECTS		Matrikelnr	Fach_Kurztitel	Note	Termin
PM	WI	5		1315	PM	1	04.02.2015
PM	MBB	4		1315	Phys	1	12.01.2015
BWL	WI	6		1315	BWL	2	13.01.2015
BWL	MBB	4		1319	BWL	3	13.01.2015
Mathe	WI	5		1319	PM	2	01.02.2014
Phys	WI	5		1321	PM	1	04.02.2015
				1321	Mathe	1	11.01.2013

Abbildung 4-6 Hochschuldatenbank zum ER-Diagramm

## 4 Datenbankentwurf

## 4.4 Entity-Relationship-Modellierung in Krähenfuß-Notation

- Alternative zur Chen-Notation
- **Primärschlüsselattribute** mit PS
- **Fremdschlüsselattribute** mit FS

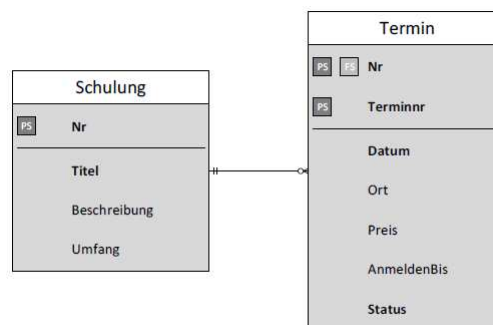


Abbildung 4-7 Beispiel Krähenfuß-Notation in der Miniwelt Schulungen



## 4 Datenbankentwurf

## 4.4.1 Entities

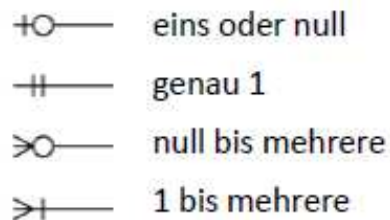
- Chen-Notation durch **Rechtecke** symbolisiert
- Vorteil:** kompakter und „vielleicht“ übersichtlicher Modellierung mittels UML Notation möglich
- Nachteil:** auf Papier nicht so leicht zu erweitern, da das Rechteck nicht mitwachsen kann, wenn weitere Attribute dazukommen



## 4 Datenbankentwurf

## 4.4.2 Kardinalitäten

- Krähenfüße geben an, wie viele Entities maximal und mindestens an einer Relationship partizipieren

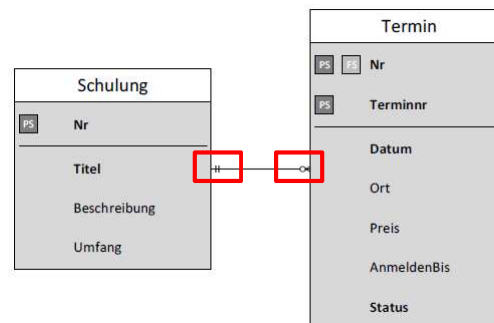


## 4 Datenbankentwurf

## 4.4.2 Kardinalitäten

- Beispiel 4.6 Kardinalitäten in Krähenfuß-Notation in der Miniwelt „Schulungen“

Es kann pro Schulung mehrere Termine geben, möglicherweise ist aber für eine Schulung noch kein Termin geplant. Also pro Schulung kein oder mehrere Termine:  $\infty$   
 Zu jedem Schulungstermin gehört genau eine Schulung:  $1$



## 4 Datenbankentwurf

## 4.4.3 Identifizierende und nicht-identifizierende Relationships

- Relationship** ist in der Krähenfuß-Notation **nur durch ein Linie** symbolisiert
- zwei Arten von Relationships:
  - Identifizierend:**
    - schwaches Entity auf der n-Seite
    - schwache Entity durch das starke Entity identifiziert
    - Transformation in das relationale Datenmodell:
      - Primärschlüssel des starken Entities einen Teil des Primärschlüssels des schwachen Entities
      - schwache Entity besitzt einen zusammengesetzten Primärschlüssel
      - Primärschlüssel des starken Entities ist Fremdschlüssel im schwachen Entity und gleichzeitig Primärschlüsselattribut
  - Nicht-identifizierend**

Identifizierend:  $1$  —  $\infty$

Nicht-identifizierend:  $1$  - - -  $\infty$

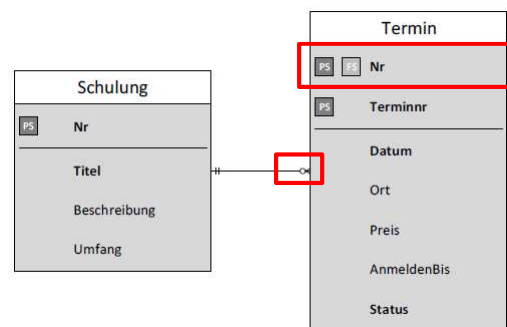
Abbildung 4-9 Krähenfuß-Notation für identifizierende und nicht-identifizierende Relationships

## 4 Datenbankentwurf

## 4.4.3 Identifizierende und nicht-identifizierende Relationships

- Beispiel 4.7 Identifizierende Relationship in der Miniwelt „Schulungen“

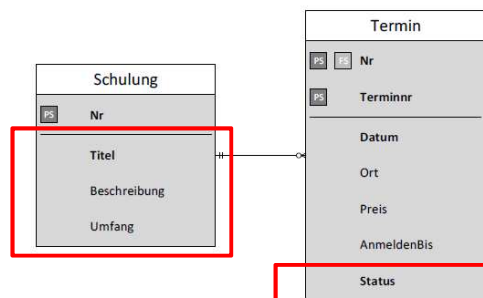
In der Anwendungsdomäne „Schulung“ werden die Termine pro Schulung durchnummeriert, daher ist im Krähfuß-Modell in Abbildung 4-7 eine identifizierende Relationship modelliert.



## 4 Datenbankentwurf

## 4.4.4 Notwendige und optionale Attribute

- Attribute eines Entities **zwingend einen Wert** enthalten
  - Attribute **fett drucken** oder durch besondere Symbole
- Attribute auch *Nullwerte* (das heißt keine Werte) zulässig
- Primärschlüsselattribute müssen zwingend einen Wert enthalten.

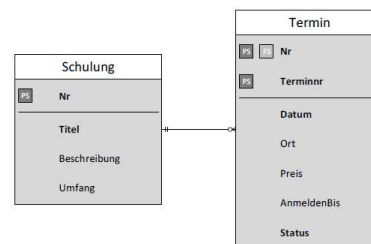


## 4 Datenbankentwurf

## 4.4.4 Notwendige und optionale Attribute

- Beispiel 4.8 Zwingend erforderliche Attribute in der Miniwelt Hochschule

In der Miniwelt der Schulungen in Abbildung 4-7 muss für jede Schulung eine Nummer und ein Titel angegeben sein. die ausführliche Beschreibung kann auch offen bleiben. Für jeden Termin muss die Nr der Schulung angegeben sein, die stattfinden soll, sowie die Nummer des Termins und das geplante Datum. Der Status zeigt an, ein Schulungstermin in Planung, offen für Anmeldungen oder bereits vergangen ist. Der Status muss immer angegeben werden, Preis und Anmeldeschluss können auch offen bleiben.



## 4 Datenbankentwurf

## 4.4.5 n:m-Relationships

- n:m-Relationships** sind in der Krähfuß-Notation **unüblich**

- Alternative:** zwei 1:n-Relationships und ein neues Entity

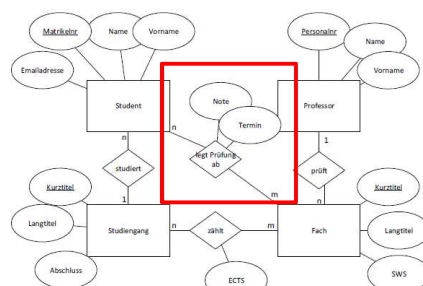


Abbildung 4-5 ER-Diagramm Hochschule in Chen-Notation

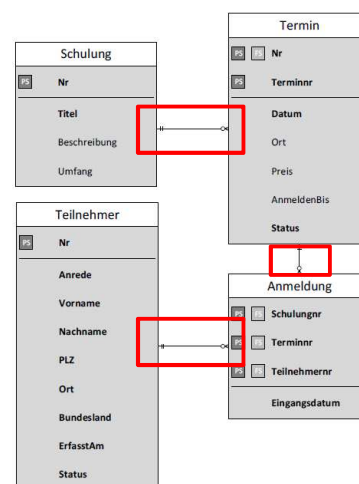


Abbildung 4-10 Krähfuß-Notation n:m-Relationship aufgelöst in zwei 1:n-Relationships

## 4 Datenbankentwurf

## 4.4.6 Datentypen von Attributen

- Werte haben einen bestimmten Datentyp
- logischen Ebene:
  - Datentypen definieren grundsätzliche Einschränkung
- technisch-physischen Ebene:
  - Effizienz (Antwortgeschwindigkeit) und den Speicherplatzverbrauch abhängig
- Standards für relationale Datenbanken:
  - erleichtern Daten zwischen verschiedenen RDBMS (relationalen Datenbankmanagementsystemen) auszutauschen oder das RDBMS zu wechseln
- Basis-Datentypen in Tabelle rechts entspricht standardkonforme RDBMS

Tabelle 4-1 Datentypen in relationalen Datenbanken

Datentyp	Beschreibung	Beispielwerte
INT	Ganzzahlen, ist für künstliche Schlüssel geeignet	6
DEC(m, n)	Dezimalzahlen mit m Stellen insgesamt, davon n Nachkommastellen	DEC(4,2): 12.34
FLOAT	Gleitkommazahl	12.0634, 56663.455669
DOUBLE		
VARCHAR(n)	Zeichenketten variabler Länge mit bis zu n Zeichen, überzählige Zeichen werden abgeschnitten	VARCHAR(8): '01f', 'Sybille', 'Matthia'
CHAR(n)	Zeichenkette fester Länge mit genau n Zeichen, fehlende Zeichen mit Leerzeichen aufgefüllt, überzählige abgeschnitten	CHAR(3): 'DE ', 'CH ', 'AUS'
DATE	Datum	'2015-10-10'
TIME	Zeit	'12:30:00'
DATETIME	Termin bestehend aus Datum und Uhrzeit	'2015-10-10 12:30:00'
TIMESTAMP	fixer Zeitpunkt unabhängig von Zeitzone, auch mit Millisekunden	'2015-10-10 12:30:00'
BLOB(N)	Binary Large Object („Bittkumpen“) zur Speicherung beliebiger Daten, zum Beispiel Bilder, bis maximal n MByte	BLOB(20M): bis 20MByte xy<01fe":1Aq1lycNA"ä02 ow4_2os::Ät1y00cFpM:"E y,11:-v,a0h0",c09A.

## 4 Datenbankentwurf

## 4.4.6 Datentypen von Attributen

- Beispiel 4.9 Krähfuß-Diagramm der Miniwelt Hochschule

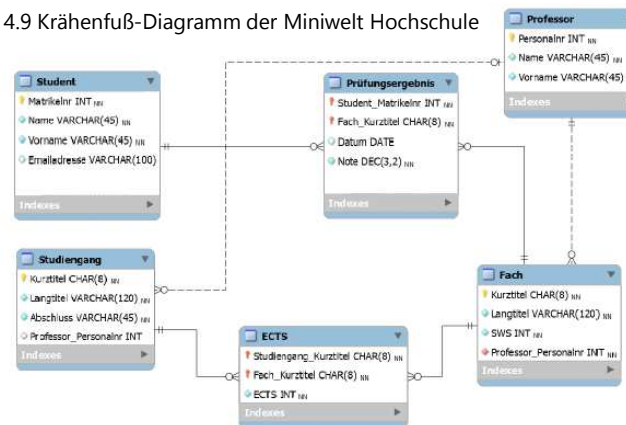


Abbildung 4-11 Krähfuß-Diagramm der Miniwelt Hochschule mit Datentypen

## 4 Datenbankentwurf

## 4.5.1 Visualisierungssoftware zum Zeichnen von Diagrammen: MS Visio

- Die Chen-Notation benötigt nur wenige und einfache Symbole
- sehr zeichenfreundlich
- Diagramme sind leicht von Hand zu erstellen

- Visualisierungsprogramme
  - MS Visio

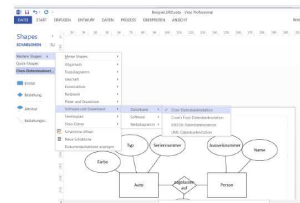
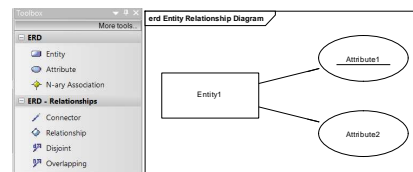


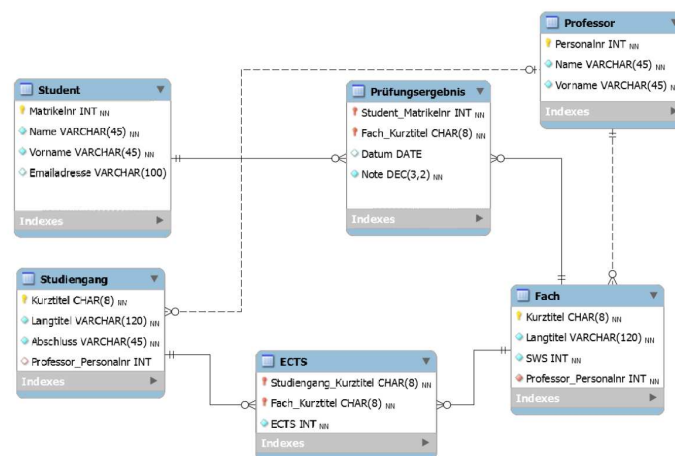
Abbildung 4-12 ER-Diagramme erstellen in MS Visio

- Sparx Enterprise Architect



## 4 Datenbankentwurf

## 4.5.2 Modellierungssoftware zum Erstellen logischer Modelle und physischer Datenbanken durch Forward Engineering



## 4 Datenbankentwurf



## 4.6 Übungen

## Aufgabe 4.1 ER-Modellierung Wareneingang

Ein Unternehmen möchte die eingegangenen Lieferungen in einer Datenbank erfassen. Jede Lieferung bekommt eine laufende Nummer LiefNr. Zu jeder Lieferung gehört ein Lieferschein mit einer LieferscheinNr (die Nummer des Lieferscheins beim Lieferanten), ein Lieferant und ein Eingangsdatum. Jeder Lieferant hat eine LNr, einen Namen und ein Land. Zu jeder Lieferung wird erfasst, welche Artikel in welcher Menge darin enthalten sind. Eine Lieferung kann mehrere Artikel enthalten. Ein Artikel hat eine Artikelnummer, eine Bezeichnung und eine Mengeneinheit. Jeder Artikel gehört zu einer Artikelgruppe. Jede Artikelgruppe hat eine Nummer (AgNr) und eine Langbezeichnung.

- a) Modellieren Sie die Miniwelt Wareneingang als ER-Diagramm in der Chen-Notation.
- b) Zeichnen Sie die Tabellen (Relationen), die diesem Diagramm entsprechen und schreiben Sie in jede Tabelle zwei plausible Beispiel-Datensätze.
- c) Geben Sie die Relationenschemata zu den Tabellen an.

## 4 Datenbankentwurf



## 4.6 Übungen

## ▪ Aufgabe 4.2 ER-Modellierung Beschaffung

Das Unternehmen Saubermann produziert Reinigungsmittel und bezieht dazu Rohstoffe von verschiedenen Lieferanten. Die Daten zur Beschaffung sollen in einer relationalen Datenbank abgelegt werden. Jeder Rohstoff hat eine eindeutige Nummer, eine Bezeichnung und eine Gefahrenklasse, die als Ganzzahl angegeben wird (1,2,3,4). Jeder Lieferant hat eine eindeutige Lieferantenummer; außer dieser sind Name und Adresse relevant. Für jeden Lieferanten soll eine Lieferantenbewertung (A, B oder C) hinterlegt werden können. Lieferanten können mehrere unterschiedliche Rohstoffe anbieten und manche Rohstoffe sind bei mehreren Lieferanten im Angebot. In der Datenbank soll hinterlegt werden, welcher Lieferant welchen Rohstoffe zu welchem Preis und mit welcher Mindestabnahmemenge anbietet. Für einige der Rohstoffe ist jeweils ein Hauptlieferant festgelegt, bei dem der Rohstoffe normalerweise bestellt wird.

- a) Modellieren Sie die Miniwelt Beschaffung als ER-Diagramm in der Chen-Notation.
- b) Zeichnen Sie die Tabellen (Relationen), die diesem Diagramm entsprechen und schreiben Sie in jede Tabelle zwei plausible Beispiel-Datensätze.
- c) Geben Sie die Relationenschemata zu den Tabellen an.

## 4 Datenbankentwurf



## 4.6 Übungen

- Aufgabe 4.3 ER-Modellierung und Implementation Autohaus

Für ein aufstrebendes Autohaus sollen Sie ein Datenbank entwickeln. Zu jedem Auto möchte man die Seriennummer, Marke, Typ und Farbe speichern können. Kunden haben eine Kundennummer, Name, Vorname, Postleitzahl, Stadt und Straße. Die Angestellten des Autohauses haben eine Personalnummer, Name und Vorname und ein Eintrittsdatum. Jedem Kunden ist ein Verkäufer zugeteilt, der ihn betreut. Wenn ein Kunde ein Auto kauft, möchte man das Verkaufsdatum, den Preis und den Verkäufer abspeichern.

- 
- a) Modellieren Sie die Miniwelt Autohaus als ER-Diagramm in der Chen-Notation.
  - b) Modellieren Sie die Datenbank als Krähenfuß-Diagramm inklusive Datentypen in der MySQL Workbench. Das Modell soll *IhrName\_autohaus* heißen.
  - c) Erzeugen Sie die dazugehörige Datenbank per Forward Engineering.

Herzlichen Dank  
für Ihre Aufmerksamkeit!

Hugo Colceag

**MHP**  
Management- und IT-Beratung GmbH

Film- und Medienzentrum | Königsallee 49 | D-71638 Ludwigsburg  
Telefon +49 (0)7141 7856-0 | Fax +49 (0)7141 7856-199  
eMail [info@mhp.com](mailto:info@mhp.com) | Internet [www.mhp.com](http://www.mhp.com)