

```
SELECT s.Name, s.MatrNr, sum(k.Dauer)
FROM Studentin s, anmelden a, Kurs k
WHERE s.MatrNr=a.MatrNr
AND a.KursNr=k.KursNr
GROUP BY s.MatrNr
```

Datenbanken: Entwurf und praktische Grundlagen

admina.at 2005
WIT, TU Wien

Maria Oelinger, Universität Duisburg ~~Essen~~

Lernziele

- Wie entwerfe ich eine Datenbank?
- Wie kann ich Daten einfügen, bearbeiten oder löschen?
- Wie kann ich Daten abfragen?

Organisatorisches

- 2 Tage
 - Entwurf / praktische Arbeit an der DB
- Vortrag und Übung abwechselnd
- Programming-in-pairs
- Zeitplan
 - Mittagspausen 12:30 h
 - Nachmittagspausen nach Bedarf
 - Zeit für Fragen

Inhalt

- Online-DB-Beispiel
- Grundlagen
- ER-Diagramme
- Relationales Modell
- SQL
 - Anfragen
 - DB erstellen

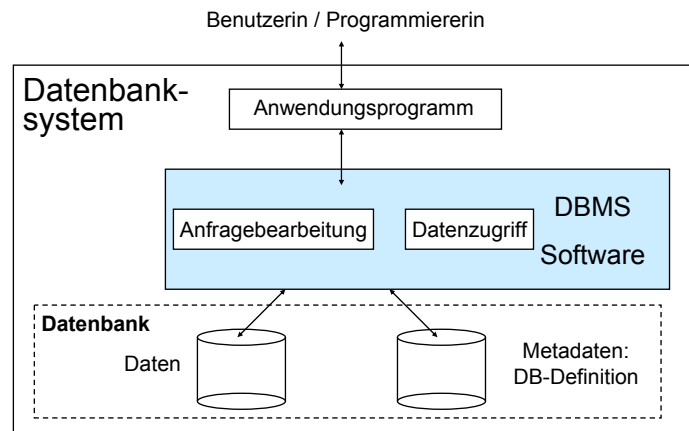
1. Grundlagen Datenbanken

- Begriffsbestimmung DB, DBS, DBMS
- Aufbau eines DBS
- Datenmodelle
- Projektübung

Datenbank und Datenbanksystem

- Anwendungen: Reisebuchungen, Bilder, Bücher, ...
- Beispiele: Oracle, Sybase, dBase, O2, MySQL, ...
- Motivation (Unterschied zu Dateisystem):
 - Redundanz und Inkonsistenz vermeiden
 - Datenverknüpfungen
 - Mehrnutzerbetrieb
 - Weniger Pflegeaufwand
 - Sicherheit

Datenbank und Datenbanksystem



Datenbank und Datenbanksystem

- DBMS: Programme zum Erzeugen und Unterhalten einer DB, z. B. MySQL
 - DB definieren: Datentypen und Strukturen
 - DB erzeugen: Eingeben der Daten
 - DB manipulieren: Anfragen und Updates
- Beispiel
admina.at DB (Studentin, Kurs, Anmeldung)

Datenbank und Datenbanksystem

- DB definieren
Studentin(Name:string, Email:string, MatrNr:integer)
Kurs(Name:string, KursNr:string, Dauer:integer)
Anmeldung(MatrnNr:integer, KursNr:string)
- DB erzeugen
Daten der Kurse und Teilnehmerinnen speichern
- Anfragen
Welche Studentinnen besuchen den DB Kurs 2005?

Datenbankschema

- Unterscheidung zwischen Beschreibung (Struktur) und Daten
- Beschreibung: DB-Schema=Metadaten

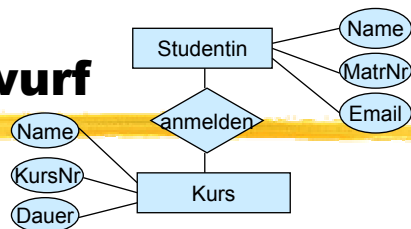
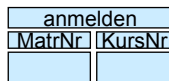
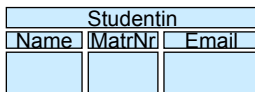


Arbeiten mit DB

- Zugriff über Anwendungsprogramm
- Anfragen mit SQL
- In Programme eingebettete Anfragen
- Entwurf und Manipulation
- Benutzererkennung und- rechte, Datensicherheit

2. Datenbankentwurf

- ER-Modellierung
(konzeptioneller Entw.)



- SQL

```
CREATE TABLE Studentin(Name varchar(15),  
                        MatrNr decimal,  
                        Email varchar(15))
```

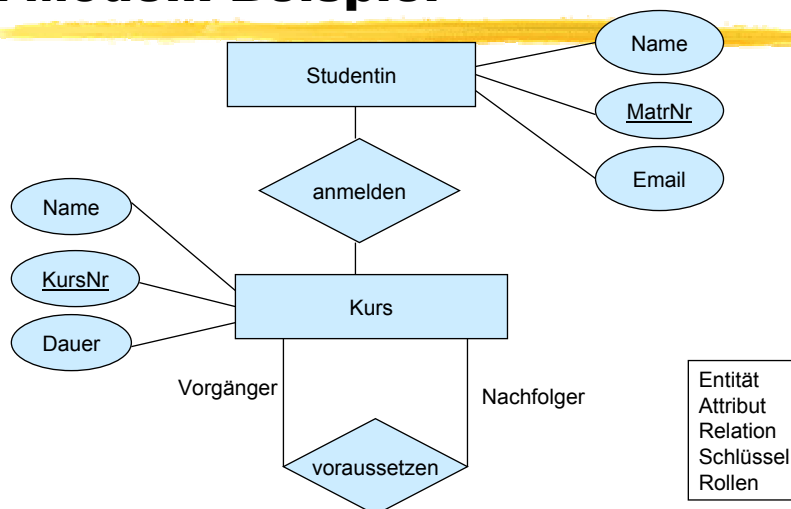
- Ziel: Eigener DB-Entwurf

- Relationales Modell
(logischer Entwurf)

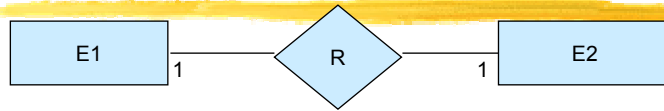
ER-Modell

- Verschiedene Variationen in Gebrauch
- Konzept:
 - Entität: Ding der realen Welt (Haus, Person)
 - Attribut: Eigenschaften von Entitäten
 - Ein Entity hat einen Wert für jedes seiner Attribute
 - Beziehungen: Relationen zwischen Entitäten

ER-Modell: Beispiel

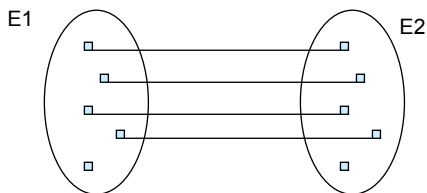


ER-Modell: Kardinalitäten



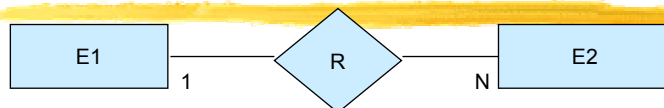
- 1:1 Beziehung

Jedem Entity aus E1 ist höchstens ein Entity aus E2 zugeordnet und umgekehrt

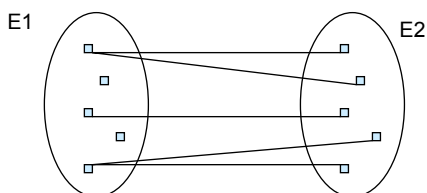


Beispiele:
Land hat Königin
Frau heiratet Mann

ER-Modell: Kardinalitäten

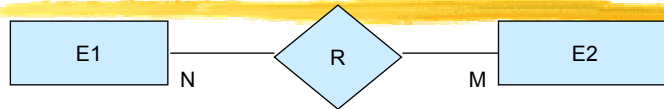


- 1:N Beziehung: jedem Entity aus E1 sind beliebig viele Entities (mind. 1) aus E2 zugeordnet, jedem Entity aus E2 aber höchstens eins aus E1

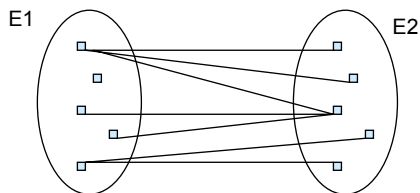


Beispiele:
Mutter hat Kind
Firma beschäftigt Angestellte

ER-Modell: Kardinalitäten

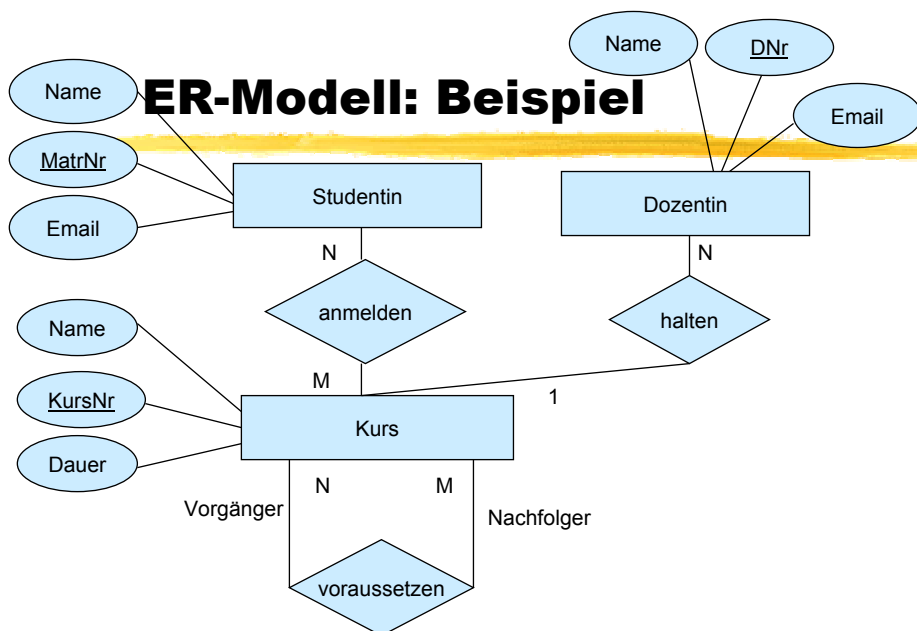


- N:M Beziehung: jedem Entity aus E1 sind beliebig viele Entities aus E2 zugeordnet und umgekehrt



Beispiele:
Studentin hört Vorlesung
Person liest Buch

ER-Modell: Beispiel



ER-Modell: Übung

- ER-Modelle mit Schlüsseln und Kardinalitäten für das Projekt (folgt sofort)
- Welche Bedingungen lassen sich nicht im ER-Modell ausdrücken?

Projektübung

- Beschreibung

Wir wollen eine Wissensdatenbank aufbauen. Verschiedene Personen sollen erfasst werden, die ein Wissensnetzwerk bilden. Eine Person kann einem oder mehreren Fachgebieten zugeordnet werden. Zu einem Fachgebiet kann es mehrere Expertinnen geben. Fachgebiete werden charakterisiert durch Schlagwörter. Jede Person hat kein, ein oder mehrere Spezialgebiete. Spezialgebiete werden in Typen eingeteilt: Dies Thema ist meine Seminararbeit, jenes Deine Diplomarbeit etc.

- ER-Diagramm. Jetzt.



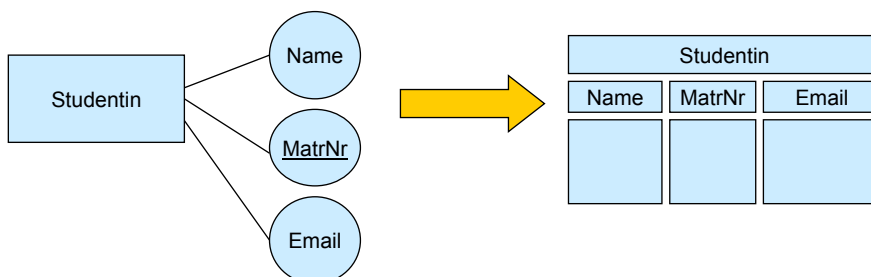
Relationales Modell

- Sehr weit verbreitet
- Konzept:
 - Relationen (Tabellen)
 - Zeilen entsprechen Datenobjekten
 - Spaltennamen sind Attribute
 - Wertebereiche für Attribute
 - Mengenorientierte Verknüpfungen

Studentin		
Name	MatrNr	Email

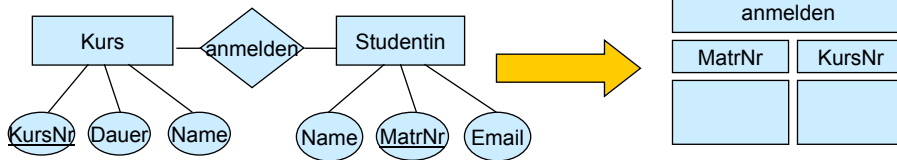
ER zu relationalem Modell

- Entität
 - Abbilden auf Relation
 - Attribute zu Tabellenspalten (=Attribute)

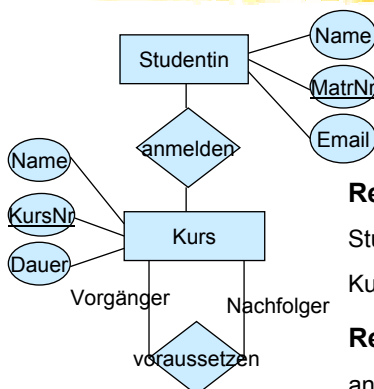


ER zu relationalem Modell

- Beziehungen
 - Ebenfalls Abbilden auf Relation
 - Schlüsselattribute aller beteiligten Entitäten übernehmen (= Fremdschlüssel)
 - Attribute der Beziehung übernehmen
 - Evtl. Umbenennungen



Relationales Modell: Beispiel



Relationale Darstellung von Entitytypen

Studentin: {[MatrNr:integer, Name:string, Email:string]}

Kurs: {[KursNr:string, Name:string, Dauer:string]}

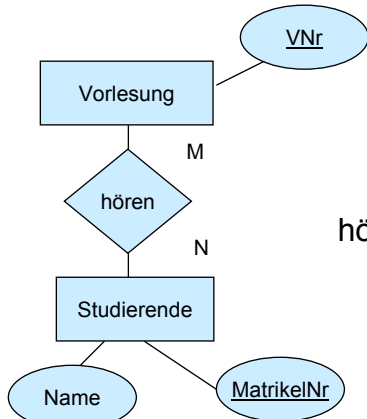
Relationale Darstellung von Relationen

anmelden: {[MatrNr:integer, KursNr:string]}

voraussetzen: {[VoKursNr:string, NaKursnr:string]}

Relationales Modell: Schlüssel

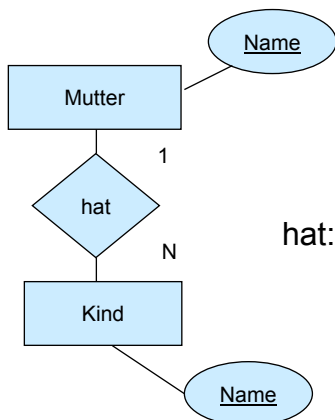
- N:M-Beziehung: Menge aller Fremdschlüssel als Schlüssel der Relation



hören: {[VNr:integer, MatrikelNr:integer]}

Relationales Modell: Schlüssel

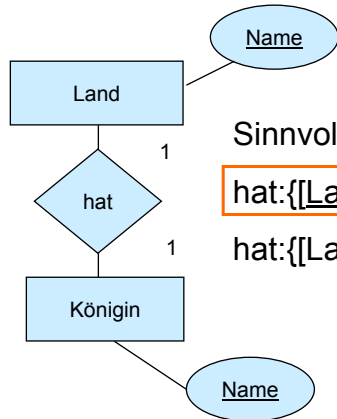
- 1:N-Beziehung: Schlüssel der bestimmenden Entität als Schlüssel der Relation



hat: {[KindName:string, MutterName:string]}

Relationales Modell: Schlüssel

- 1:1-Beziehung: einer der Fremdschlüssel als Schlüssel der Relation



Sinnvolle Wahl treffen:

hat: {[Landesname:string, Königinname:string]}

hat: {[Landesname:string, Königinname:string]}

Relationales Modell: Schemaverfeinerung

- Eliminierung von Relationen möglich
- NUR Relationen mit gleichem Schlüssel !

Land(Landesname),
Königin(Königinname)
hat:(Landesname, Königinname)



Land(Landesname, regiertVon),
Königin(Königinname)

regiertVon ist Fremdschlüssel
auf Königinname

Relationales Modell: Übung

- Umformen der ER-Modelle der Beispiele in relationale Modelle
- Relationales Modell für das Projekt

Structured Query Language

- Ursprünglich SEQUEL (Structured English Query Language)
- 1986: Standard SQL1 von ANSI und ISO
- 1999: SQL99, SQL/MM
- Anfragesprache in den meisten kommerziellen DBMS

SQL: Mächtigkeit

- Datendefinition(DDL)
 - Erzeugen / Löschen von Datenbanken und Relationen
 - Einfügen / Löschen von Attributen
- Datenmanipulation (DML)
 - Dateneingabe und -veränderung
 - Anzeige von Relationen
 - Anfragen
- Viewmanagement
- Integritätskontrolle
- Transaktionsmanagement

SQL: Schemadefinition

- Befehle
 - `create database`
 - `create table`
- Beispiel

```
create table Studentin
(MatrnNr integer primary key,
Name varchar(20),
Email varchar(20) );
```

SQL: Datentypen

- Zahlen

http://www.mysql.com/doc/de/Column_types.html

- `integer`

- Zeichenketten

- `varchar(n)`,

- `char(n)` (feste Länge, mit Leerzeichen aufgefüllt)

- Datum

- `date`

SQL: Dateneingabe

- "Zu Fuss"

- `use <datenbank>;`

- `insert into table [(column1,...)]
values (value1,...);`

- Als Batch

- Daten in Datei speichern + alle einlesen

- Je nach DB-System

- Zur Kontrolle eine Abfrage

- `select * from table;`

SQL: Daten entfernen

- Daten aus Tabelle entfernen
`delete from table [where bedingung];`
- Alle Daten aus Tabelle entfernen
`delete from table;`
- Tabelle entfernen
`drop table table;`
 - Vorsicht! Löscht alle Daten und die Tabellendefinition

SQL: Übung

- MySQL starten
- Anlegen eines Datenbankschemas zum Projekt (analog relationalem Entwurf)
- Einfügen der Daten (selber ausdenken!)

3. Datenbankanfrage

- Inhalt
 - Basisoperationen relationaler Sprachen
 - Mit SQL Daten manipulieren
 - SQL-Übungen
 - Anfragen an die Projektdatenbank

Basisoperationen auf relationalen Datenbanken

- Selektion

studentin		
Name	MatrNr	Email



Name	MatrNr	Email

- Projektion

studentin		
Name	MatrNr	Email



Email

Basisoperationen auf relationalen Datenbanken

- Union

studentinW		
Name	MatrNr	Email

studentinDU-E		
Name	MatrNr	Email



Name	MatrNr	Email

- Kreuzprodukt

anmelden	
MatrNr	KursNr

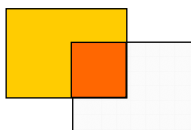
studentin		
Name	MatrNr	Email



Name	MatrNr	Email	MatrNr	KursNr

Basisoperationen auf relationalen Datenbanken

- Schnitt



studentinW		
Name	MatrNr	Email

studentinDU-E		
Name	MatrNr	Email



Name	MatrNr	Email

- Differenz



studentinW		
Name	MatrNr	Email

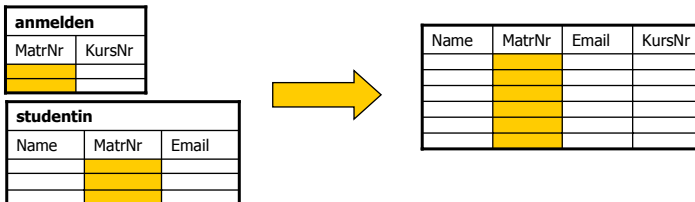
studentinDU-E		
Name	MatrNr	Email



Name	MatrNr	Email

Basisoperationen auf relationalen Datenbanken

- Natürlicher Join
- = Kreuzprodukt, Auswahl gleicher Einträge in übereinstimmenden Spalten, Projektion



SQL-Anfrage

- Basiskonstrukt
`SELECT attribut`
`FROM tabelle`
`WHERE bedingung`

- Beispiel

```
SELECT registertVon  
FROM Land  
WHERE Landname=Dänemark
```

Projection (yellow arrow pointing to 'registertVon')
Selektion (yellow arrow pointing to 'WHERE Landname=Dänemark')

SQL: Join



```
SELECT Name FROM Studentin s, Anmeldung a
WHERE s.MatrNr=a.MatrNr
AND KursNr=20;
```

```
... AND KursNr>20;
```

Joins über beliebig viele Tabellen möglich

→ Performance-Problem

SQL: Übung I

- Beispiel: Folgendes Schema BIB(Buch,Autoren,kaufen,Bibliothek)
Buch(ISBN, Titel, Verlag, AutorenID)
Autoren(ID, Name, Vorname, Telefon)
kaufen(ISBN, BibName, BibOrt, Preis)
Bibliothek(Name, Ort, Strasse, www)
- Fragen formulieren in SQL
Wie heißt das Buch mit ISBN 1234567?
Wie heißt die Autorin des Buches mit ISBN 1234567?
Welche Bibliotheken haben Bücher von Rita Schmidt gekauft?
- Fragen für unser Projekt / zur Beispiel  formulieren 

SQL: Mengenoperationen

- Union
- (Sinnloses) Beispiel

```
SELECT Name
FROM Studentin, Anmeldung
WHERE KursNr=20
UNION
SELECT Name
FROM Studentin, Anmeldung
WHERE KursNr=30;
```
- Eliminiert automatisch Duplikate

SQL: Duplikate

- In normalen Anfragen
 - Keine Duplikatentfernung, da aufwändig
 - `SELECT DISTINCT`: Entfernt Duplikate
- In Mengenoperationen
 - Eliminiert Duplikate
 - `ALL`: Erhält Duplikate (... UNION ALL ...)

SQL: Aggregatsfunktionen

- Summe `sum`, Durchschnitt `avg`, Maximum `max`, Minimum `min`

```
SELECT SUM(dauer)
FROM kurs;
```

- Anzahl `count`

```
SELECT COUNT(*)
FROM kurs;
```

SQL: group by

- Für Aggregate
- Beispiel
Wieviele Kurse hat jede belegt?

```
SELECT MatrNr, count(*)
FROM Anmeldung
GROUP BY MatrNr;
```

- Gruppierungsattribut muss auch im Select-Statement stehen

SQL: Explizite Mengen

- Beispiel

```
SELECT MatrNr
FROM Anmeldung
WHERE KursNr IN (Kurs3, Kurs14)
```

- Auch möglich

```
IS NULL oder IS NOT NULL
```

SQL: Übung II



- Beispiel: Schema BIB(Buch,Autoren,kaufen,Bibliothek)

```
Buch(ISBN, Titel, Verlag, AutorenID)
Autoren(ID, Name, Vorname, Telefon)
kaufen(ISBN, BibName, BibOrt, Preis)
Bibliothek(Name, Ort, Strasse, www)
```

- Fragen formulieren in SQL

Wieviele Bücher hat Rita Schmidt veröffentlicht?

Wieviel hat jede Bibliothek für Bücher von Rita Schmidt bezahlt und wieviele haben sie gekauft?

- Fragen für unser Projekt / zur Beispiel  formulieren 

4. SQL: Schachtelungen

- Vollständige Anfragen im `WHERE` Statement einer anderen Anfrage

- Beispiel

```
SELECT Name, MatrNr
FROM Studentin
WHERE Name IN (SELECT Name FROM Dozentin WHERE...);
```

- `IN` vergleicht einen Wert mit einer Menge von Werten

SQL: Schachtelungen

- Erlaubt Rekursion

```
SELECT v1.Vorgänger FROM voraussetzen v1
WHERE v1.Nachfolger IN (
    SELECT v.Vorgänger FROM voraussetzen v, Kurs k
    WHERE v.Nachfolger=k.KursNr and k.Name="DB")
```

- Schachtelungen können meist umformuliert werden

SQL: Schachtelungen

- Weitere Operatoren

IN | = | < | > | <= | >= | !=

- Erweiterungen

- ANY (gilt, wenn wahr für manche)
- ALL (gilt, wenn wahr für alle)

- **Beispiel** Welcher Kurs dauert am längsten?

```
SELECT Name
FROM Kurs
WHERE dauer >= ALL (SELECT dauer FROM Kurs);
```

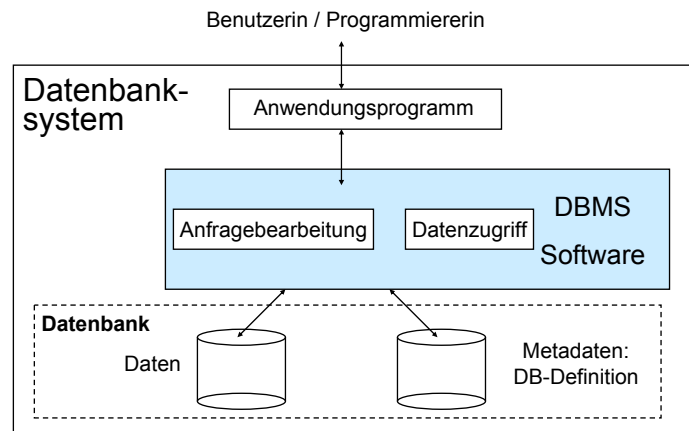
Verschiedenes

- Verschiedene weitere Themen, die im Kontext von Datenbanken eine Rolle spielen

- Sichten
- Normalformen
- Anfrageoptimierung
- Transaktionen
- Mehrbenutzersynchronisation

- Sicherheit
- Objektorientierte, objektrelationale DBen

Datenbank und Datenbanksystem



Referenzen

- Buchmann, Smolarek (2005) "SQL.MySQL 5", dpunkt Verlag
- Barthelmann, Loehr (2001) "SQL" www.informatik.uni-mainz.de/~barthel/WSGP01/17_72001/SQL.html
- Hinze (2002) "Einführung in Datenbanksysteme" www.inf.fu-berlin.de/lehre/SS02/DBS/
- Hinze, Oelinger (2003) "Kurs Datenbanken" <http://zeitung.informatica-feminale.de/kwiki/index.cgi?KursDatenbanken>
- Oelinger (2005) "Datenbanken: Entwurf und praktische Grundlagen" www.collide.info/~oelinger/