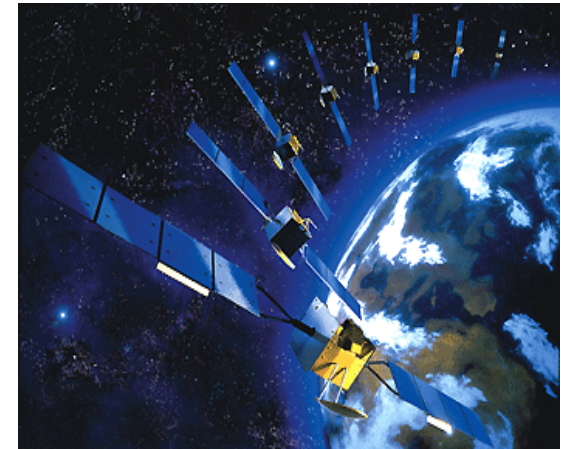




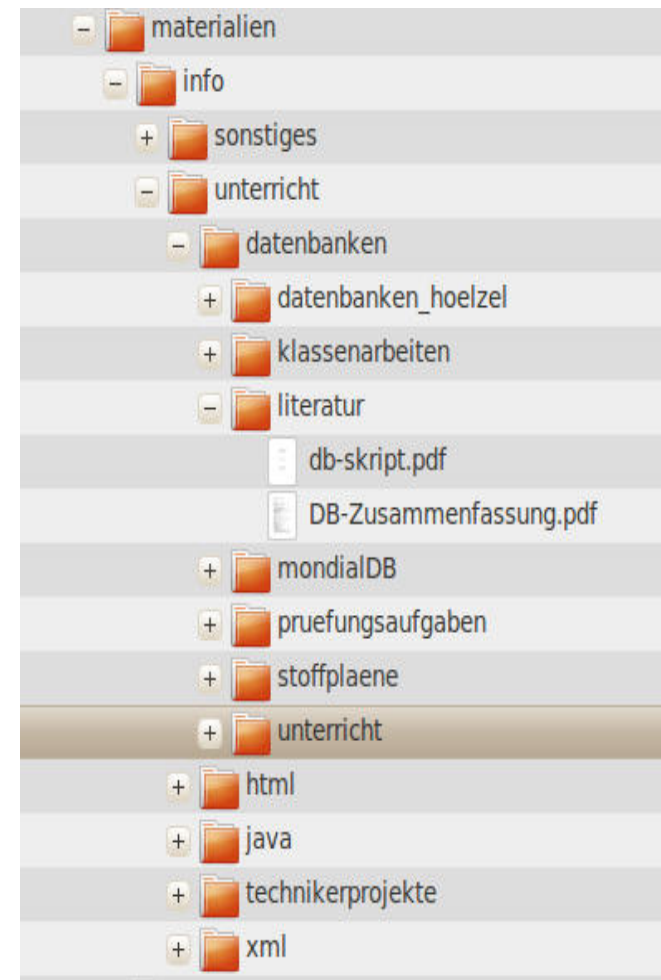
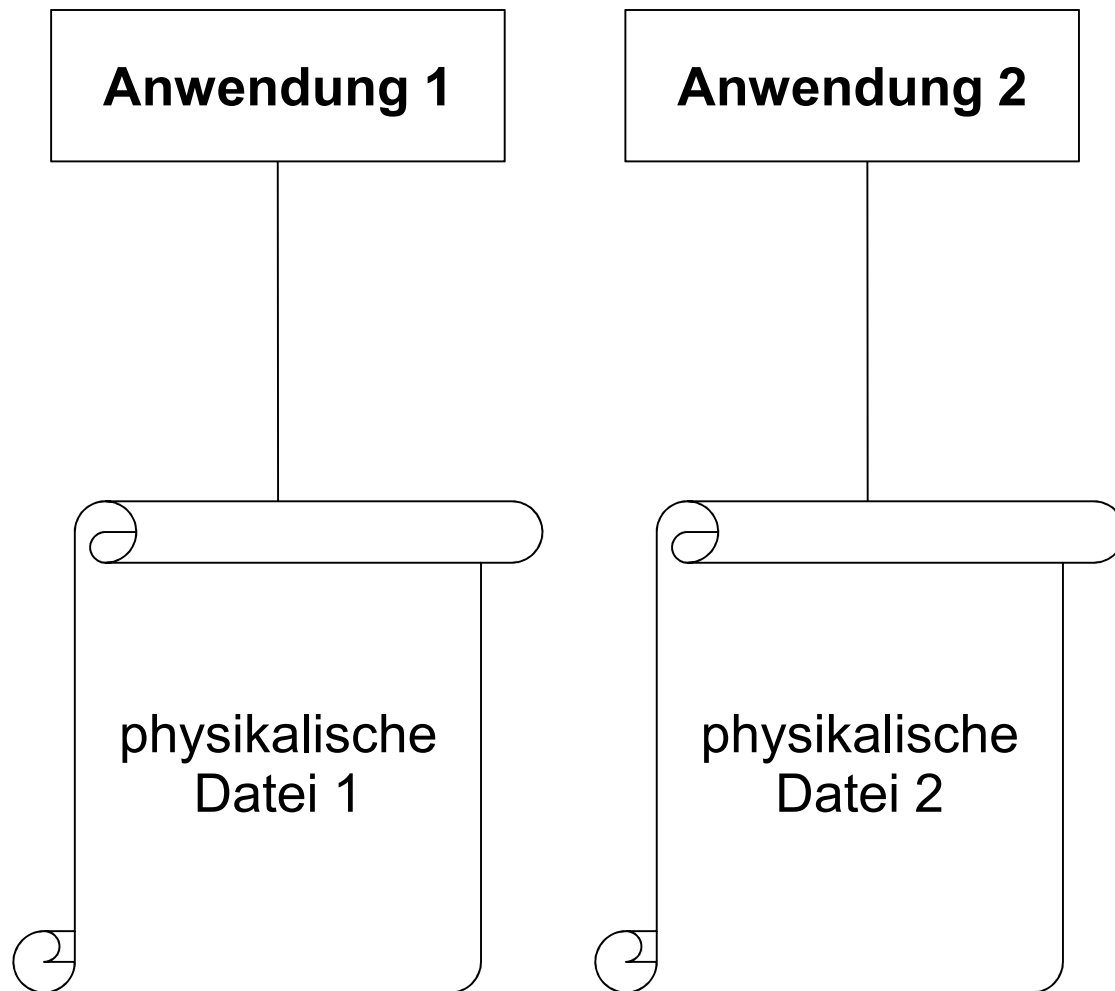
# ***Datenbanken Einführung***

***N. Nazar  
S. Baldes***

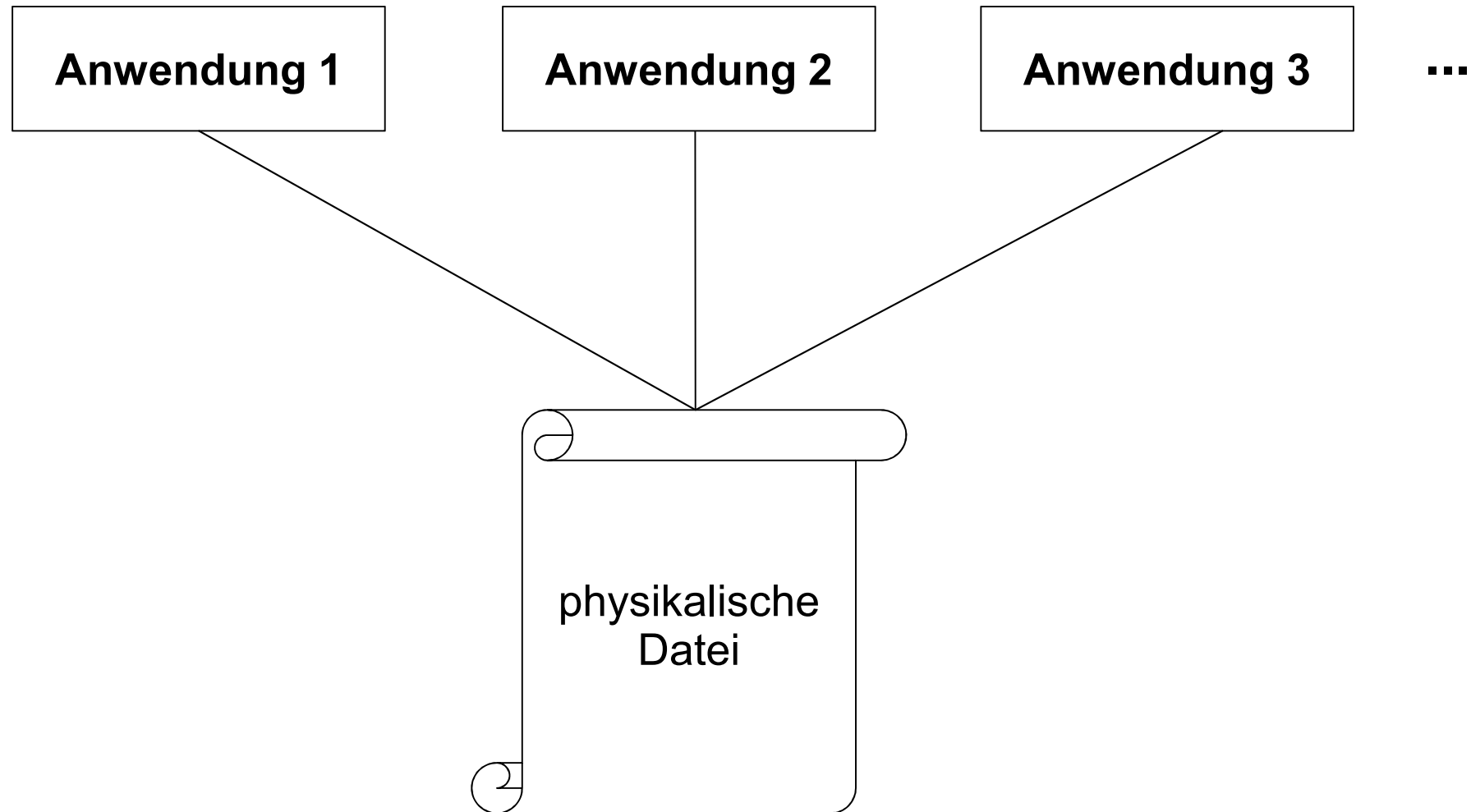
# Daten, Daten, Daten, ...



# Speichern im Dateisystemen



# Speichern im Dateisystem



# Grenzen der Datenverwaltung in Dateien

- **Datenredundanz**

Daten werden mehrfach gespeichert, größerer Speicherverbrauch, evtl. Missdeutung der Daten usw..

- **Verletzung der Datenintegrität**

Datensätze die nicht fehlen dürfen, werden versehentlich gelöscht. Daten, die einen bestimmten Wert nicht annehmen dürfen, bekommen diese zugewiesen..

- **Mehrbenutzerprobleme**

Gleichzeitiges Editieren derselben Datei führt zu Anomalien (lost update).

- **Verlust von Daten**

Versehentliche Änderungen der Daten sind nicht so einfach wieder rückgängig zu machen

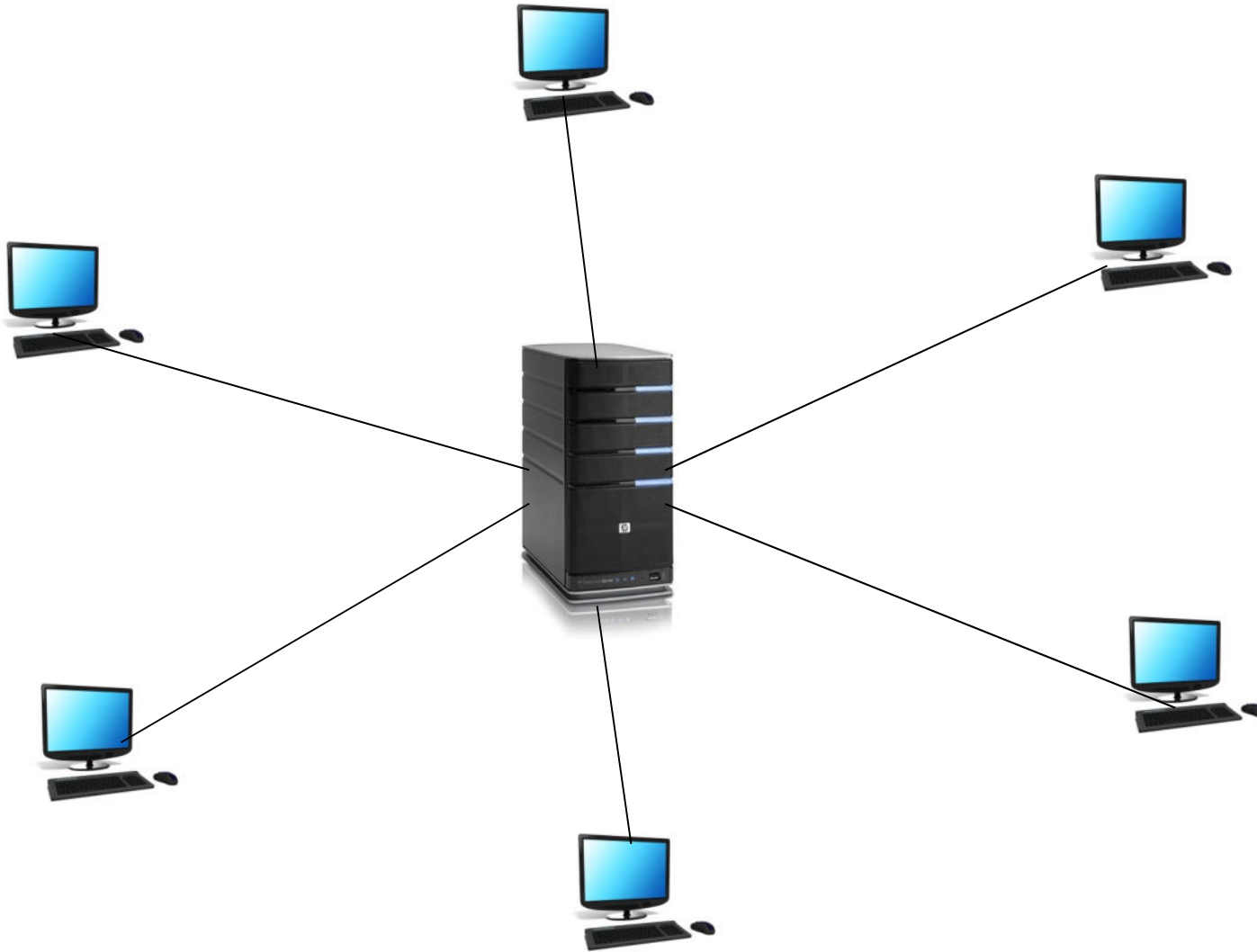
- **Sicherheitsprobleme**

Zugriffsrechte auf nur auf Dateien, nicht auf Datensätze..

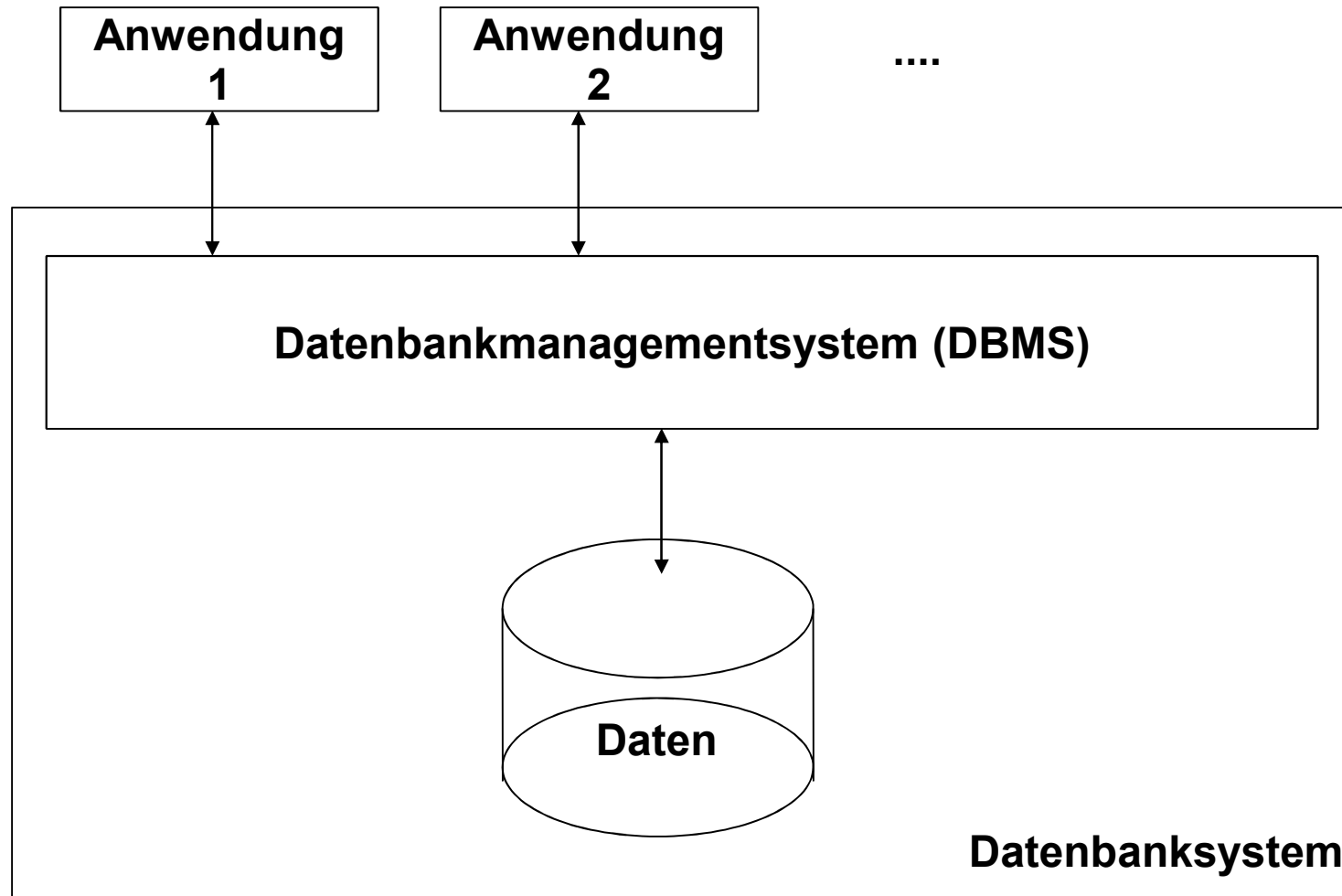
- **Keine (optimierte) Anfragesprache**

Oft ist es notwendig, sehr flexibel und einfach, Daten aus unterschiedlichsten Gesichtspunkten abzurufen.

# Architektur eines Datenbanksystems Server-Client



# Architektur eines Datenbanksystems



# Vorteile der Datenverwaltung in Datenbanken

- **Keine Datenredundanz**

Daten werden nur einmal gespeichert. Dafür ist ein guter Datenbankentwurf vom Entwickler notwendig.

- **Datenintegrität**

Unerlaubte Veränderungen der Daten durch den Benutzer werden durch das DBMS verhindert.

- **Auflösung von Mehrbenutzerprobleme: Transaktionen**

Gleichzeitiges Verändern von Daten wird verhindert. Änderungen an den Daten werden nicht sofort ausgeführt (Commit).

- **Aktionen rückgängig machen**

Einzelnen Aktionen können wieder rückgängig gemacht werden (Rollback).

- **Rechtevergabe**

Anlegen von Benutzer mit detaillierten Rechten.

- **SQL als Anfragesprache**

Strukturierte, standardisierte Sprache zum Auffinden und Bearbeiten von Datensätzen.



# ACID-Eigenschaften von DBMS

- **Atomicity:** Ein Datenzugriff (**Transaktion**) ist (logisch) eine nicht weiter zerlegbare Einheit; die dadurch bewirkten Änderungen in der Datenbank werden entweder vollständig oder gar nicht vorgenommen (**Alles-oder-nichts-Eigenschaft**).
- **Consistency:** Der Datenbestand bleibt in einem gültigen Zustand (konsistent). Änderungen in der Datenbank sind stets konform mit den vorher definierten **Integritätsbedingungen**!
- **Isolation:** Transaktionen laufen ungestört von anderen Transaktionen ab. (Es können nicht die selben Daten von verschiedenen Transaktionen bearbeitet werden). **Einbenutzerbetrieb**.
- **Durability:** Wenn eine Transaktion erfolgreich ihr Ende erreicht hat (commit), dann sind alle von ihr verursachten Änderungen dauerhaft (**persistent**).

# Bekannte DBMS

Einige bekannte **Datenbankmanagementsysteme (DBMS)**:

- **DB2** - das kommerzielle DBMS der Firma IBM
- **Microsoft Access** - das kommerzielle DBMS von Microsoft für PCs
- **Microsoft SQL Server** - das kommerzielle DBMS von Microsoft für große Anwendungen
- **MySQL** - das kostenfreie Open-Source-DBMS von Sun (heute Oracle) vielfach auf Internet-Servern eingesetzt
- **Oracle Database** - das kommerzielle DBMS der Firma Oracle
- **PostgreSQL** - ein kostenfreies Open-Source-DBMS

[http://de.wikipedia.org/wiki/Liste\\_der\\_Datenbankmanagementsysteme](http://de.wikipedia.org/wiki/Liste_der_Datenbankmanagementsysteme)



# ***Relationelle Datenbanken***

# Relationen-Modell

<u>Code</u>	Name	Hauptstadt	Fläche
AT	Österreich	Wien	84
ET	Ägypten	Kairo	1001
DE	Deutschland	Berlin	357

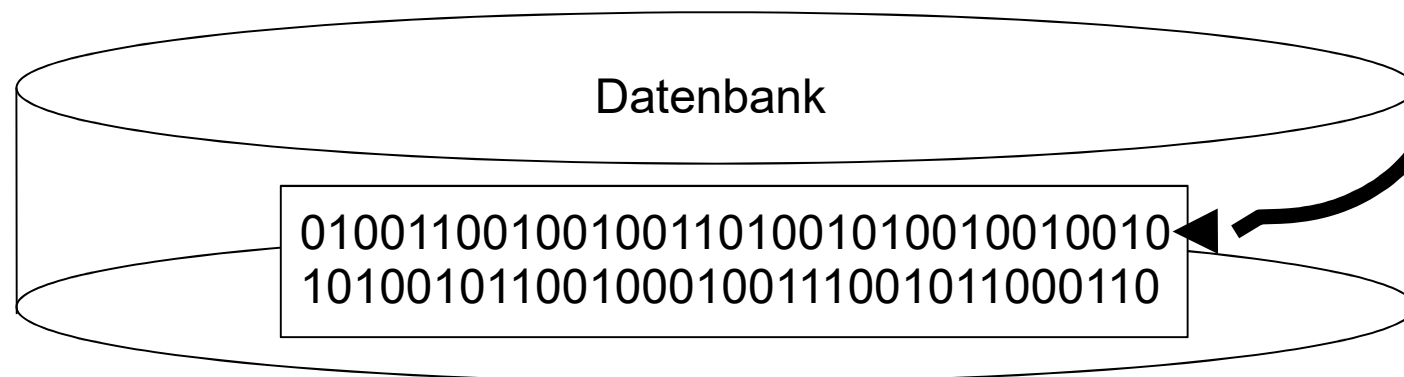
- In **relationalen Datenbanken** werden die zu speichernden Daten in Form von **Tabellen** (Relationen) **logisch** strukturiert.
- **Weitere Beispiele für logische Datenmodelle:**
  - Netzwerkmodell
  - Hierarchische Datenmodell
  - Objektorientierte Datenmodell
- Am weitesten verbreitet, mit etwa 80% Marktanteil, ist aber das Relationen-Modell!

# Logisches und physisches Datenmodell

logisches  
Datenmodell

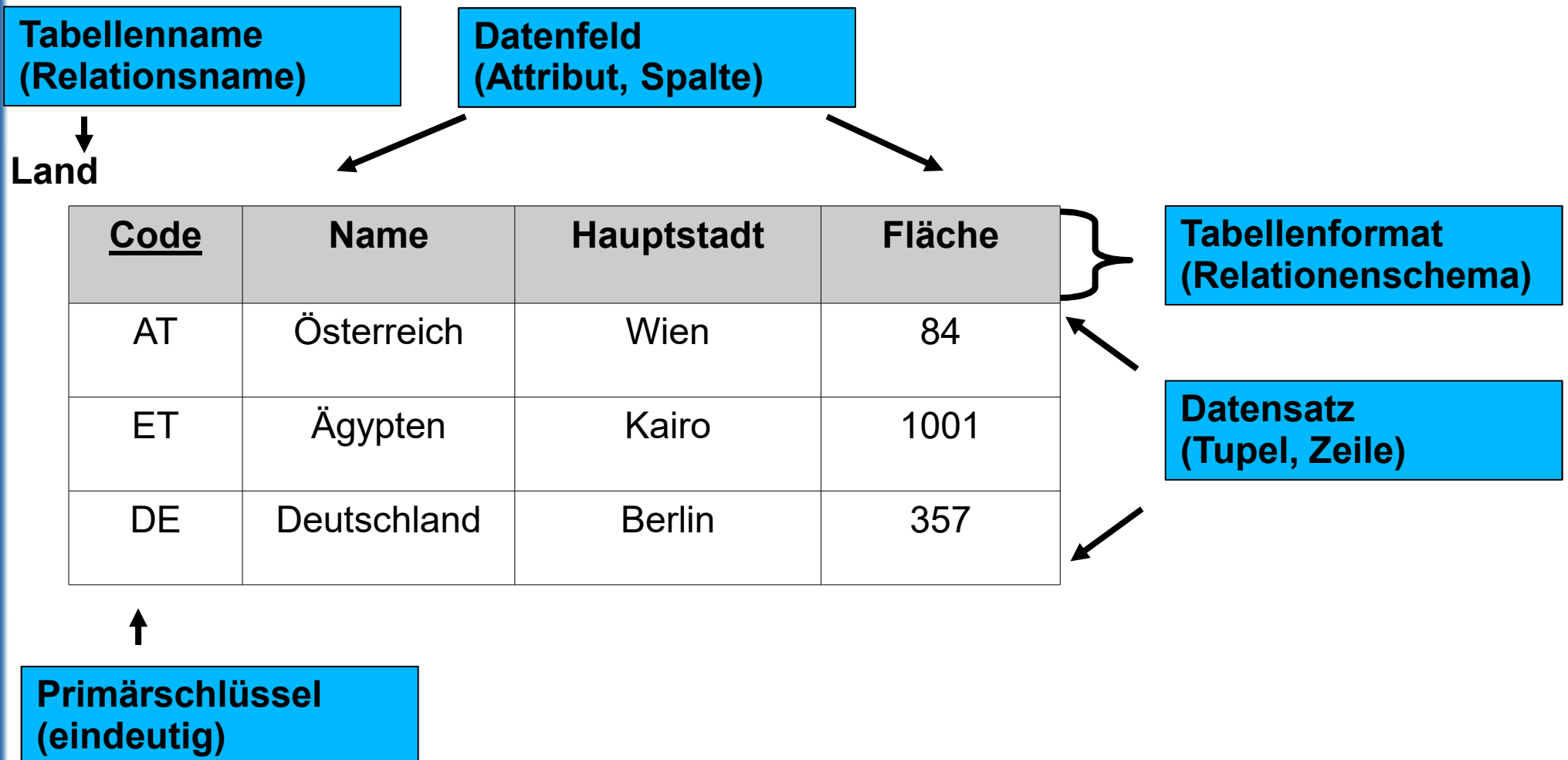
<u>Code</u>	Name	Hauptstadt	Fläche
AT	Österreich	Wien	84
ET	Ägypten	Kairo	1001
DE	Deutschland	Berlin	357

physisches  
Datenmodell



# Begriffe

Eine relationale Datenbank besitzt eine oder mehrere Tabellen



# Übung 1



In der Datenbank sollen zusätzlich zu den Informationen der Tabelle **Land** noch die unten stehenden Daten abgespeichert werden:

- Deutschland hat eine Fläche von 357T qkm mit Hauptstadt Berlin. Italien ist 301T qkm groß mit Rom als Hauptstadt.
- Die Provinz Baden ist eine deutsche Provinz mit der Fläche 15T qkm.
- Die Provinz Lazio ist eine italienische Provinz mit der Fläche 17T qkm.
- Die Stadt Freiburg ist eine deutsche Stadt aus der Provinz Baden. Sie hat 229T Einwohner und liegt auf dem Breitengrad 47,59 und dem Längengrad 7,51.
- Die Stadt Berlin ist eine deutsche Stadt aus der Provinz Berlin mit einer Fläche von 0,9T qkm. Sie hat 3472T Einwohner und liegt auf dem Breitengrad 52,45 und dem Längengrad 13,2.
- Die Stadt Rom ist eine italienische Stadt, die in der Provinz Lazio liegt, 2800T Einwohner hat und auf dem Breitengrad 41,8 und dem Längengrad 12,6 liegt.

# Übung 1

## Eine Lösungsmöglichkeit

Code	Name	Hstadt	Fläche	Provinz	PFläche	Stadt	SEinwohner	LG	BG
DE	Deutschland	Berlin	357	Baden	15	FR	229	7,51	47,6
DE	Deutschland	Berlin	357	Berlin	0,9	Berlin	3472	52,4	13,2
IT	Italien	Rom	301	Lazio	17	Rom	2800	41,8	12,6

Viele Informationen werden mehrfach (**redundant**) abgespeichert.

### Nachteile:

- erhöhter Speicherbedarf
- Schwierigkeiten bei Änderungen
- Beim Löschen einer Stadt oder Provinz gehen evt. auch Informationen über das jeweilige Land verloren.



# Übung 1

## Zweite Lösungsmöglichkeit

Land

<u>Code</u>	Name	Fläche	Hstadt
DE	Deutschland	357	Berlin
IT	Italien	301	Rom

Provinz

<u>Provinz</u>	PFläche	Land
Baden	15	DE
Berlin	0,9	DE
Lazio	17	IT

Stadt

Provinz	<u>Stadt</u>	SEinwohner	LG	BG
Baden	FR	229	7,51	47,6
Berlin	Berlin	3472	52,4	13,2
Lazio	Rom	2800	41,8	12,6

# Primärschlüssel

<u>Code</u>	Name	Hauptstadt	Fläche
AT	Österreich	Wien	84
ET	Ägypten	Kairo	1001
DE	Deutschland	Berlin	357

Der Primärschlüssel (primary key) ist eine **Spalte** in der Tabelle. Sie legt jeden **Datensatz eindeutig** fest.

- Der Primärschlüssel darf **nicht leer** sein und muss **eindeutig** sein, d.h. es gibt keine doppelten Werte (Constraint).
- Im Relationenmodell wird der Primärschlüssel **unterstrichen**.
- Der Primärschlüssel kann auch aus **mehreren Spalten** bestehen.

# Fremdschlüssel und Referentielle Integrität

Land

<u>Code</u>	Name	Hauptstadt	Fläche
AT	Österreich	1	84
ET	Ägypten	2	1001
DE	Deutschland	3	357

Stadt

<u>Id</u>	Name	Einwohner
1	Wien	1793
2	Kairo	7947
3	Berlin	3472

Ein Fremdschlüssel (foreign key) ist eine **Spalte** in der Tabelle, die **Werte von Primärschlüsseln** anderer Tabellen besitzt. Dadurch wird eine Verbindung zwischen Tabellen hergestellt.

Referentielle Integrität (foreign key constraint).

- Fremdschlüssel haben nur solche **Werte**, die in **Primärschlüsseln** anderer Tabellen vorkommen.
- Ein **Primärschlüssel** kann **nicht gelöscht werden**, falls er als Fremdschlüssel in einer anderen Tabelle noch auftaucht.

# Quellen

- Ahmad Nessar Nazar: Unterrichtsunterlagen
- Michael Dienert: Unterrichtsunterlagen
- Wikipedia