Datenbanksysteme
1 - Motivation und Grundlagen von Datenbanken

Prof. Dr. Katja Zeume

katja.zeume@w-hs.de

WHS Gelsenkirchen

Inhalt

In dieser Vorlesung

- Motivation: Warum Datenbanken?
- Historische Entwicklung Datenbanken und Datenbankmanagementsysteme (DBMS)
- Grundlegende Merkmale von Datenbanken
- Aufgaben eines DBMS nach Codd
- DBMS im Software Stack

Motivation Datenbanksysteme

Wir haben schon kurz gesehen was Datenbanken sind und wofür diese verwendet werden, aber ...

warum brauchen wir Datenbanken?

Beispiel

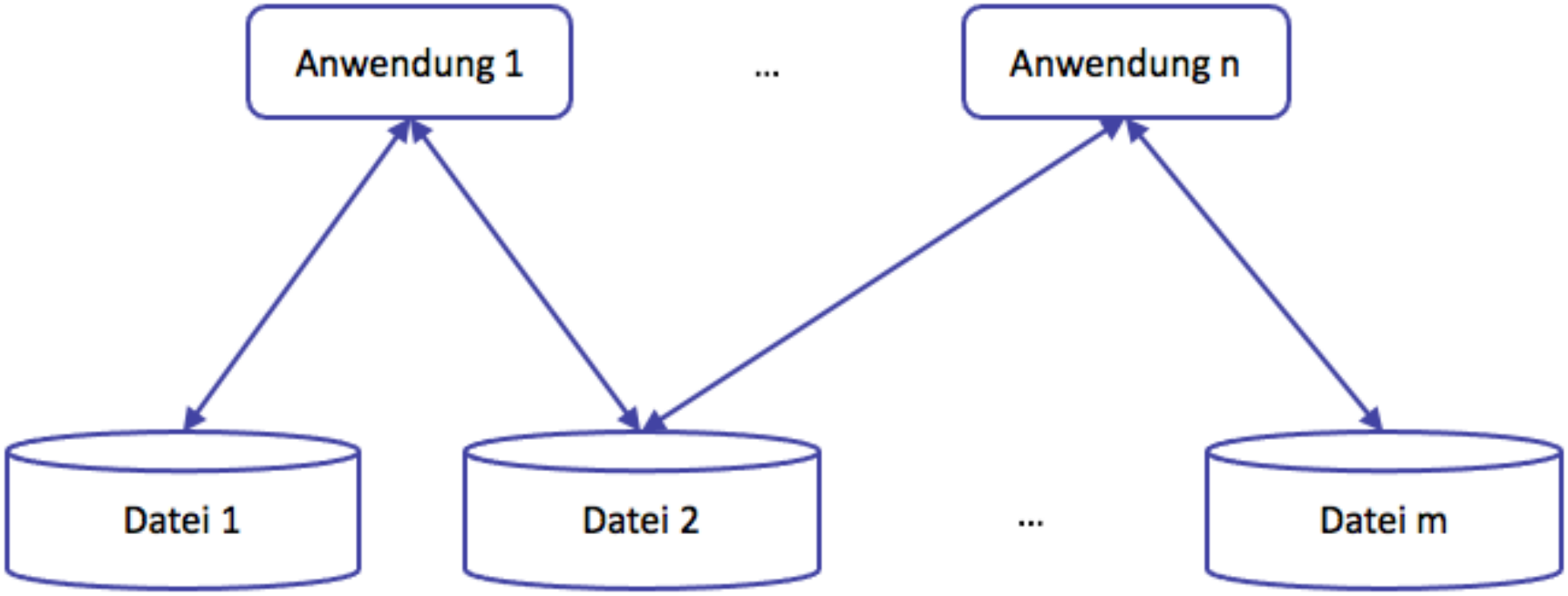
Aktuelle Filme (23/24):

Film-ID	Titel	Erscheinungsdatum	Genre
1	Wonka	2023-12-07	Family
2	Dune:Part Two	2024-03-02	Action
3	Barbie	2023-07-20	Comedy
4	Oppenheimer	2023-07-20	Thriller
5	John Wick 4	2023-03-23	Thriller

Diese Daten werden in der Datei `filme_2324.txt` gespeichert.

Was sind **Vor- und Nachteile** von diesem Vorgehen?
Welche Probleme können auftreten? Und wie kann man diese lösen?

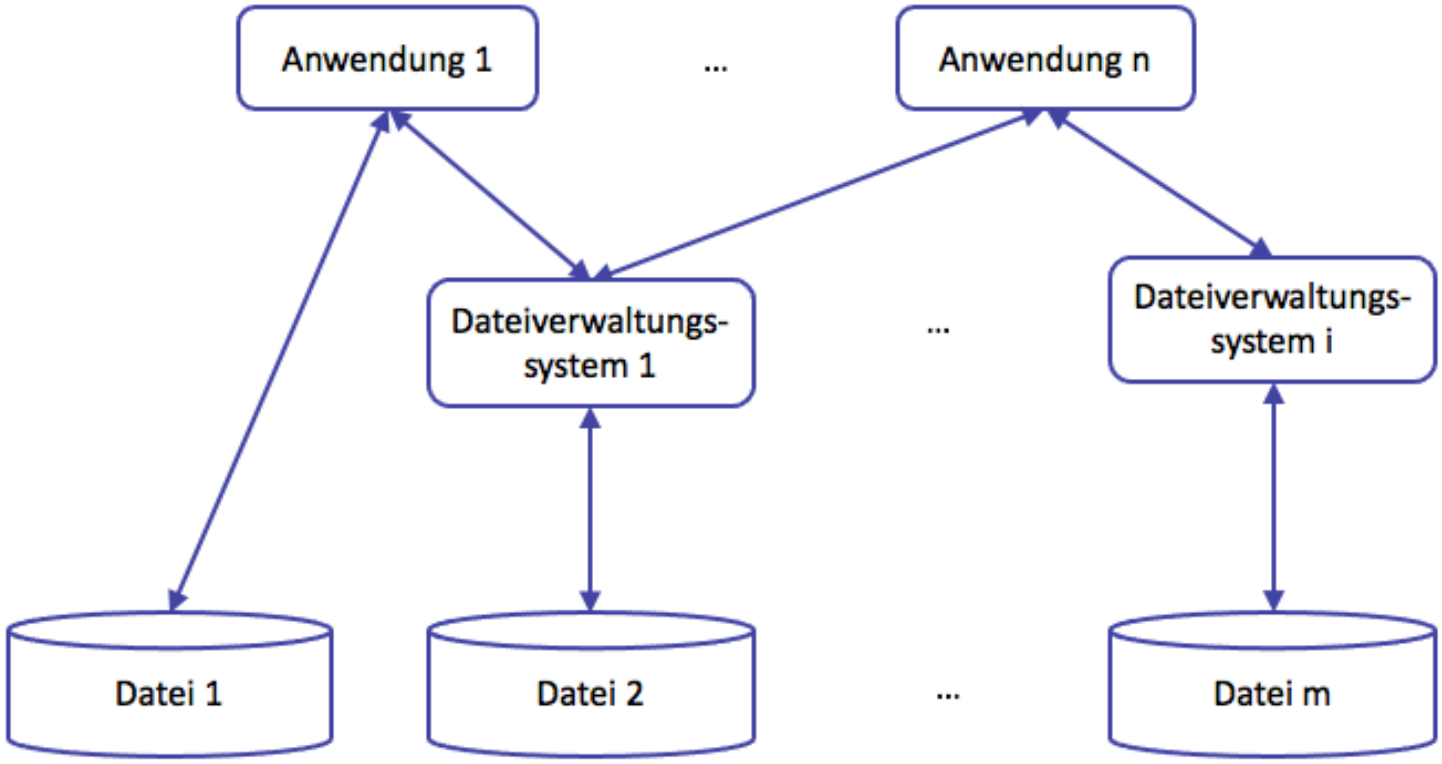
Historische Entwicklung von Datenbanken



[Quelle Saake et al.]

- Beginn der 60er Jahre
- Daten in einzelnen anwendungsspezifischen Dateien organisiert
- geräteabhängig
- Probleme mit Redundanz und Inkonsistenz

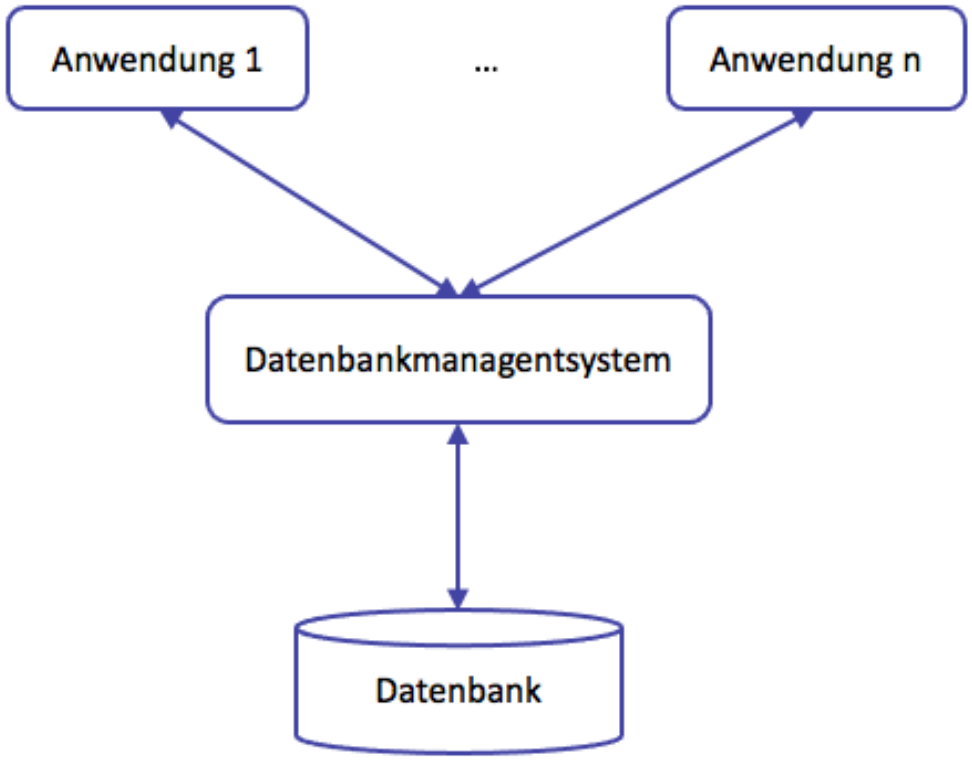
Historische Entwicklung von Datenbanken



[Quelle Saake et al.]

- Ende der 60er Jahre
- Daten in Dateiverwaltungssystemen
- *geräteunabhängig*
- Probleme mit Redundanz und Inkonsistenz bleiben

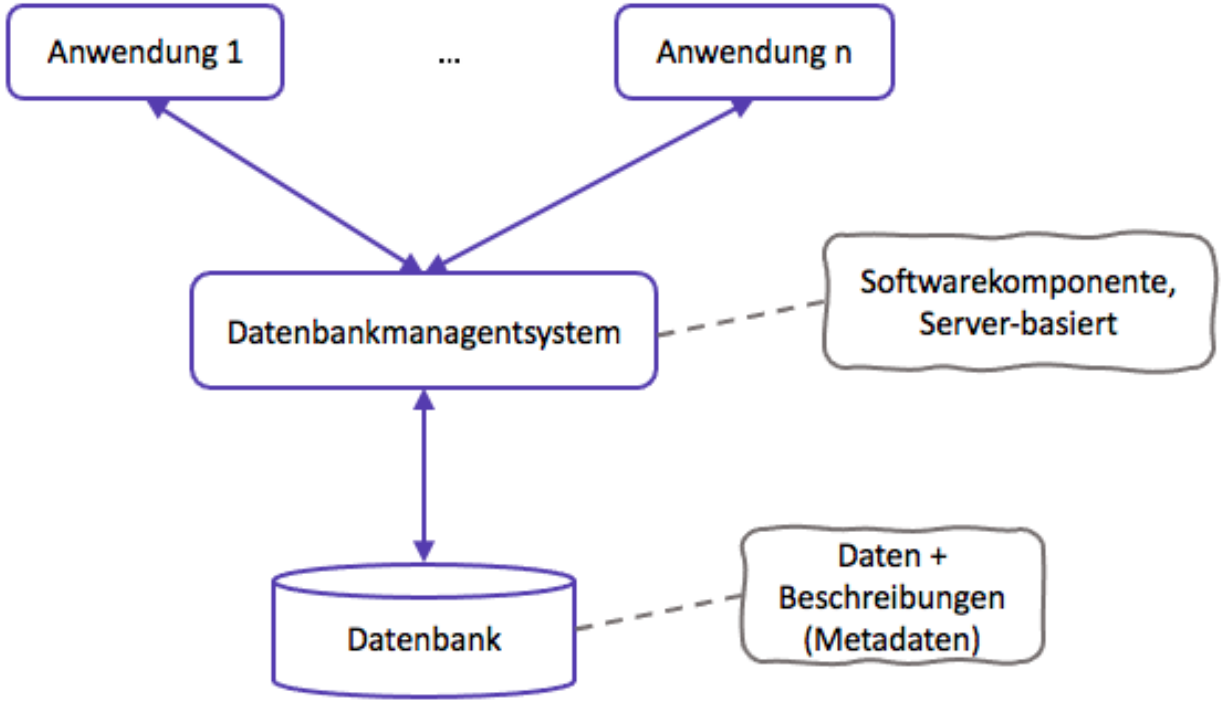
Historische Entwicklung von Datenbanken



[Quelle Saake et al.]

- Ab 70er Jahre
- Einsatz Datenbanksysteme
- geräte- und datenunabhängig
- Redundanzfreie und konsistente Datenhaltung möglich
- Codd's Relationenmodell

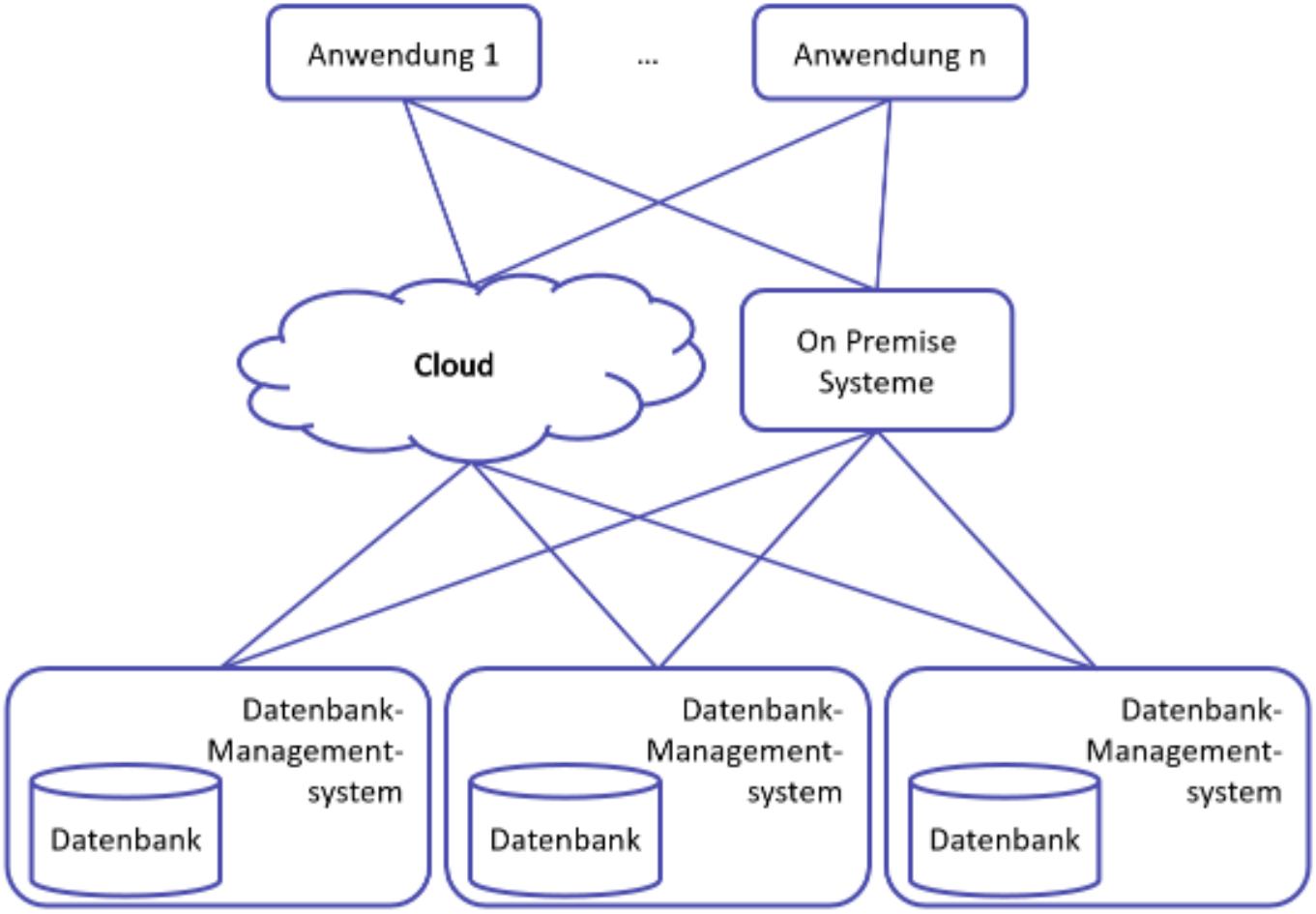
Historische Entwicklung von Datenbanken



[Quelle Saake et al.]

- Ab 70er Jahre
- Einsatz Datenbanksysteme
- geräte- und datenunabhängig
- Redundanzfreie und konsistente Datenhaltung möglich
- Codd's Relationenmodell

Historische Entwicklung von Datenbanken



- Ab 00er Jahre
- NOSQL Datenbanken
- Cloud Computing
- Verteilte Datenbanken
- ACID vs. CAP Theorem
- Heterogene Architekturen

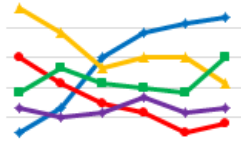
Datenbanksysteme Beispiele

`https://db-engines.com/de/ranking`

DB-Engines Ranking

Das DB-Engines Ranking ist eine Rangliste der Popularität von Datenbankmanagementsystemen. Die Rangliste wird monatlich aktualisiert.

Lesen Sie mehr über die Berechnungsmethode der Bewertungen.



Trend-Diagramm

425 Systeme im Ranking, Oktober 2025

Rang			DBMS	Datenbankmodell	Punkte		
Okt 2025	Sep 2025	Okt 2024			Okt 2025	Sep 2025	Okt 2024
1.	1.	1.	Oracle	Relational, Multi-Model	1212,77	+42,15	-96,67
2.	2.	2.	MySQL	Relational, Multi-Model	879,66	-12,11	-143,09
3.	3.	3.	Microsoft SQL Server	Relational, Multi-Model	715,05	-2,27	-87,04
4.	4.	4.	PostgreSQL	Relational, Multi-Model	643,20	-13,97	-8,96
5.	5.	5.	MongoDB	Document, Multi-Model	368,01	-12,49	-37,20
6.	6.	↑ 7.	Snowflake	Relational	198,65	+8,46	+58,05
7.	7.	↓ 6.	Redis	Key-value, Multi-Model	142,33	-2,84	-7,30
8.	↑ 9.	↑ 14.	Databricks	Multi-Model	128,80	+4,74	+43,21
9.	↓ 8.	9.	IBM Db2	Relational, Multi-Model	122,37	-1,81	-0,40
10.	10.	↓ 8.	Elasticsearch	Multi-Model	116,67	-1,59	-15,18
11.	↑ 12.	11.	Apache Cassandra	Wide column, Multi-Model	105,16	-1,82	+7,56
12.	↓ 11.	↓ 10.	SQLite	Relational	104,56	-3,32	+2,64
13.	13.	↑ 15.	MariaDB	Relational, Multi-Model	87,77	-3,69	+2,88
14.	14.	↓ 12.	Microsoft Access	Relational	80,79	-2,82	-11,36
15.	15.	↑ 17.	Amazon DynamoDB	Multi-Model	75,91	-4,37	+4,06

Grundlegende Begriffe

Grundlegende Begriffe von Datenbanksystemen

- **Datenbank (DB)**
  Als Datenbank, oder kurz DB, wird der strukturierte Datenbestand oder Datensatz bezeichnet.
- **Datenbankmanagementsystem (DBMS)**
  Die Software zur Verwaltung von Datenbanken wird Datenbankmanagementsystem, oder kurz DBMS, genannt.
- **Datenbanksystem (DBS)**
  Ein Datenbanksystem, oder kurz DBS, besteht aus DBMS und DB zusammen.

$\rightarrow$ Im Folgenden werden wir uns genauer anschauen welche Eigenschaften wir uns von einem Datenbanksystem wünschen bzw. welche Aufgaben es erfüllen muss.

Grundlegende Eigenschaften von Datenbanken

Eigenschaften von Datenbanksystemen

1. Welche Daten werden gespeichert? Wie?
2. Wie werden Daten abgefragt?
3. Wie wird Effizienz und Sicherheit garantiert?

Grundlegende Eigenschaften von Datenbanken

Eigenschaften von Datenbanksystemen

1. Welche Daten werden gespeichert? Wie?
   Ein **logisches Datenmodell** mit und **Datenschema** wird genutzt.
2. Wie werden Daten abgefragt?

3. Wie wird Effizienz und Sicherheit garantiert?

Grundlegende Eigenschaften von Datenbanken

Eigenschaften von Datenbanksystemen

1. Welche Daten werden gespeichert? Wie?
   Ein **logisches Datenmodell** mit und **Datenschema** wird genutzt.
2. Wie werden Daten abgefragt?
   Es stehen **Operationen zur Datenmanipulation** zur Verfügung.
   Diese werden oft auch als Sprache (Datendefinitionssprache, Anfragesprache etc.) bezeichnet.
3. Wie wird Effizienz und Sicherheit garantiert?

Grundlegende Eigenschaften von Datenbanken

Eigenschaften von Datenbanksystemen

1. Welche Daten werden gespeichert? Wie?
   Ein **logisches Datenmodell** mit und **Datenschema** wird genutzt.
2. Wie werden Daten abgefragt?
   Es stehen **Operationen zur Datenmanipulation** zur Verfügung.
   Diese werden oft auch als Sprache (Datendefinitionssprache, Anfragesprache etc.) bezeichnet.
3. Wie wird Effizienz und Sicherheit garantiert?
   Die Daten werden **persistent**, **konsistent** und **effizient** verwaltet und gegen unerwünschte Zugriffe geschützt.
   Es wird ein **Transaktionskonzept** unterstützt. Logisch zusammengehörende Transaktionen werden zusammen (atomar/unteilbar) ausgeführt. Parallele Anfragen von mehreren Benutzern sind möglich.

9 Aufgaben nach Codd

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden schon 1982 von Codd formuliert.

1. Datenintegration
2. Operationen und Anfragesprachen
3. Datenkatalog
4. Benutzersichten
5. Konsistenz
6. Zugriffskontrolle
7. Transaktionskonzept
8. Synchronisation
9. Datensicherung

Im folgenden werden diese im Detail erläutert.

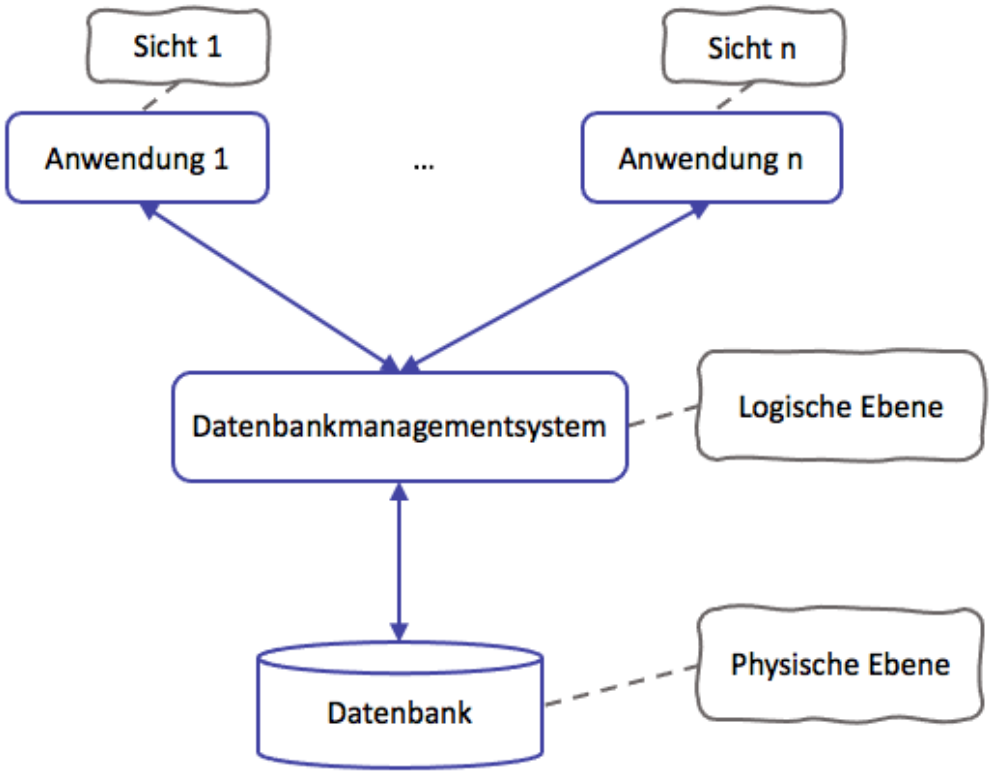
9 Aufgaben nach Codd

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden schon 1982 von Codd formuliert.

1. **Datenintegration**
2. Operationen und Anfragesprachen
3. Datenkatalog
4. Benutzersichten
5. Konsistenz
6. Zugriffskontrolle
7. Transaktionskonzept
8. Synchronisation
9. Datensicherung

Im folgenden werden diese im Detail erläutert.

Datenintegration



Als **Datenintegration** bezeichnet man die einheitliche Datenhaltung der Daten in der Datenbank (oder mehreren Datenbanken).

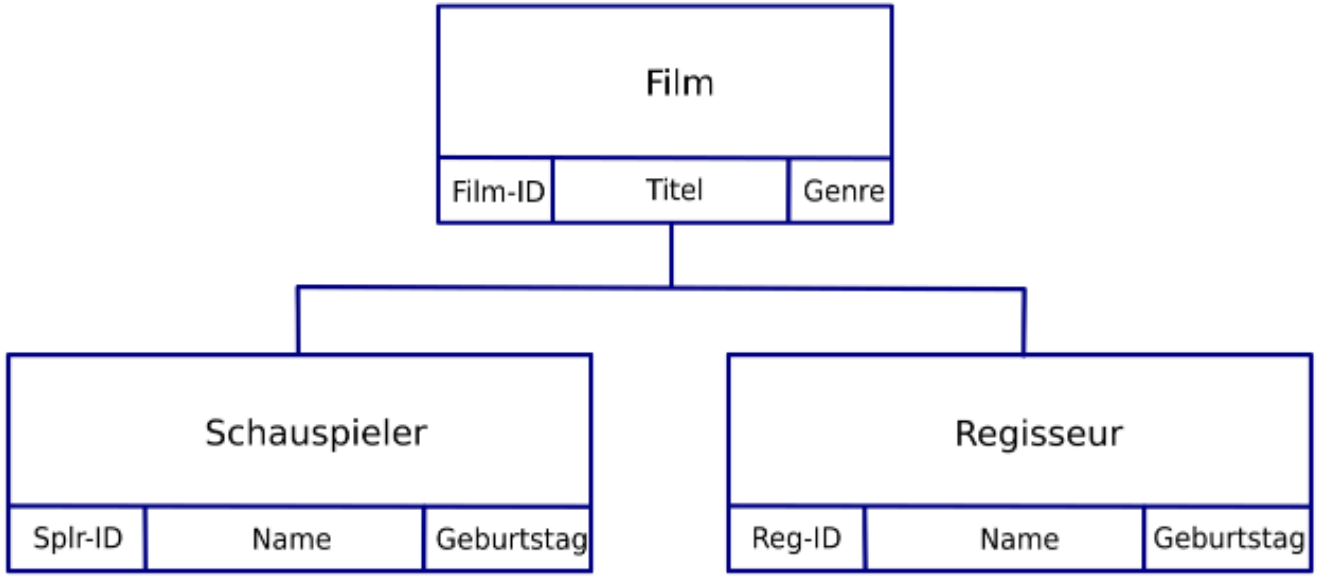
Ein **logisches Datenmodell** wird benutzt um den Inhalt der Datenintegration für verschiedene Anwendungen zu beschreiben und zu kommunizieren.

Datenintegration - Logische Datenmodelle

Es gibt verschiedene Formate logische Datenmodelle zu beschreiben:

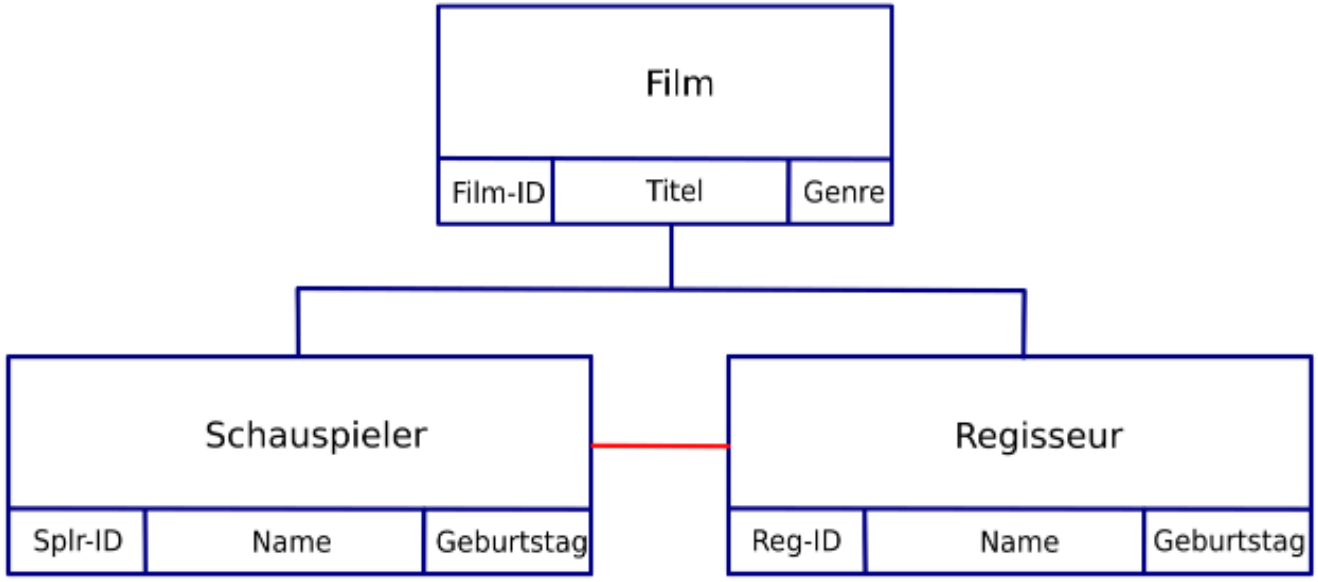
- Hierarchisches Modell (∼1960er)
- Netzwerkmodell (∼1970er)
- Relationales Modell (∼1970er)
- XML Modell (∼1990er)
- Objektorientiertes Modell (∼1990er)
- „Neue“ Datenmodelle (∼2000er)

Hierarchisches Modell



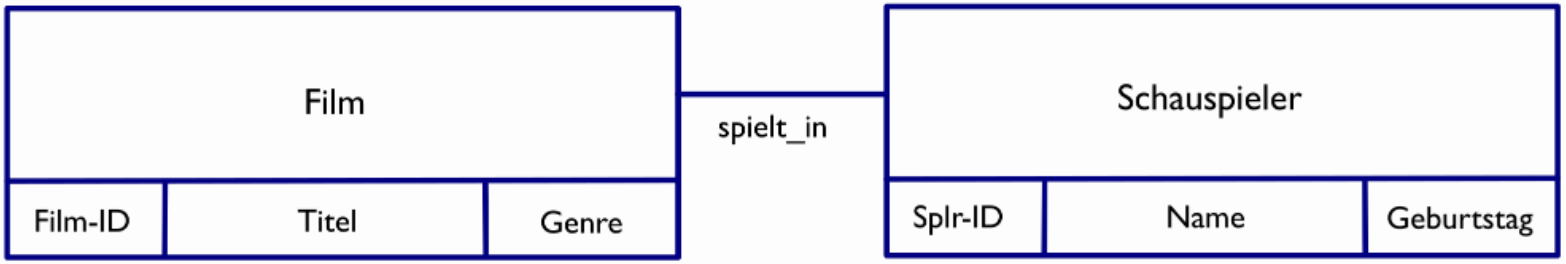
- Reale Welt durch Baumstruktur abgebildet.
- Ausprägungen sind z. Bsp. die einzelnen Vorlesungen, wie (DBA, Datenbanksysteme, 5).
- Nachteil: Nur 1 : n-Beziehungen, komplexe Verknüpfungen zwischen Objekten nicht möglich.

Netzwerkmodell



- Keine strenge Hierarchie erforderlich
- Ungefähr zeitgleich mit Relationalen Modell
- Auch komplexe Verbindungen und $n : m$-Beziehungen erlaubt.

Relationales Modell



Film-ID	Titel	Genre
1	Wonka	Family
2	Dune:Part Two	Action
3	Barbie	Comedy

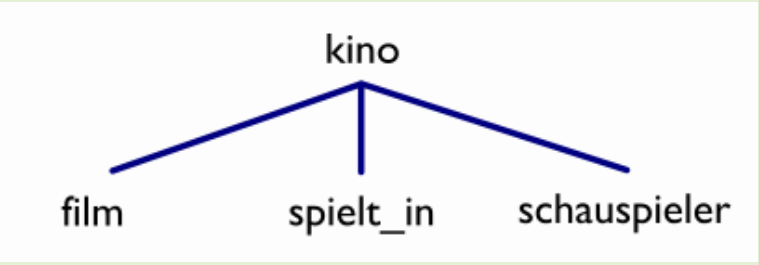
Film-ID	Splr-ID
2	1
1	2
2	2

Splr-ID	Name	Geburtstag
1	Zendaya Coleman	1996-09-01
2	Keanu Reeves	1964-09-02
3	Timothee Chalamet	1995-12-27

- Objekte und Beziehungen werden in Tabellen modelliert.
- Die Informationen in den Tabellen können mit Operationen ausgewertet werden.
- Das relationale Datenmodell wird in der nächsten Vorlesung genauer betrachtet.

XML Modell

```
<kino>
    <film>
        <film-id> 1 </film-id>
        <titel> Wonka </film-id>
        <genre> Family </genre>
    </film>
    <film>
        <film-id> 2 </film-id>
        <titel> Dune: Part Two </film-id>
        <genre> Action </genre>
    </film>
    <spielt_in>
        <film-id> 1 </film-id>
        <schauspieler-id> 4 </schauspieler-id>
    </spielt_in>
    <spielt_in>
        <film-id> 2 </film-id>
        <schauspieler-id> 4 </schauspieler-id>
    </spielt_in>
    <schauspieler>
        <schauspieler-id> 4 </schauspieler-id>
        <vorname> Timothée </vorname>
        <nachname> Chalamet </nachname>
    </schauspieler>
</kino>
```



- Objekte und Beziehungen werden in Baumstrukturen modelliert.
- Die Baumstruktur kann durch Schemata beschrieben werden.
- Eigene Anfragesprachen werten die enthaltenen Daten aus (z. Bsp. über die Auswertung von Pfaden in der Baumstruktur).

Objektorientiertes Modell

```
class Film {
        attribute int film-id;
        attribute String titel;
        attribute String genre;
        relationship set <Regisseur> hat_regie;
}

class Schauspieler {
        attribute int splr-id;
        attribute String name;
        attribute String geburtstag;
        relationship set <Film> spielt_in;
}

class Regisseur {
        attribute int reg-id;
        attribute String name;
        attribute String geburtstag;
        relationship set <Film> fuehrt_regie;
}
```

- Daten und Funktionen werden in einem Objekt gespeichert.
- Selten in der Praxis genutzt.
- Nachteile: Komplexität bei der Datenverarbeitung und komplexe Schnittstellen.

Datenintegration - Logische Datenmodelle

Es gibt verschiedene Formate logische Datenmodelle zu beschreiben:

- Hierarchisches Modell (∼1960er)
- Netzwerkmodell (∼1970er)
- Relationales Modell (∼1970er)
- XML Modell (∼1990er)
- Objektorientiertes Modell (∼1990er)
- „Neue“ Datenmodelle (∼2000er)

Datenintegration - Logische Datenmodelle

Es gibt verschiedene Formate logische Datenmodelle zu beschreiben:

- Hierarchisches Modell (∼1960er)
- Netzwerkmodell (∼1970er)
- **Relationales Modell** (∼1970er)
- XML Modell (∼1990er)
- Objektorientiertes Modell (∼1990er)
- „Neue“ Datenmodelle (∼2000er)

Hier in der Vorlesung werden wir das **Relationale Modell** genauer betrachten um unsere Daten zu beschreiben.
Neuere Datenmodelle (rund um das Thema NOSQL) sind Thema der weiterführenden Datenbankvorlesungen.

Aufgaben

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden allerdings schon 1982 von Codd formuliert.

1. **Datenintegration**
2. Operationen und Anfragesprachen
3. Datenkatalog
4. Benutzersichten
5. Konsistenz
6. Zugriffskontrolle
7. Transaktionskonzept
8. Synchronisation
9. Datensicherung

Im folgenden werden diese im Detail erläutert.

Aufgaben

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden allerdings schon 1982 von Codd formuliert.

1. Datenintegration
2. **Operationen und Anfragesprachen**
3. Datenkatalog
4. Benutzersichten
5. Konsistenz
6. Zugriffskontrolle
7. Transaktionskonzept
8. Synchronisation
9. Datensicherung

Im folgenden werden diese im Detail erläutert.

Operationen und Anfragesprachen

. . . werden zur Datenverarbeitung benötigt (Speichern, Ändern, Suchen) und müssen auf realen Daten (>1000 Tabellen mit >1Mrd. Zeilen) anwendbar sein.

Der de-facto Standard für relationale Datenbanken ist **SQL**.

Structured Query Language (SQL)

- Entwickelt in den 70er Jahren von Edgar F. Codd, Donald D. Chamberlin und Raymond F. Boyce (Nachfolger von SEQUEL)
- Basiert auf der *relationalen Algebra*
- Ursprünglich an englischer Sprache angelehnt und entwickelt als End-User Sprache

SQL - Hello World

Eine der wesentlichsten Anweisungen in SQL ist die SELECT-FROM-WHERE-Klausel:

```
SELECT Titel, Genre
FROM Film
WHERE Dauer > '150'
```

`SELECT` von einer oder mehreren Spalten einer Tabelle

`FROM` eine (kombinierte) Tabelle von der die Daten gesammelt werden

`WHERE` optionale Bedingungen, die die gesuchten Daten filtern

SQL - Hello World

Tabelle: Student

Film-ID	Titel	Erscheinungsdatum	Genre	Dauer
1	Wonka	2023-12-07	Family	117
2	Dune:Part Two	2024-03-02	Action	166
3	Barbie	2023-07-20	Comedy	114
4	Oppenheimer	2023-07-20	Thriller	180
5	John Wick 4	2023-03-23	Thriller	169

Anfrage:

```
SELECT Titel, Genre
FROM Film
WHERE Dauer > 150
```

Ergebnis:

Titel	Genre
Dune:Part Two	Action
Oppenheimer	Thriller
John Wick 4	Thriller

Aufgaben

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden allerdings schon 1982 von Codd formuliert.

1. Datenintegration
2. **Operationen und Anfragesprachen**
3. Datenkatalog
4. Benutzersichten
5. Konsistenz
6. Zugriffskontrolle
7. Transaktionskonzept
8. Synchronisation
9. Datensicherung

Im folgenden werden diese im Detail erläutert.

Aufgaben

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden allerdings schon 1982 von Codd formuliert.

1. Datenintegration
2. Operationen und Anfragesprachen
3. **Datenkatalog**
4. Benutzersichten
5. Konsistenz
6. Zugriffskontrolle
7. Transaktionskonzept
8. Synchronisation
9. Datensicherung

Im folgenden werden diese im Detail erläutert.

Datenkatalog

Das Datenbanksystem benutzt einen sogenannten **Datenkatalog** um die Definitionen der Anwendungsdaten, sowie deren Beziehungen untereinander beschreiben.

- Wird auch *Data Dictionary* genannt.
- Enthält Metadaten und Beschreibungen der Daten in der Datenbank und hängt somit stark mit dem logischen Datenmodell zusammen.
- Definitiert durch eine *Data Definition Language (DDL)*
- Üblicherweise *Read-Only* (Änderungen selten und teuer)

Aufgaben

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden allerdings schon 1982 von Codd formuliert.

1. Datenintegration
2. Operationen und Anfragesprachen
3. **Datenkatalog**
4. Benutzersichten
5. Konsistenz
6. Zugriffskontrolle
7. Transaktionskonzept
8. Synchronisation
9. Datensicherung

Im folgenden werden diese im Detail erläutert.

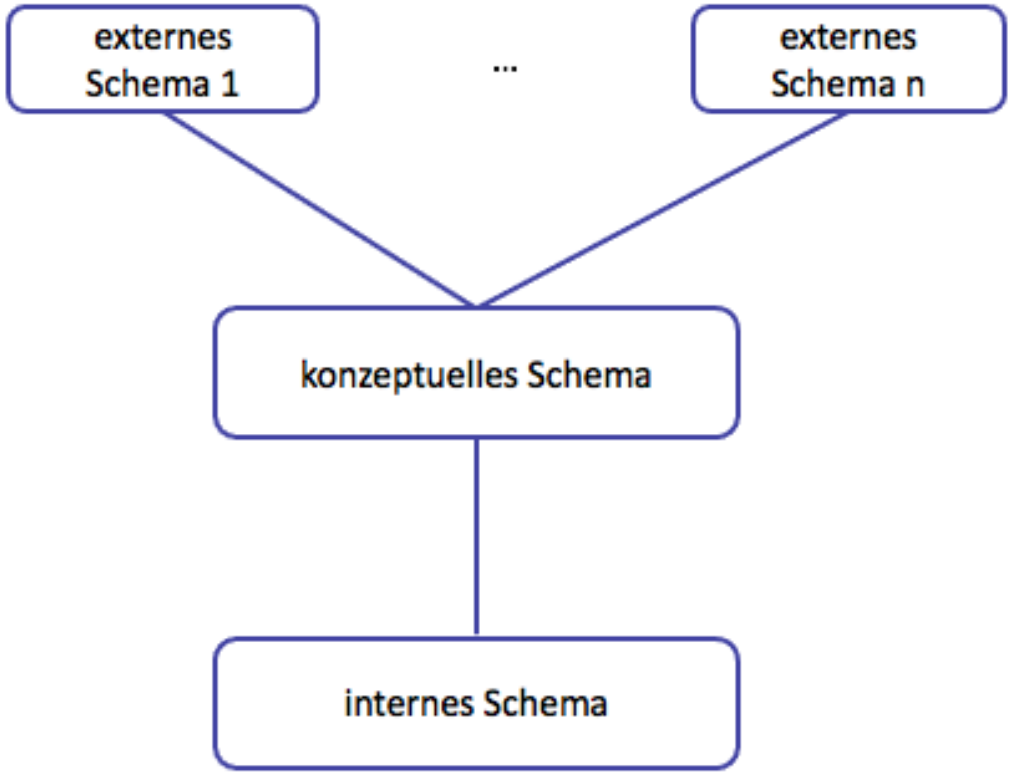
Aufgaben

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden allerdings schon 1982 von Codd formuliert.

1. Datenintegration
2. Operationen und Anfragesprachen
3. Datenkatalog
4. **Benutzersichten**
5. Konsistenz
6. Zugriffskontrolle
7. Transaktionskonzept
8. Synchronisation
9. Datensicherung

Im folgenden werden diese im Detail erläutert.

Benutzersichten



[Quelle Saake et al.]

- mehrere Tabellen werden zu einer Sicht (*virtuellen* Tabelle) kombiniert
- Daten werden gefiltert
- auch *Views* genannt
- üblicherweise anwendungsspezifisch

Aufgaben

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden allerdings schon 1982 von Codd formuliert.

1. Datenintegration
2. Operationen und Anfragesprachen
3. Datenkatalog
4. **Benutzersichten**
5. Konsistenz
6. Zugriffskontrolle
7. Transaktionskonzept
8. Synchronisation
9. Datensicherung

Im folgenden werden diese im Detail erläutert.

Aufgaben

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden allerdings schon 1982 von Codd formuliert.

1. Datenintegration
2. Operationen und Anfragesprachen
3. Datenkatalog
4. Benutzersichten
5. **Konsistenz**
6. **Zugriffskontrolle**
7. Transaktionskonzept
8. Synchronisation
9. Datensicherung

Im folgenden werden diese im Detail erläutert.

Konsistenz und Zugriffskontrolle

Konsistenz

Die Datenbank muss frei von Widersprüchen (**konsistent**) sein.
Inkonsistente Daten könnten entstehen

- wenn falsche Benutzereingaben gespeichert werden, oder
- Daten falsch gespeichert werden.

In beiden Fällen muss die Datenbank Kontrollstrukturen besitzen, die dies verhindern (siehe auch Transaktionskonzept).

Zugriffskontrolle

Im Datenbankbetrieb sollen

- unauthorisierte Zugriffe verhindert werden,
- Nutzer über geeignete Rollen nur auf die notwendigen Daten zugreifen können, und
- die Datenintegrität weiter gewährt werden.

Aufgaben

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden allerdings schon 1982 von Codd formuliert.

1. Datenintegration
2. Operationen und Anfragesprachen
3. Datenkatalog
4. Benutzersichten
5. **Konsistenz**
6. **Zugriffskontrolle**
7. Transaktionskonzept
8. Synchronisation
9. Datensicherung

Im folgenden werden diese im Detail erläutert.

Aufgaben

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden allerdings schon 1982 von Codd formuliert.

1. Datenintegration
2. Operationen und Anfragesprachen
3. Datenkatalog
4. Benutzersichten
5. Konsistenz
6. Zugriffskontrolle
7. **Transaktionskonzept**
8. Synchronisation
9. Datensicherung

Im folgenden werden diese im Detail erläutert.

Transaktionskonzept

Die Korrektheit und Integrität der Daten in der Datenbank muss immer gewährleistet werden. Dafür ist ein gutes **Transaktionskonzept** wichtig:

- Als **Transaktion** bezeichnet man eine Menge an Datenbankoperationen.
- Alle Transaktionen müssen eine gewisse Qualität besitzen. Einige Eigenschaften, die diese Qualität beschreiben, werden wir später mit dem sogenannten *ACID-Prinzip* kennenlernen.

ACID-Prinzip (kurz)

- **Atomaricity** Transaktionen werden "ganz oder gar nicht" ausgeführt.
- **Consistency** Transaktionen führen nur zu zulässigen Zuständen.
- **Isolation** Transaktionen werden isoliert für einen Nutzer ausgeführt.
- **Durability** Das Ergebniss der Transaktion ist persistent.

Aufgaben

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden allerdings schon 1982 von Codd formuliert.

1. Datenintegration
2. Operationen und Anfragesprachen
3. Datenkatalog
4. Benutzersichten
5. Konsistenz
6. Zugriffskontrolle
7. **Transaktionskonzept**
8. Synchronisation
9. Datensicherung

Im folgenden werden diese im Detail erläutert.

Aufgaben

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden allerdings schon 1982 von Codd formuliert.

1. Datenintegration
2. Operationen und Anfragesprachen
3. Datenkatalog
4. Benutzersichten
5. Konsistenz
6. Zugriffskontrolle
7. Transaktionskonzept
8. **Synchronisation**
9. **Datensicherung**

Im folgenden werden diese im Detail erläutert.

Synchronisation und Datensicherung

Synchronisation

Wenn mehrere Nutzer gleichzeitig auf eine Datenbank zugreifen, müssen Transaktionen, die sich gegenseitig beeinflussen, synchronisiert werden. Dafür müssen Schreibkonflikte vermieden werden und Inkonsistenzen ausgeschlossen werden.

Datensicherung

Die Daten in der Datenbank müssen gegen Verlust gesichert werden. Hier kann eine Reihe von Fehlern auftreten, die beachtet werden müssen: von lokalen Systemfehlern bis größeren Katastrophen (*disaster recovery*).

Aufgaben

Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden allerdings schon 1982 von Codd formuliert.

1. Datenintegration
2. Operationen und Anfragesprachen
3. Datenkatalog
4. Benutzersichten
5. Konsistenz
6. Zugriffskontrolle
7. Transaktionskonzept
8. **Synchronisation**
9. **Datensicherung**

Aufgaben

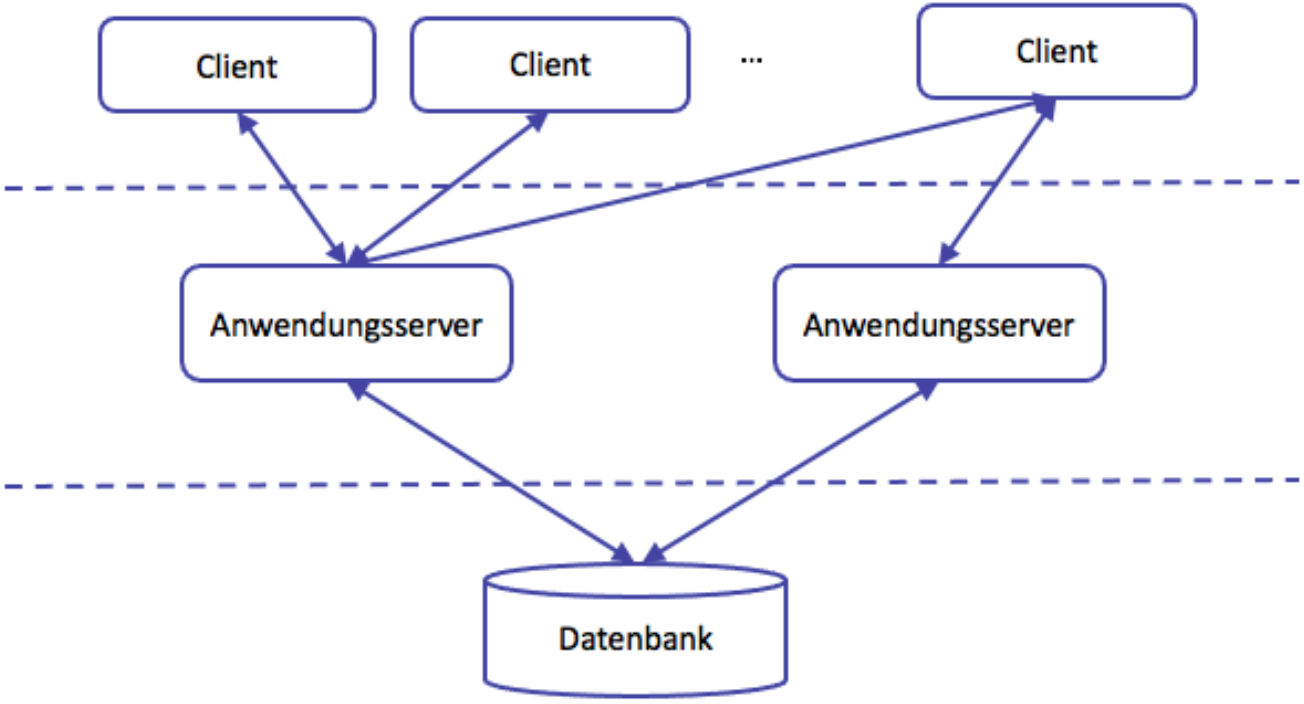
Um diese Funktionalität zu gewährleisten haben sich im Laufe der Jahre Basisfunktionen herausgestellt. Diese wurden allerdings schon 1982 von Codd formuliert.

1. Datenintegration
2. Operationen und Anfragesprachen
3. Datenkatalog
4. Benutzersichten
5. Konsistenz
6. Zugriffskontrolle
7. Transaktionskonzept
8. Synchronisation
9. Datensicherung

$\rightarrow$ Wie werden diese Aufgaben praktisch in der Datenbankarchitektur umgesetzt?

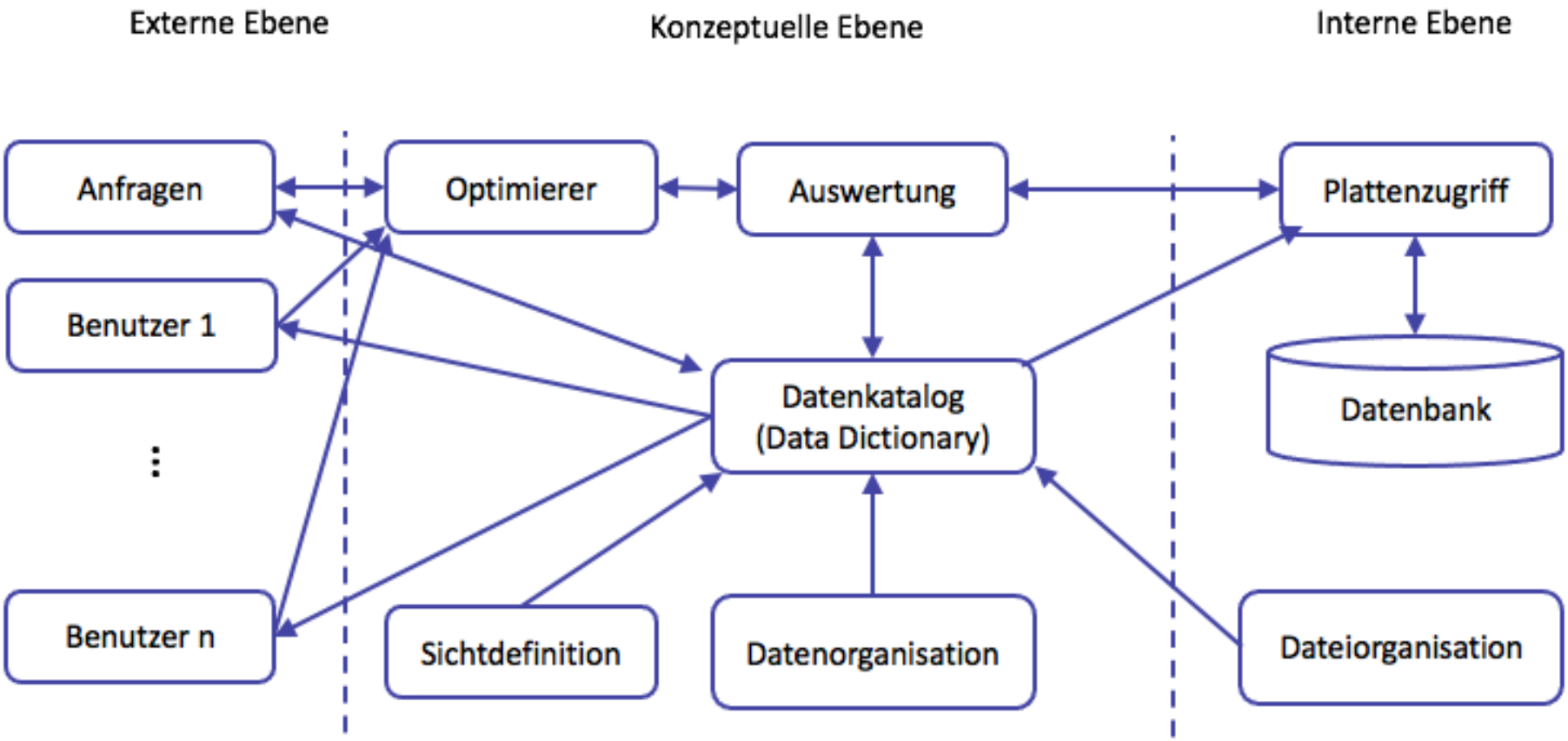
DBMS im Software Stack

Datenbanken sind üblicherweise in einer dreistufigen Architektur eingebaut:



Datenbanken sind oft das *Herz* der Anwendungen. In der Praxis schlagen aber oft mehrere Herzen in einer Anwendung.

Architekturübersicht DMBS



Zusammenfassung

In dieser Vorlesung

- Motivation: Warum Datenbanken?
- Historische Entwicklung Datenbanken und Datenbankmanagementsysteme (DBMS)
- Grundlegende Merkmale von Datenbanken
- Aufgaben eines DBMS nach Codd
- DBMS im Software Stack

- Heuer, Sattler, Saake. Datenbanken: Konzepte und Sprachen. mitp-Verlag
- Saake, Heuer: Datenbanken - Implementierungstechniken, mitp-Verlag, 2005
- Serge Abiteboul, Rick Hull, Victor Vianu. Foundations of Databases. 1995
- Beispiele und Folien basieren teilweise auf Folien von Prof. Schwentick
- Beispiele und Folien basieren teilweise Folien von Prof. Teubner TU Dortumund
  http://dbis.cs.tu-dortmund.de/cms/de/lehre/ss17/infosys/index.html
- Beispiele und Folien basieren teilweise auf Folien von Prof. Naumann HPI
  https://hpi.de/naumann/teaching/teaching/ss13/datenbanksysteme-i.html