



Vorlesung

**Informations- und
Wissensmanagement
(Teil 1)**

Erik Buchmann



Zur Person

Vorab in
eigener
Sache

- Beruflicher Hintergrund
 - Studium und Promotion an der Uni Magdeburg
 - seit September '06 Mitarbeiter am IPD
- Interessensfelder
 - Sensornetzwerke, Ubiquitous Computing, Peer-to-Peer, Datenbanken
 - Datenschutz und vergleichbare gesellschaftliche Probleme moderner Informationssysteme
- Aktuelle Arbeitsschwerpunkte
 - energiebewußte Anfrageverarbeitung in Sensornetzwerken
 - Privatheit in Sensornetzwerken und Ubiquitous Computing-Szenarien





Organisatorisches

Vorab in
eigener
Sache

- **Kontakt**
 - per EMail: buchmann@ipd.uni-karlsruhe.de
 - Sprechzeiten:
 - jeden Donnerstag zwischen 13 und 17 Uhr
 - Gebäude 50.34
 - Raum 339
 - *besser: vorher per Mail anmelden*
- **Folien:** werden wenn möglich vor der Vorlesung, spätestens aber kurz danach ins Internet gestellt
<http://www.aifb.uni-karlsruhe.de/Lehre/Winter2007-08/IWM>
- **Erste Übung:** 07.11.2007





Literatur

[Einleitung](#)

[Welt
ohne DB](#)

[Datenbanken
Übersicht](#)

[Terminologie](#)

[Querying](#)

[DB –
Merkmale](#)

[Anwend.](#)

[Historie](#)

[Schema-
Arch.](#)

[Schluß](#)

- Heuer, A., Saake, G.:
Datenbanken – Konzepte und Sprachen.
2. Aufl., mitp-Verlag, Bonn, Januar 2000.
- Kemper, A., Eickler, A.: *Datenbanksysteme.*
Oldenbourg, 2004.
- Heuer, A., Saake, G., Sattler, K.;
Datenbanken kompakt.
mitp-Verlag, Bonn, 2001.
- Elmasri, R.; Navathe, S.B.;
Fundamentals of Database Systems.
Addison-Wesley, 1999.





Kapitel 1:
**Grundlegende Konzepte und
Architekturen**



Grundlegende Konzepte

[Einleitung](#)

[Welt
ohne DB](#)

[Datenbanken
Übersicht](#)

[Terminologie](#)

[Querying](#)

[DB –
Merkmale](#)

[Anwend.](#)

[Historie](#)

[Schema-
Arch.](#)

[Schluß](#)

- Motivation,
- Komponenten und Funktionen,
- Einsatzgebiete und Grenzen,
- Entwicklungslinien,
- Referenzarchitektur.





Motivation

[Einleitung](#)

[Welt
ohne DB](#)

[Datenbanken
Übersicht](#)

[Terminologie](#)

[Querying](#)

[DB –
Merkmale](#)

[Anwend.](#)

[Historie](#)

[Schema-
Arch.](#)

[Schluß](#)

- Unser Thema: Verwaltung großer Datenmengen und Entwicklung entsprechender Anwendungen.
- Warum Datenbanken?
 - Standardsoftware reduziert Komplexität bei der Anwendungsentwicklung
 - 50% weniger Aufwand bei Anwendungsentwicklung mit Datenbanken, gesamthaft betrachtet, bei überschaubaren Projekten.
 - Größere Differenz bei größeren Projekten.
⇒ Verwendung von Datenbank-Technologie ermöglicht neue Anwendungen; ihre Entwicklung ohne Datenbank-Technologie wäre zu komplex.





Vergleich mit Bekanntem

[Einleitung](#)

[Welt
ohne DB](#)

[Datenbanken
Übersicht](#)

[Terminologie](#)

[Querying](#)

[DB –
Merkmale](#)

[Anwend.](#)

[Historie](#)

[Schema-
Arch.](#)

[Schluß](#)

- Informatik – zwei wichtige Aspekte:
 - Erkennen von Regelmäßigkeiten/Mustern bei der Anwendungsentwicklung. Ausfaktorisieren.
 - Schaffung von Entwicklungswerkzeugen auf abstrakterer Ebene.
- Datenbanken:
 - Anwendung dieser Prinzipien für Entwicklung von Applikationen zur Verwaltung großer Datenmengen.





Situation ohne Datenbanken (1)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

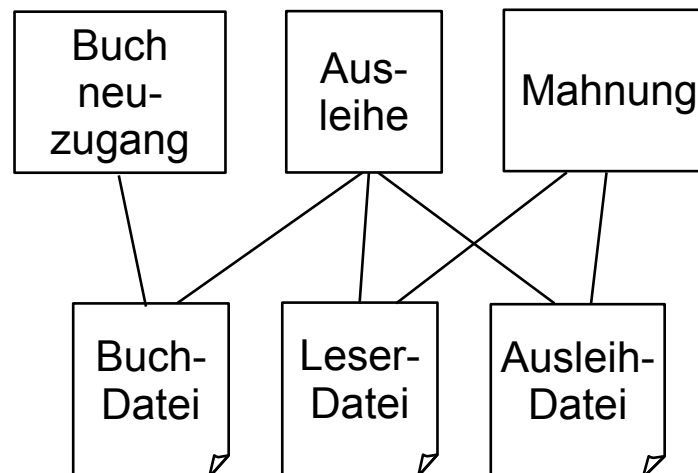
Anwend.

Historie

Schema-
Arch.

Schluß

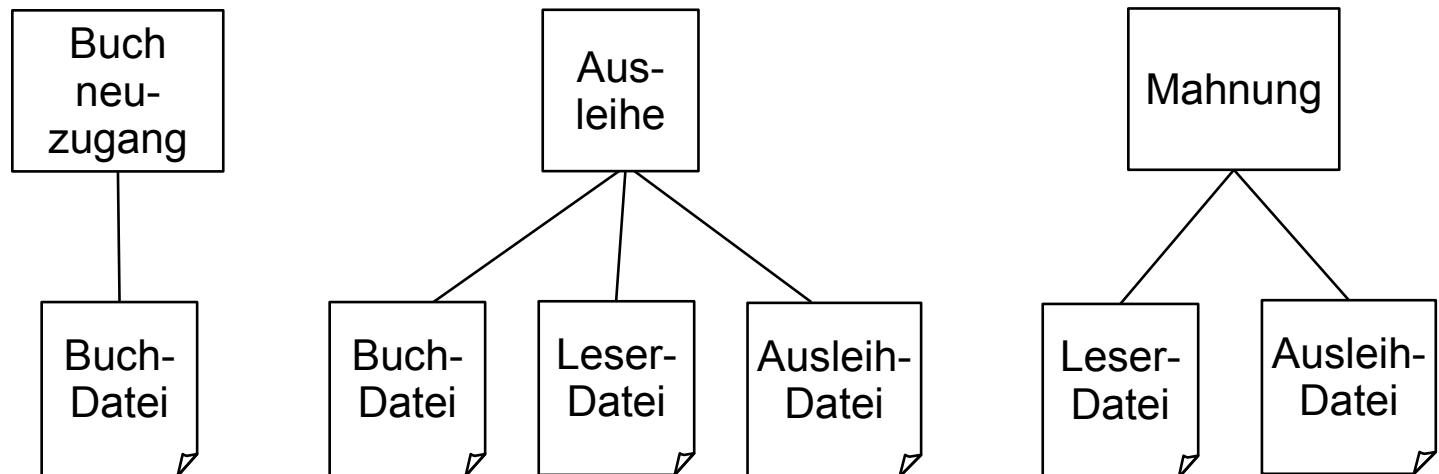
- Zugriff auf Daten, in Files abgelegt.
- Funktionalität hierfür Teil der Anwendungen.



Situation ohne Datenbanken (2)

Einleitung
[Welt ohne DB](#)
Datenbanken
Übersicht
Terminologie
Querying
DB –
Merkmale
Anwend.
Historie
Schema-
Arch.
Schluß

- Redundanz.





Datenredundanz (1)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

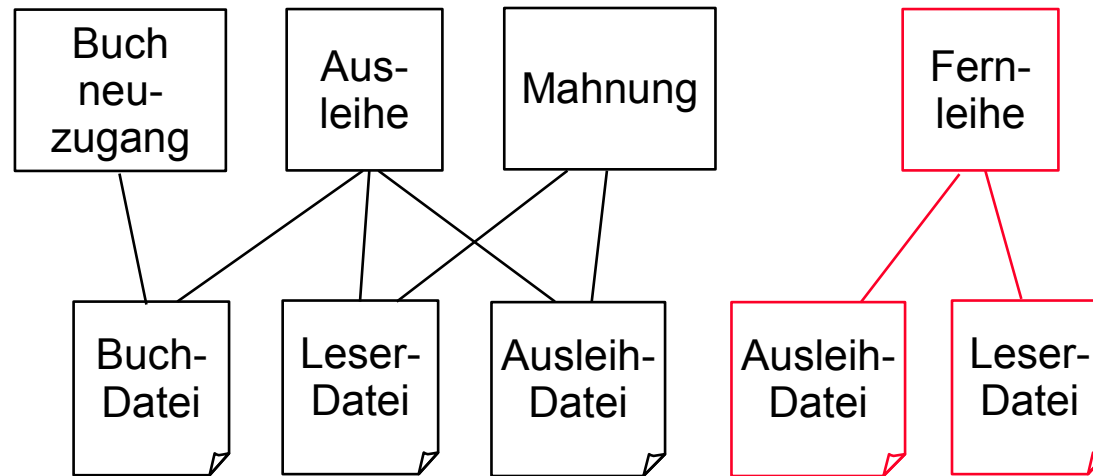
Schema-
Arch.

Schluß

- Daten sind **redundant**: mehrfach gespeichert;
- Probleme:
 - Verschwendung von Speicherplatz,
 - „Vergessen“ von Änderungen,
 - Inkonsistenzen; keine zentrale, „genormte“ Datenhaltung.



Datenredundanz (2)



- **Beispiel:**
 - Benutzer i. Allg. mehrmals erfaßt.
 - Was passiert, wenn Benutzer umzieht?
 - Wenn er neue Adresse nur der Ausleihe, nicht aber der Fernleihe mitteilt?



Datenredundanz (3)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Basis- oder Anwendungssoftware verwaltet ihre eigenen Daten in ihren eigenen (Datei-)Formaten.
 - Textverarbeitung: Texte, Artikel und Adressen,
 - Buchhaltung: Artikel, Adressen,
 - Lagerverwaltung: Artikel, Aufträge,
 - Auftragsverwaltung: Aufträge, Artikel, Adressen,
 - CAD-System: Artikel, technische Bauteile.
- keine integrierten Systeme, Datenaustausch nur über Umwege möglich



Weitere Probleme ohne Datenbanken (1)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

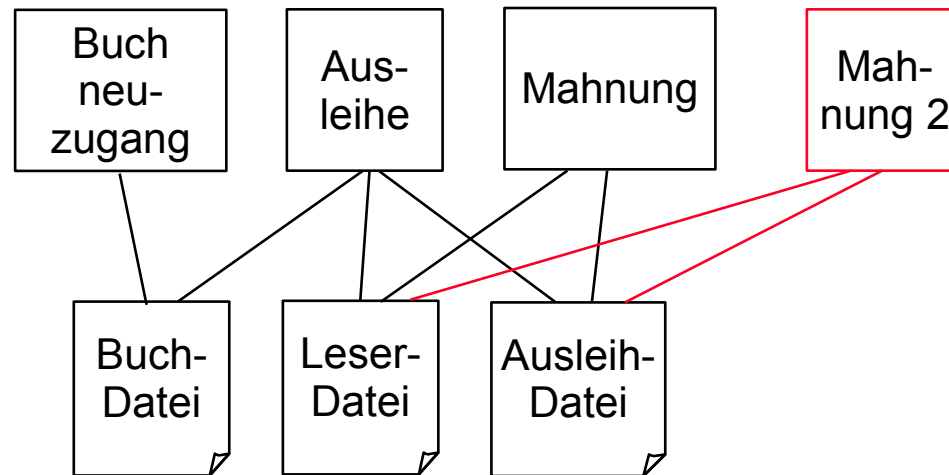
Anwend.

Historie

Schema-
Arch.

Schluß

- Mehrere Benutzer oder Anwendungen können i. a. nicht parallel auf den gleichen Daten arbeiten („**Nebenläufigkeit**“, Concurrency), ohne sich zu stören.



- *Es wäre extrem aufwendig, Lösungen hierfür selbst entwickeln und realisieren zu müssen.*

Weitere Probleme ohne Datenbanken (2)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Transaktionseigenschaften – insbesondere **Atomarität** und **Isolation**.

- Atomarität

- Beispiel, „Bank-Szenario“:

Name	Konto	Guthaben
Erik B.	1892	1000
Mirco S.	2434	2000

- Überweisung – zwei Elementaroperationen.
 - Abbuchung(Erik B., 500),
 - Einzahlung(Mirco S., 500).
- Isolation
 - was passiert, wenn parallele Transaktionen die gleichen Datensätze schreiben wollen?



Weitere Probleme ohne Datenbanken (3)

- **Keine physische Datenunabhängigkeit:**
 - ‚Naives‘ file-basiertes Vorgehen bei Programmierung/Benutzung von Anwendungen: Man muß interne (physische) Repräsentation der Daten kennen, z.B. bei Speicherung als Tabelle Reihenfolge der Zeilen und Spalten.

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

Weitere Probleme ohne Datenbanken (4)

- Beispiel für physische Datenunabhängigkeit: sind die Relationen identisch?

Name	Vorname	Konto	Adresse
Erik	Buchmann	1892	Im Nirgendwo 7
Mirco	Stern	2434	Grube 3
Klemens	Böhm	4593	Auf dem Holzweg 5
Markus	Bestehorn	6725	Umgehungsstraße 42

Vorname	Name	Adresse	Konto
Böhm	Klemens	Auf dem Holzweg 5	4593
Stern	Mirco	Grube 3	2434
Bestehorn	Markus	Umgehungsstraße 42	6725
Buchmann	Erik	Im Nirgendwo 7	1892

Einleitung
[Welt ohne DB](#)
Datenbanken Übersicht
Terminologie
Querying
DB – Merkmale
Anwend.
Historie
Schema-Arch.
Schluß



Weitere Probleme ohne Datenbanken (5)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- physische Datenunabhängigkeit:
 - Verstecken physischen Eigenschaften vor dem Anwendungsentwickler.
 - herausragendes Merkmal von Datenbanken
 - Beispiel:

```
select NAME, VORNAME
from PERSON
where ALTER > 40
```
 - deklarative Anfrage, unabhängig von der Art der Datenspeicherung



Weitere Probleme ohne Datenbanken (6)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- **Datenschutz und Datensicherheit** sind nicht gewährleistet.
 - Datenschutz (kein unbefugter Zugriff),
 - Wer darf welche Daten unter welchen Umständen sehen?
 - Datensicherheit (kein ungewollter Datenverlust).
 - Ab wann genau sind Daten 'fest' gespeichert?
 - Was ist mit Betriebssystem-Cache, Festplatten-Cache, Network-Attached Storage etc. pp.?
- Mittel des Betriebssystems sind nicht ausreichend





Weitere Probleme ohne Datenbanken (7)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Überwachung der **Datenintegrität** schwierig
 - welche Zustände der Datenbank sind korrekt?
 - Datenbank muss von einem konsistenten Zustand in einen anderen konsistenten Zustand übergehen
 - Beispiele für Integritätsbedingungen:
 - Geburtsdatum < aktuelles Datum
 - Kontostand > -1 * Dispo-Kredit
 - Name != *NULL*
 - Wenn *Auto* in der DB existiert, muss auch *Fahrzeughalter* existieren
- Datenintegrität ohne DB schwer zu garantieren

Von Dateien zu Datenbanken

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

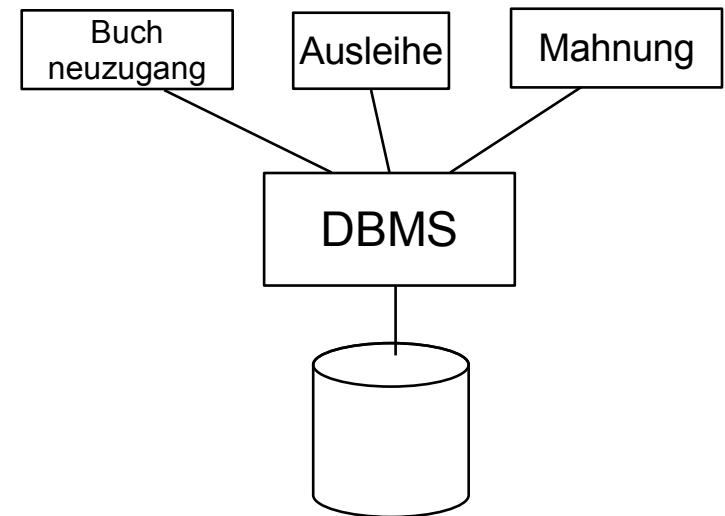
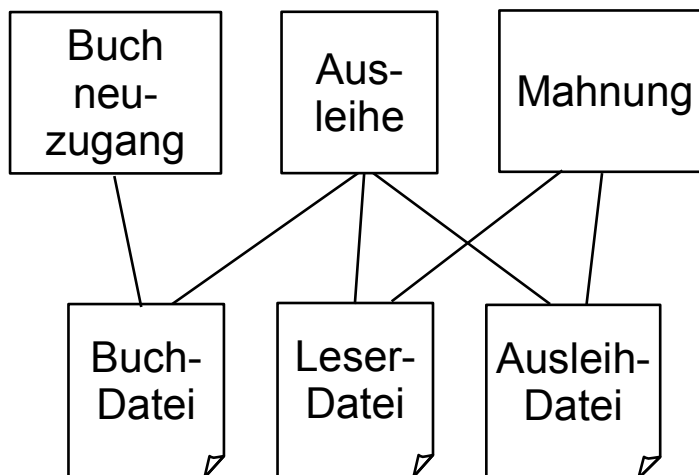
Anwend.

Historie

Schema-
Arch.

Schluß

- Zugriff auf Daten, in Files abgelegt.
- Funktionalität hierfür Teil der Anwendungen (Berücksichtigung der physischen Ebene, Nebenläufigkeit, Datenschutz, Konsistenz).
- Datenbanken:
Ausfaktorieren dieser Funktionalität.





Mit Datenbanken: Datenintegration

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Basis- und Anwendungssoftware **arbeitet auf den selben Daten.** (z.B. Adressen nur einmal gespeichert)
 - Datenbanksysteme können große Datenmengen effizient verwalten.
→ Kein Problem, große Datenmengen an einem Ort anzuhäufen.
 - mehrere Benutzer können parallel auf Datenbanken arbeiten (Transaktionskonzept).
- Auch ohne Datenbanken möglich, Redundanz zu vermeiden. Datenbanken machen dies aber leicht und bieten weitere Features.

Relationenmodell (1)

- Konzeptuell ist eine Datenbank üblicherweise eine Menge von Tabellen.

AUSLEIH	INV.NR	NAME
	4711	Meyer
	1201	Schulz
	0007	Müller
	4712	Meyer

BUCH	INV.NR	TITEL	ISBN	AUTOR
	0007	Dr. No	3-324	Fleming
	1201	Objektbanken	3-111	Heuer
	4711	Datenbanken	3-345	Vossen
	4712	Datenbanken	3-345	Ullman
	4717	PASCAL	3-989	Wirth

Einleitung
Welt
ohne DB
Datenbanken
Übersicht
Terminologie
Querying
DB –
Merkmale
Anwend.
Historie
Schema-
Arch.
Schluß



Relationenmodell (2)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Tabellen = „**Relationen**“
- Relationen sind Mengen von „**Tupeln**“.



Relationenmodell (3)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

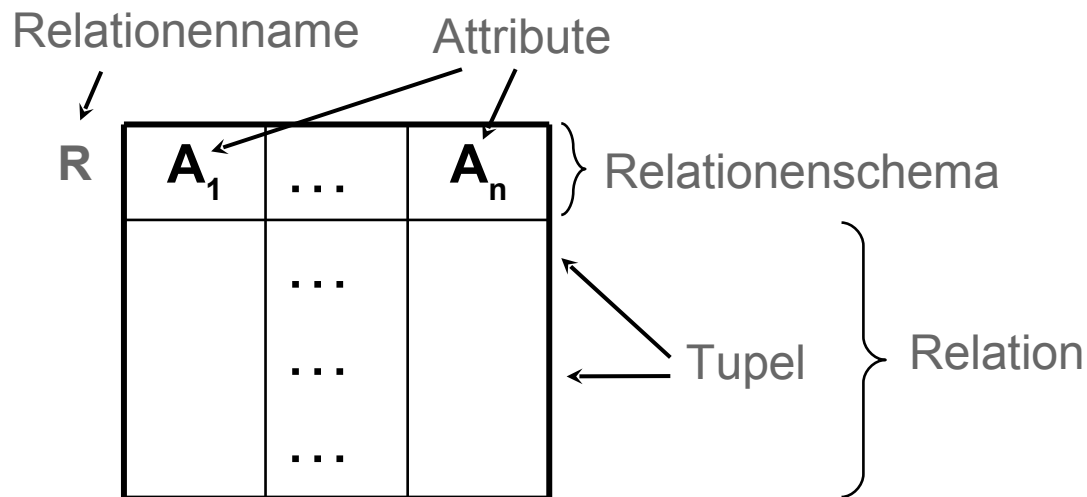
Anwend.

Historie

Schema-
Arch.

Schluß

- Fett geschriebene Zeilen: **Relationenschema**
- Weitere Einträge in der Tabelle: **Relation**
- Eine Zeile der Tabelle: **Tupel**
- Eine Spaltenüberschrift: **Attribut**



DBMS vs. DBS

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

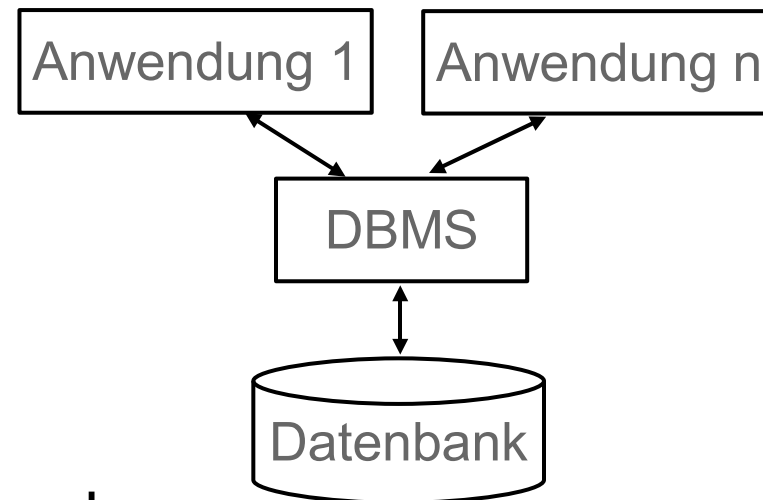
Anwend.

Historie

Schema-
Arch.

Schluß

- **DBMS:** Datenbank-Management-System
- **DBS:** Datenbanksystem (DBMS + Datenbank)



- **DB:** Datenbank
 - DBMS ist Software zur Datenverwaltung, Datenbank sind die eigentlichen Daten.
 - Ein DBMS, aber mehrere (viele) Datenbanken.



Integritätsbedingungen

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Relationenschema:
lokale Integritätsbedingungen
 - INV.NR ist Schlüssel für BUCH,
 - INV.NR darf nicht doppelt vergeben werden.
- Datenbankschema: Menge von
Relationenschemata
globale Integritätsbedingungen
 - INV.NR in AUSLEIH ist Fremdschlüssel bezüglich BUCH,
 - INV.NR taucht in einem anderen Relationenschema als Schlüssel auf.
 - Integrität der DB bei Änderungs- und Löschoperationen gewährleisten!





SQL als Standard-Anfragesprache (1)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Wie greife ich auf Informationen in meiner Datenbank zu?
 - Manipulation und Abfrage von Datenbank *und* Datenbankschema in SQL möglich
- Anfragen, Queries
 - arbeiten auf Relationen
 - Query-Ergebnisse sind auch wieder Relation

Verbindung zum Datenbankserver erfolgreich

Bitte Anfrage eingeben:

```
select * from Bücher
```

Zuvor wurde eingegeben:
INSERT INTO Bücher
VALUES ('12345', 'Datenbanken', 'MITP');
commit

Abfrage starten

```
select * from Bücher
```

ISBN	TITEL	
12345	Datenbanken	MITP

time	percent	rows	query
0.001728	100%	1	select * from Bücher
0.001728			

Verbindung zum Datenbankserver erfolgreich

Bitte Anfrage eingeben:

```
select * from Bücher
```

Zuvor wurde eingegeben:
INSERT INTO Bücher
VALUES ('54321', 'Tante Julia', NULL);
commit

Abfrage starten

```
select * from Bücher
```

ISBN	TITEL	
12345	Datenbanken	MITP
54321	Tante Julia	

time	percent	rows	query
0.001733	100%	2	select * from Bücher
0.001733			



SQL als Standard-Anfragesprache (2)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- weiteres Beispiel:

```
select INVENTARNR, TITEL, NAME
from BUCH, AUSLEIH
where NAME = 'Meyer'
        and Titel = 'Datenbanken'
```



Relation für die folgenden Beispiele

- Szenario: Bibliothek

AUSLEIH	INV.NR	NAME
	4711	Meyer
	1201	Schulz
	0007	Müller
	4712	Meyer

BUCH	INV.NR	TITEL	ISBN	AUTOR
	0007	Dr. No	3-324	Fleming
	1201	Objektbanken	3-111	Heuer
	4711	Datenbanken	3-345	Vossen
	4712	Datenbanken	3-345	Ullman
	4717	PASCAL	3-989	Wirth

Einleitung
Welt
ohne DB
Datenbanken
Übersicht
Terminologie
Querying
DB –
Merkmale
Anwend.
Historie
Schema-
Arch.
Schluß

Anfrageoperationen (1)

- Einleitung
- Welt ohne DB
- Datenbanken Übersicht
- Terminologie
- Querying
- DB – Merkmale
- Anwend.
- Historie
- Schema-Arch.
- Schluß

- SELEKTION: Zeilen (Tupel) auswählen.

$\sigma_{\text{NAME}=\text{Meyer}}$ (AUSLEIH)

INV.NR	NAME
4711	Meyer
4712	Meyer

- PROJEKTION: Spalten (Attribute) auswählen.

$\pi_{\text{INV.NR}, \text{TITEL}}$ (BUCH)

INV.NR	TITEL
0007	Dr. No
1201	Objektbanken
4711	Datenbanken
4712	Datenbanken
4717	PASCAL

Anfrageoperationen (2)

- Einleitung
- Welt ohne DB
- Datenbanken Übersicht
- Terminologie
- Querying
- DB – Merkmale
- Anwend.
- Historie
- Schema-Arch.
- Schluß

- Beispiel für komplexen Algebraausdruck:

$$\pi_{\text{INV.NR, TITEL}}(\sigma_{\text{AUTOR}=\text{'Heuer'}}(\text{BUCH}))$$

Ausgangsrelation:

INV.NR	TITEL	ISBN	AUTOR
0007	Dr. No	3-324	Fleming
1201	Objektbanken	3-111	Heuer
4711	Datenbanken	3-345	Vossen
4712	Datenbanken	3-345	Ullman
4717	PASCAL	3-989	Wirth

Ergebnis:

INV.NR	TITEL
1201	Objektbanken



Anfrageoperationen (3)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Weitere Operationen: Join, Vereinigung, Differenz, Durchschnitt, Umbenennung.
- Alle Operationen beliebig kombinierbar („Algebra“, *Query-Algebra*).





Benutzersichten (1)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Datenbankabfragen (Queries) unter einem „Sichtnamen“ als „virtuelle“ Relation speichern
 - **Sicht = Query (+ Name)**



Definition von Sichten in SQL

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

MGA	Mitarbeiter	Gehalt	Abteilung
	Böhm	60	IPD
	Buchmann	30	IPD
	Lockemann	90	IPD
	Studer	60	AIFB

- **create view** MA **as**
select Mitarbeiter, Abteilung
from MGA
where Gehalt > 40

MA	Mitarbeiter	Abteilung
	Böhm	IPD
	Lockemann	IPD
	Studer	AIFB

- **select** * **from** MA **where** Gehalt < 80



Benutzersichten (2)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- werden häufig für Datenschutzmechanismen eingesetzt
 - unterschiedliche Benutzer sehen unterschiedlichen Ausschnitt der Datenbank.
 - Beispiel – Universität (Studierende, Dozenten, Prüfungsamt, Dekanat etc.):
 - Datenschutz,
 - Übersichtlichkeit,
 - organisatorische Gründe (Ablenkung, Einmischung).





Optimierer

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Problem: Finde einen Relationenalgebra-Ausdruck,
 - der äquivalent ist („das gleiche Ergebnis liefert“) zum gegebenen,
 - aber effizienter auszuwerten ist.





Algebraische Optimierung

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Allgemeine Regel: äquivalent sind
 - $\sigma_{A1 = \text{Konst}}(\sigma_{A2 = \text{Konst}}(\text{REL}))$
 - $\sigma_{A2 = \text{Konst}}(\sigma_{A1 = \text{Konst}}(\text{REL}))$
- Beispiel:
 - $\sigma_{\text{Vorname} = \text{'Klemens'}}(\sigma_{\text{Wohnort} = \text{'KA'}}(\text{PERSON}))$
 - $\sigma_{\text{Wohnort} = \text{'KA'}}(\sigma_{\text{Vorname} = \text{'Klemens'}}(\text{PERSON}))$
- Welche Variante ist besser?
 - Beobachtung: zweite Variante bei realistischen Beispieldaten überlegen. *Warum?*





Physische Datenunabhängigkeit (1)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Anfragen sind **deklarativ**, Anwender sagt nur, welches Ergebnis, nicht wie es ermittelt werden soll.
- Beispiel:
select *
from PERSON
where VORNAME = 'Klemens'
and WOHNORT = 'KA'
- Datenbank 1:
10 Tupel mit VORNAME = 'Klemens',
1000 Tupel mit WOHNORT = 'KA'.
- Datenbank 2: umgekehrt.



Physische Datenunabhängigkeit (2)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Physische Datenunabhängigkeit – DBMS stellt sicher,
 - dass Anfrage weiterhin gut funktioniert, auch wenn physische Darstellung der Daten sich geändert hat.
 - dass Anfrage in unterschiedlichen Datenbanken (gleiches Schema, aber unterschiedliche Häufigkeiten der Daten) funktioniert.
- Verringert Komplexität bei der Anwendungsentwicklung.





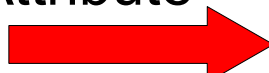
Logische Datenunabhängigkeit

- Einleitung
- Welt ohne DB
- Datenbanken Übersicht
- Terminologie
- Querying
- DB – Merkmale
- Anwend.
- Historie
- Schema-Arch.
- Schluß

- Änderungen an der logischen Sicht bleiben für Anfragen und Anwendungsprogramme unsichtbar
- Realisierung über Sichten, z.B.:

REL (vorher)

Mitarbeiter	Abteilung
Buchmann	IPD

Attribute

umbenennen

REL (nachher)

Employee	Dept
Buchmann	IPD

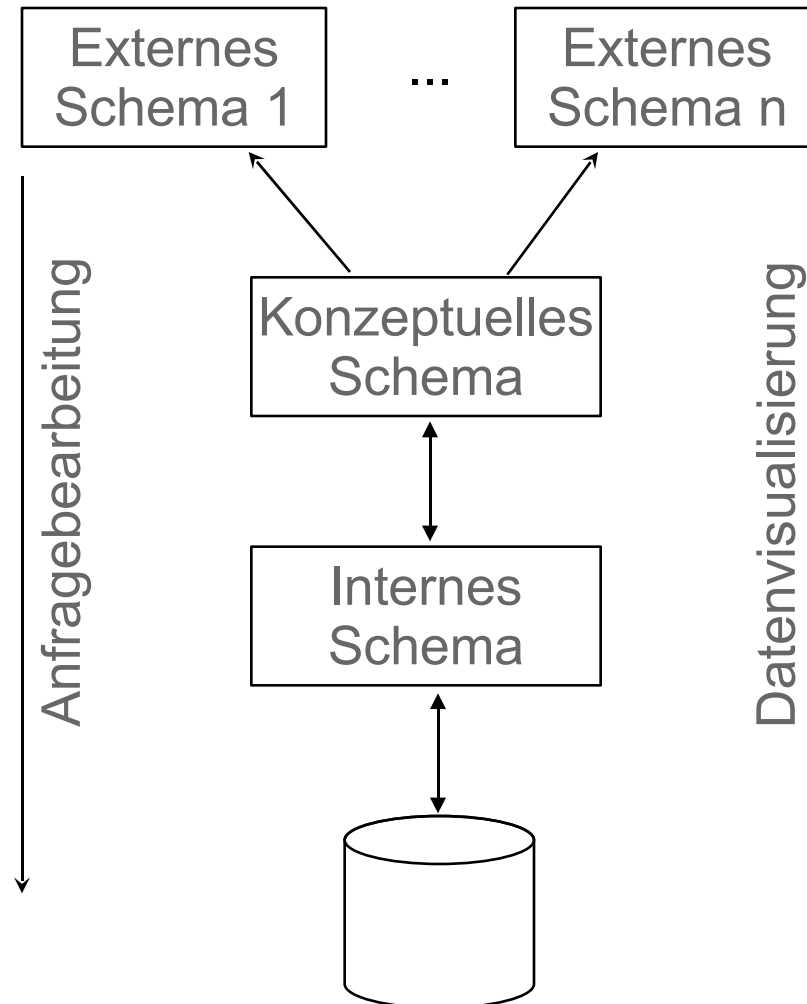
```
create view MA as
select
  Mitarbeiter,
  Abteilung
from REL
```

```
create view MA as
select
  Employee as Mitarbeiter,
  Dept as Abteilung
from REL
```





3-Ebenen Architektur



- Einleitung
- Welt ohne DB
- Datenbanken Übersicht
- Terminologie
- Querying
- DB – Merkmale
- Anwend.
- Historie
- Schema-Arch.
- Schluß





Prinzipien

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Grundprinzip moderner Datenbanksysteme
 - 3-Ebenen-Architektur
physische Datenunabhängigkeit,
logische Datenunabhängigkeit
 - Trennung zwischen **Schema**
(etwa Tabellenstruktur)
und **Instanz** (etwa Tabelleninhalt).
- Angelehnt an 9 Codd'sche Regeln.
(Eigenschaften, die Datenbank haben muß,
um als relational zu gelten.)





Die neun Codd'schen Regeln

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- 1. Integration:** nichtredundante Datenverwaltung,
- 2. Operationen:** Speichern, Suchen, Ändern,
- 3. Katalog:** Metadaten im Data Dictionary,
- 4. Benutzersichten,**
- 5. Integritätssicherung:** Korrektheit der Datenbank,
- 6. Datenschutz:** Ausschluß unautorisierter Zugriffe,
- 7. Transaktionen:** mehrere atomare DB-Operationen als Funktionseinheit,
- 8. Synchronisation:** parallele Transaktionen koordinieren,
- 9. Datensicherung:** Wiederherstellung von Daten nach Systemfehlern.





Data Dictionary – Illustration

AUSLEIH	INV.NR	NAME

BUCH	INV.NR	NAME	TITEL	ISBN	AUTHOR

REL	RNR	NAME
	1	AUSLEIH
	2	BUCH

ATTR	RNR	NAME	TYP	SCHLÜSSEL
	1	INV.NR	int	true
	1	NAME	String	false
	2	INV.NR	int	true
	2	TITEL	String	false

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß





Einbindung in Programme

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Nahtlose Einbindung von Querysprachen in Programme.
 - Queryergebnis ans Programm.
 - Programmvariablen als Bestandteil der Anfrage.

- Beispiel JDBC:

```
...  
String query = "SELECT titel, preis, "  
+ "bestand FROM buch";
```

```
Statement stmt = con.createStatement ();  
ResultSet rs = stmt.executeQuery (query);
```



Zugriff über WWW

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

```
<?php
/* Verbindung aufbauen, auswählen einer Datenbank */
$link = mysql_connect("mysql_host", "mysql_user", "mysql_password")
or die("Keine Verbindung möglich!");
print "Verbindung zum Datenbankserver erfolgreich";
mysql_select_db("Meine_DB") or die("Auswahl der DB fehlgeschlagen");

// ausführen einer SQL Anfrage
$query = "SELECT * FROM Meine_Tabelle";
$result = mysql_query($query) or die("Anfrage fehlgeschlagen");

// Ausgabe der Ergebnisse in HTML
print "<table>\n";
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
    print "\t<tr>\n";
    foreach ($line as $col_value) {
        print "\t\t<td>$col_value</td>\n";
    }
    print "\t</tr>\n";
}
print "</table>\n";

mysql_free_result($result); // Freigeben des Resultsets

mysql_close($link); // schliessen der Verbindung
?>
```



Wie siehts mit der Performanz aus?

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Andere Software-Systeme können große Mengen von Daten nicht effizient verarbeiten.
- Datenbank-Optimierer ist Ergebnis jahrzehntelanger Forschung
- Schein-Gegenargument:
 - „Generischer Code stets langsamer.“
 - B-Baum Implementierung in MS-SQL
→ Weltrekord TPC-C.





Datenbanken sind groß

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Warum dieser Aufwand?
Reichen herkömmliche Techniken wirklich nicht?
- Datenbanken sind groß:
 - Größe einzelner Relationen,
 - Anzahl der Relationen,
 - Anfragen sind komplex
(synthetisch generierte Anfragen haben leicht
ein Dutzend oder mehr Joins),
 - viele Benutzer,
 - viele gleichzeitige Zugriffe,
 - hohe anliegende Arbeitslast.





Zahlen – Bank

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

Große europäische Bank; Zahlen von 05/2005:

- > 70.000 Datenbanken
(zumeist DB2, Oracle, IMS, SQLServer)
- Größte Datenbank:
 - 25 Mio. Transaktionen pro Tag
(30 pro Sekunde),
 - Datenvolumen: 80 TB.
- 99,7% Verfügbarkeit,
- 210 Mio. gedruckte Seiten pro Jahr,
- 60 Mitarbeiter nur für Datenbankadministration.





Zahlen – S.A.P.

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

Zahlen von 05/2005:

- Anzahl Tabellen in SAP ERP (früher R/3):
ca. 20000,
 - eher wenige große Tabellen; viele kleine, die den Charakter von .cfg oder .ini Files in einem OS haben
- Anzahl Nutzer
 - registrierte Benutzer: Tausende
 - “concurrent users”: 20-50 (i. Allg.)





Datenbanken sind groß. (2)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Unter diesen Umständen
 - spielt Reihenfolge der Joins große Rolle (physische Datenunabhängigkeit),
 - sind logische Datenunabhängigkeit,
 - ausgefeilte Nebenläufigkeits-Zugriffsmechanismen,
 - Benutzerverwaltung etc. unumgänglich.





Einsatzgebiete und Grenzen (1)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Klassische Einsatzgebiete:
 - Viele Objekte (15000 Bücher, 300 Benutzer, 100 Ausleihvorgänge pro Woche, ...)
 - wenige Objekttypen (BUCH, BENUTZER, AUSLEIHUNG),
 - etwa Buchhaltungssysteme, Auftragserfassungssysteme, Bibliothekssysteme, Handel, Lagerhaltung und Produktion, Logistik, Finanzen (Banken, Versicherungen usw.), Forschung und Entwicklung.





Einsatzgebiete und Grenzen (2)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Aktuelle Anwendungen:
 - entscheidungsunterstützende Systeme (Data Warehouses, OLAP),
 - E-Commerce,
 - NASA Earth Observation System (Petabyte-Datenbanken),
 - Data Mining.





Einsatzgebiete und Grenzen (3)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Normalerweise sind DBMS überfordert mit:
 - Expertensysteme – wenige Objekte, viele Objekttypen, kompliziertere Operationen;
ABER: Deduktive Datenbanksysteme,
 - Ähnlichkeitssuche, Ranking – anderes Paradigma, keine gute interne Unterstützung,
ABER: Anfragesprachen, aktuelle Forschung zu entsprechenden Erweiterungen existierender Systeme.
 - XML, Multimedia – zuviele Datenobjekte; für einfache Informationsbedürfnisse zuviele komplexe Anfragen, kein Multimedia-Support;
ABER: XML-Datenbanken, Multimedia-Erweiterungen, ORDBMS.



Historie

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Anfang 60er Jahre: elementare Dateien, anwendungsspezifische Datenorganisation (geräteabhängig, redundant, inkonsistent)
- Ende 60er Jahre: Dateiverwaltungssysteme (SAM, ISAM) mit Dienstprogrammen (Sortieren) (geräteunabhängig, aber redundant und inkonsistent)
- 1970: Codd, E.F. „A Relational Model of Data for Large Shared Data Banks“. Communications of the ACM 13 (6): 377-387. → *Ursprung relationaler DB*
- 70er Jahre: Datenbanksysteme (Geräte- und Datenunabhängigkeit, redundanzfrei, konsistent)





System-Architekturen

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

- [Einleit.](#)

- ANSI-
SPARC

Schluß

- Beschreibung der Komponenten eines Datenbanksystems.
- Standardisierung der Schnittstellen zwischen Komponenten.
- Architekturvorschläge:
 - **ANSI-SPARC-Architektur**
~>Drei-Ebenen-Architektur.
 - Fünf-Schichten-Architektur
~>beschreibt Transformationskomponenten.





ANSI-SPARC-Architektur (1)

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

- Einleit.

- ANSI-
SPARC

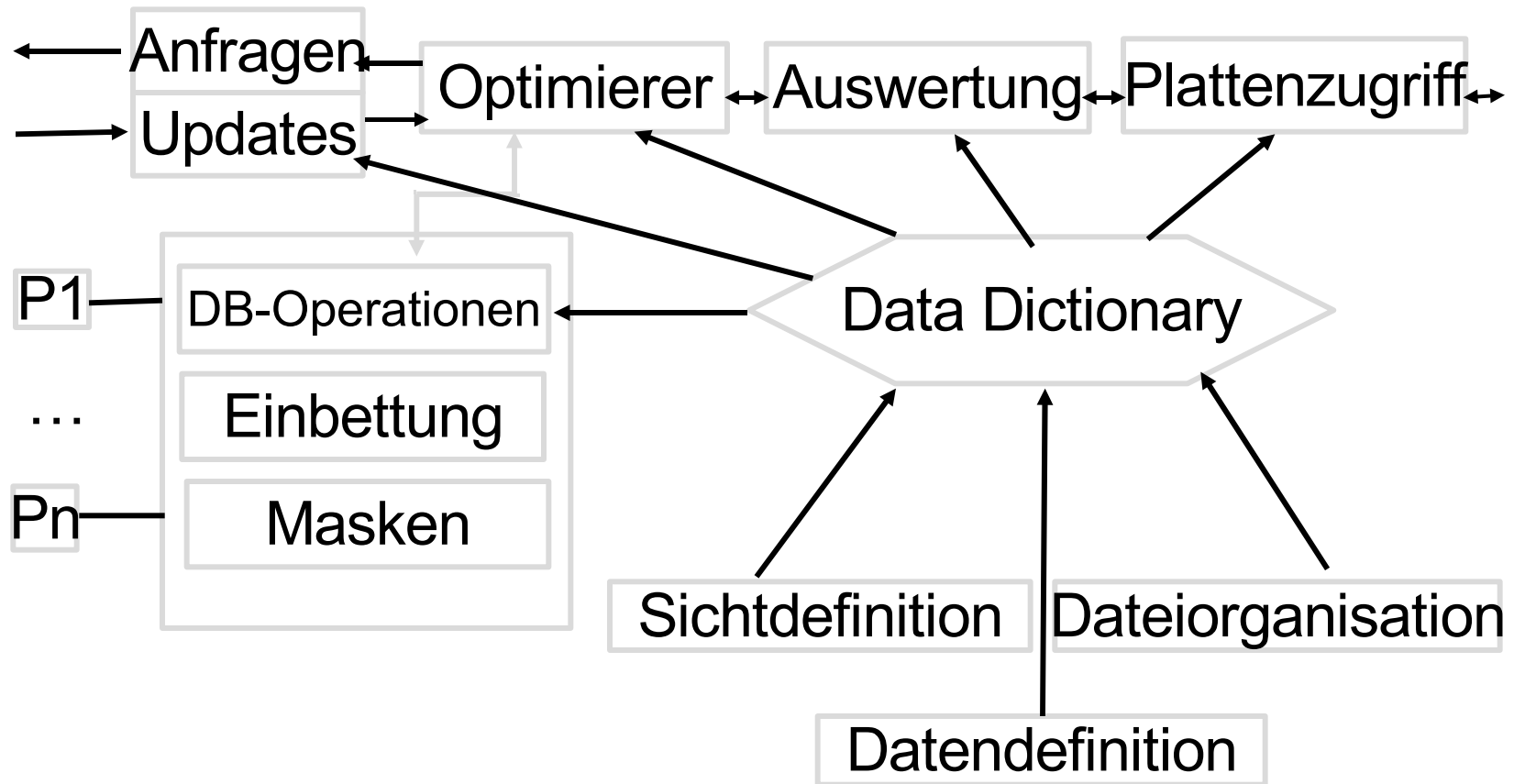
Schluß

- ANSI: American National Standards Institute.
- SPARC: Standards Planning and Requirement Committee.
- Vorschlag von 1978.
- Im wesentlichen Grobarchitektur verfeinert:
 - Interne Ebene / Betriebssystem verfeinert,
 - mehr interaktive und Programmier-Komponenten,
 - Schnittstellen bezeichnet und normiert.



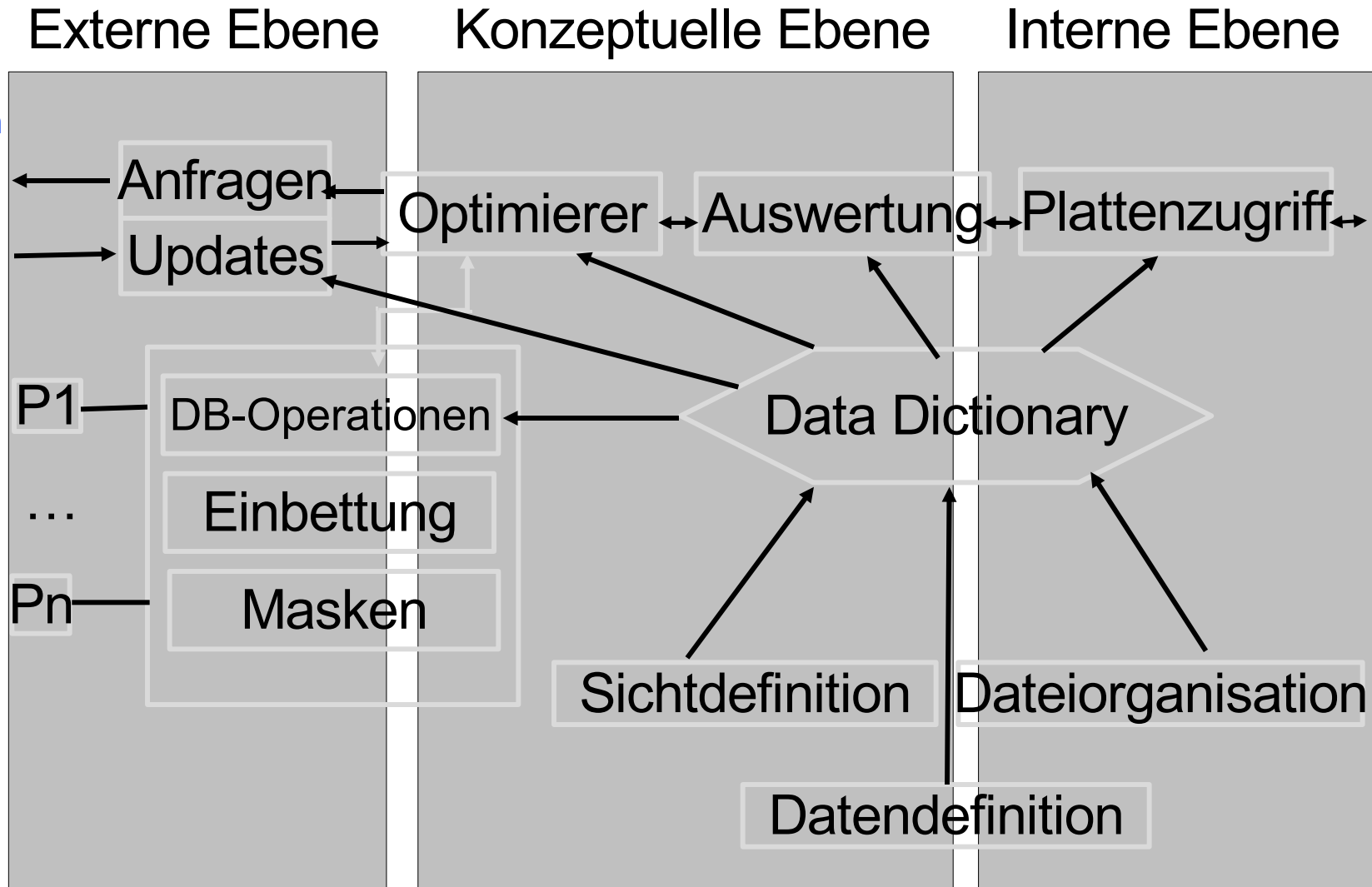
ANSI-SPARC-Architektur (2)

- Einleitung
- Welt ohne DB
- Datenbanken Übersicht
- Terminologie
- Querying
- DB – Merkmale
- Anwend. Historie
- Schema-Arch.
- Einleit.
- [ANSI-SPARC](#)
- Schluß



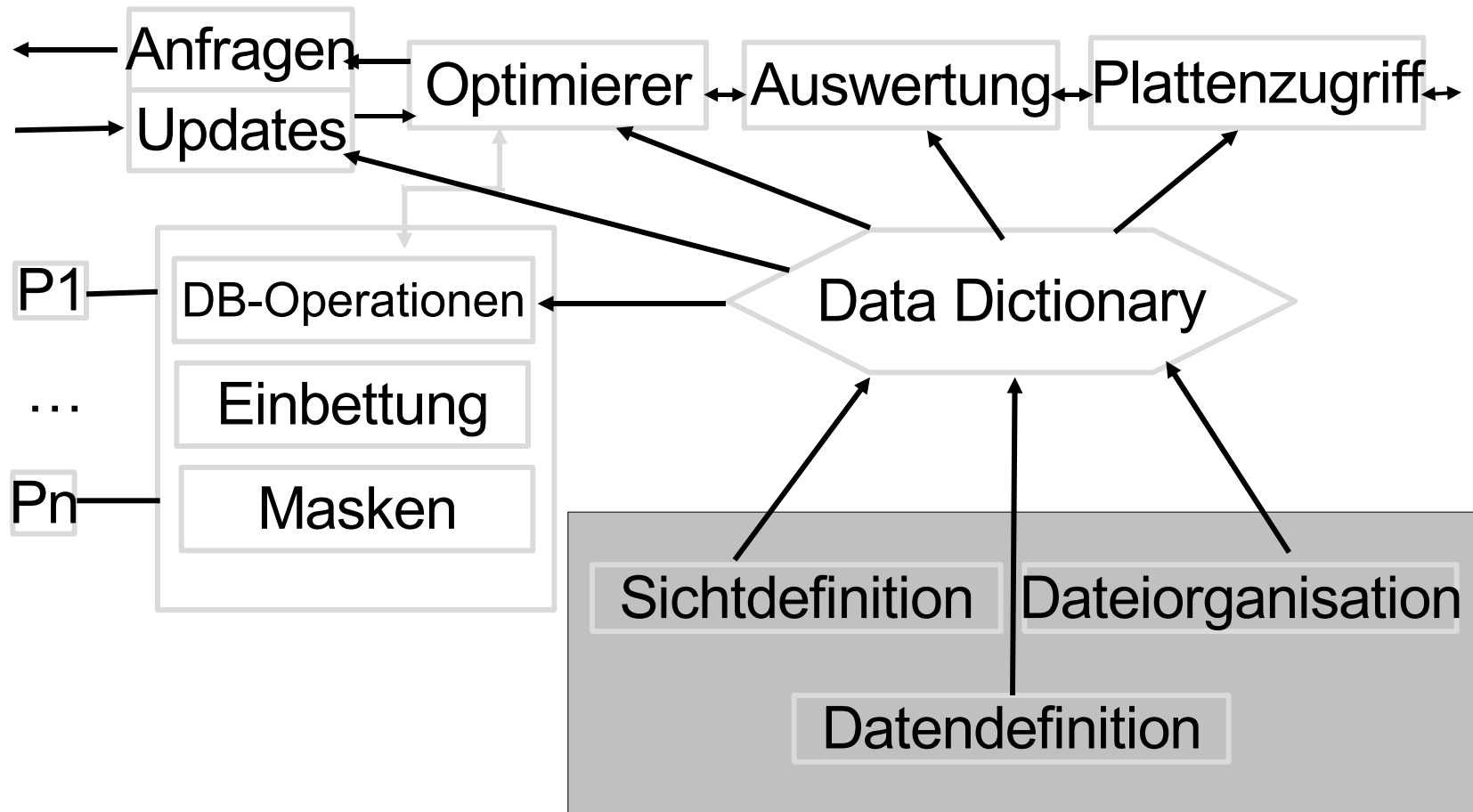
ANSI-SPARC-Architektur (3)

- Einleitung
- Welt ohne DB
- Datenbanken Übersicht
- Terminologie
- Querying
- DB – Merkmale
- Anwend. Historie
- Schema-Arch.
- Einleit.
- ANSI-SPARC
- Schluß



ANSI-SPARC-Architektur (4)

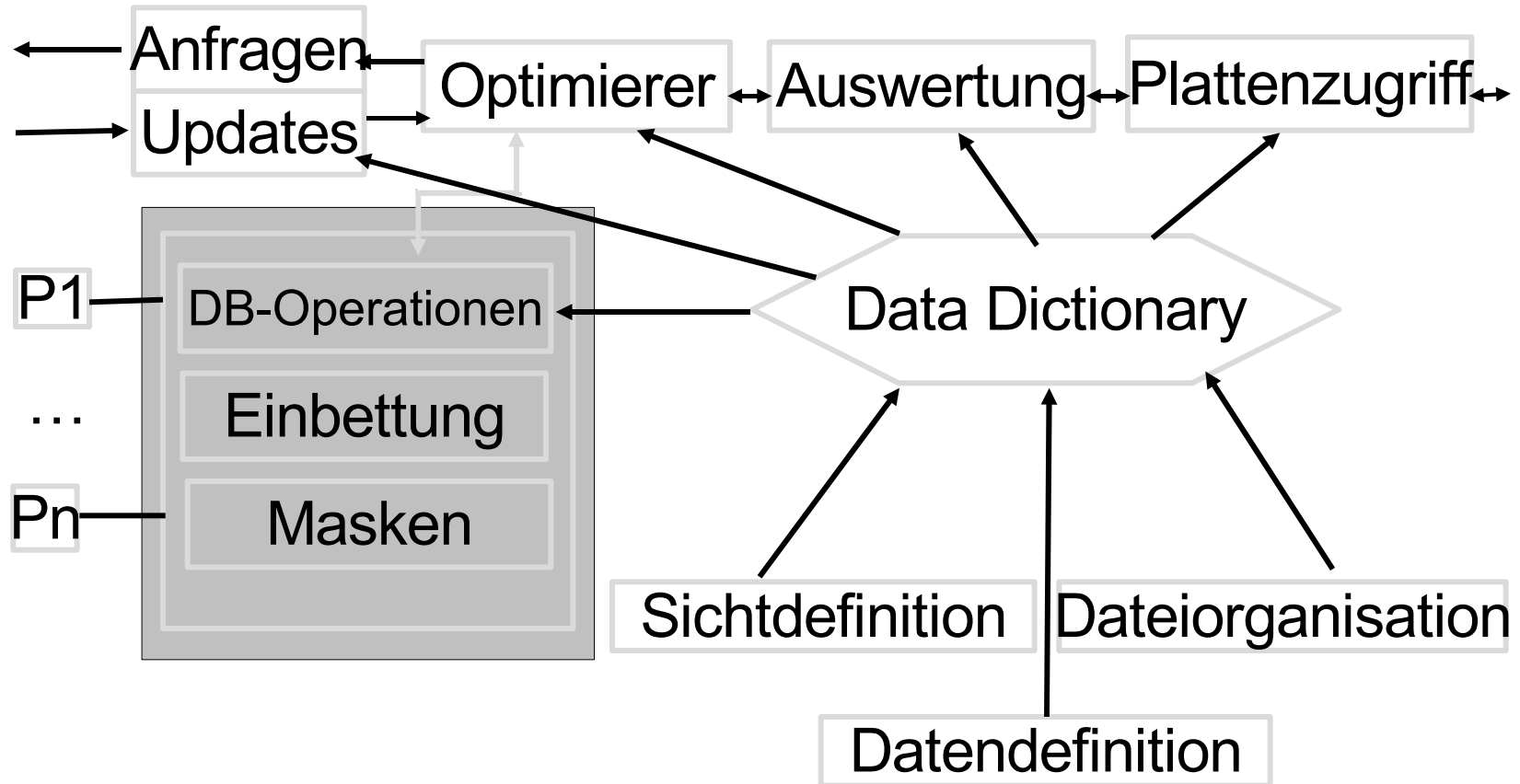
- **Definitionskomponenten:** Datendefinition, Dateiorganisation, Sichtdefinition



Einleitung
Welt ohne DB
Datenbanken Übersicht
Terminologie
Querying
DB – Merkmale
Anwend. Historie
Schema-Arch.
- Einleit.
- [ANSI-SPARC](#)
Schluß

ANSI-SPARC-Architektur (5)

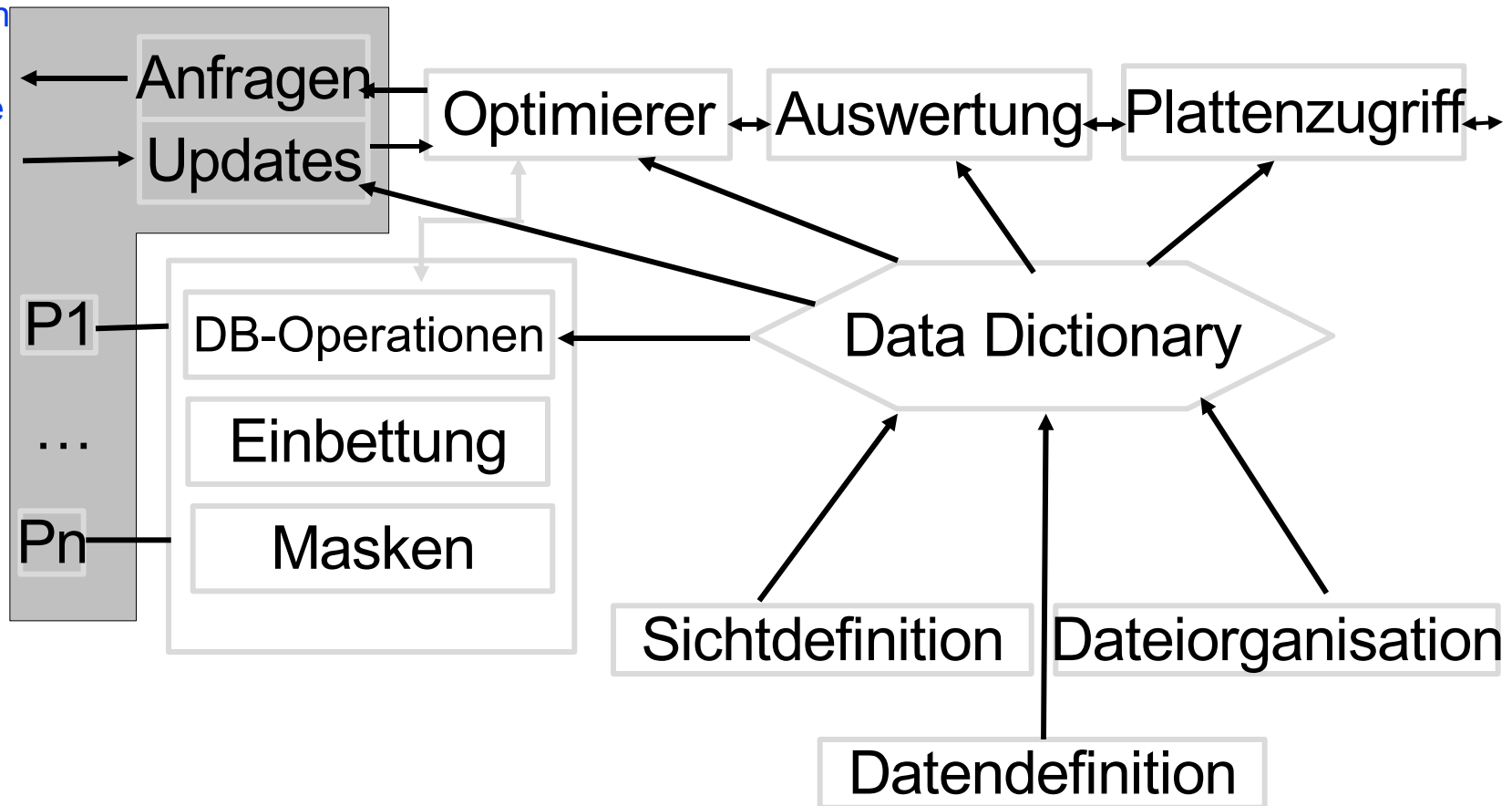
- **Programmierkomponenten:** DB-Programmierung mit eingebetteten DB-Operationen



Einleitung
Welt ohne DB
Datenbanken Übersicht
Terminologie
Querying
DB – Merkmale
Anwend. Historie
Schema-Arch.
- Einleit.
- ANSI-SPARC
Schluß

ANSI-SPARC-Architektur (6)

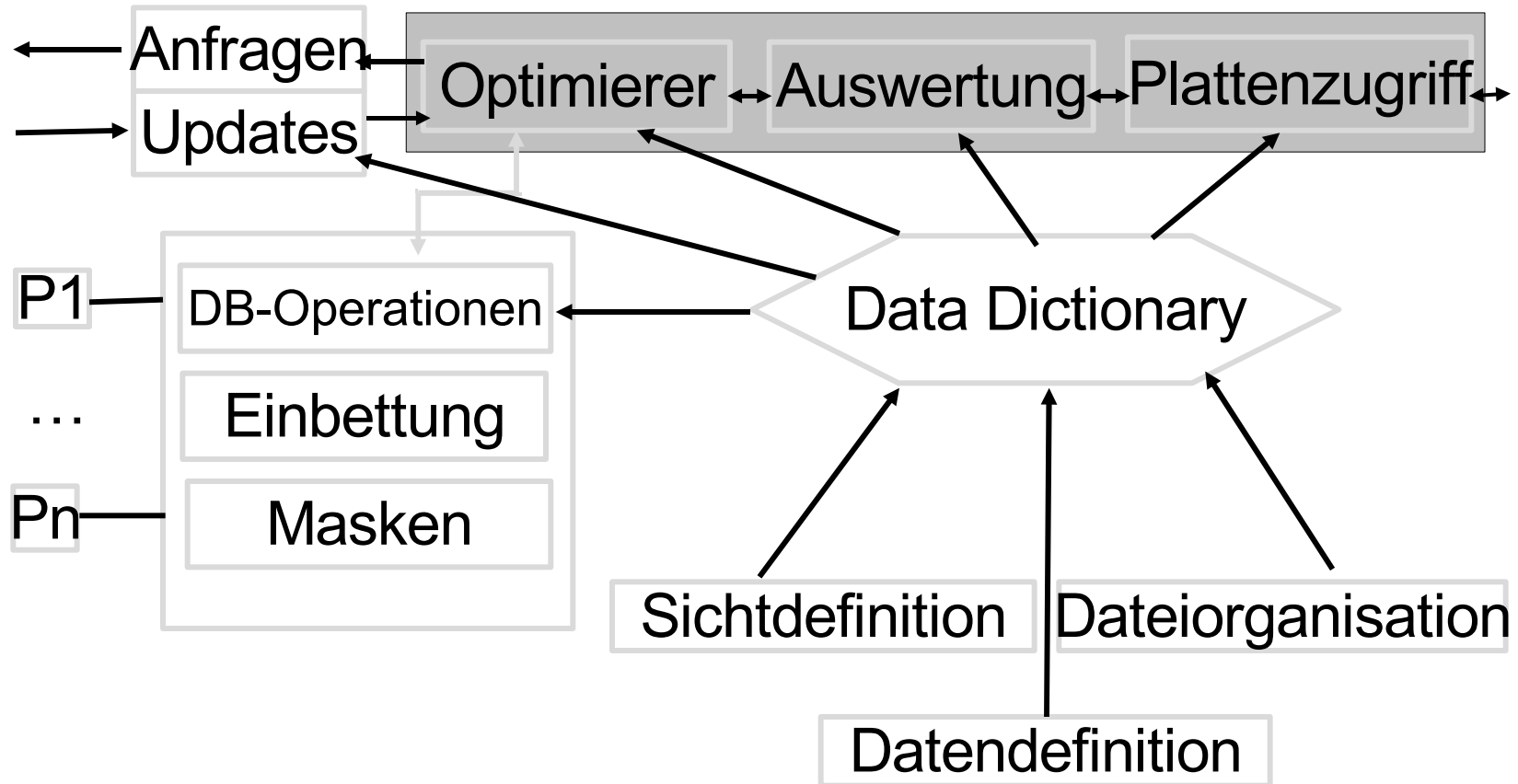
- **Benutzerkomponenten:** Anwendungsprogramme, interaktive Anfragen und Updates



Einleitung
Welt
ohne DB
Datenbanken
Übersicht
Terminologie
Querying
DB –
Merkmale
Anwend.
Historie
Schema-
Arch.
- Einleit.
- [ANSI-
SPARC](#)
Schluß

ANSI-SPARC-Architektur (7)

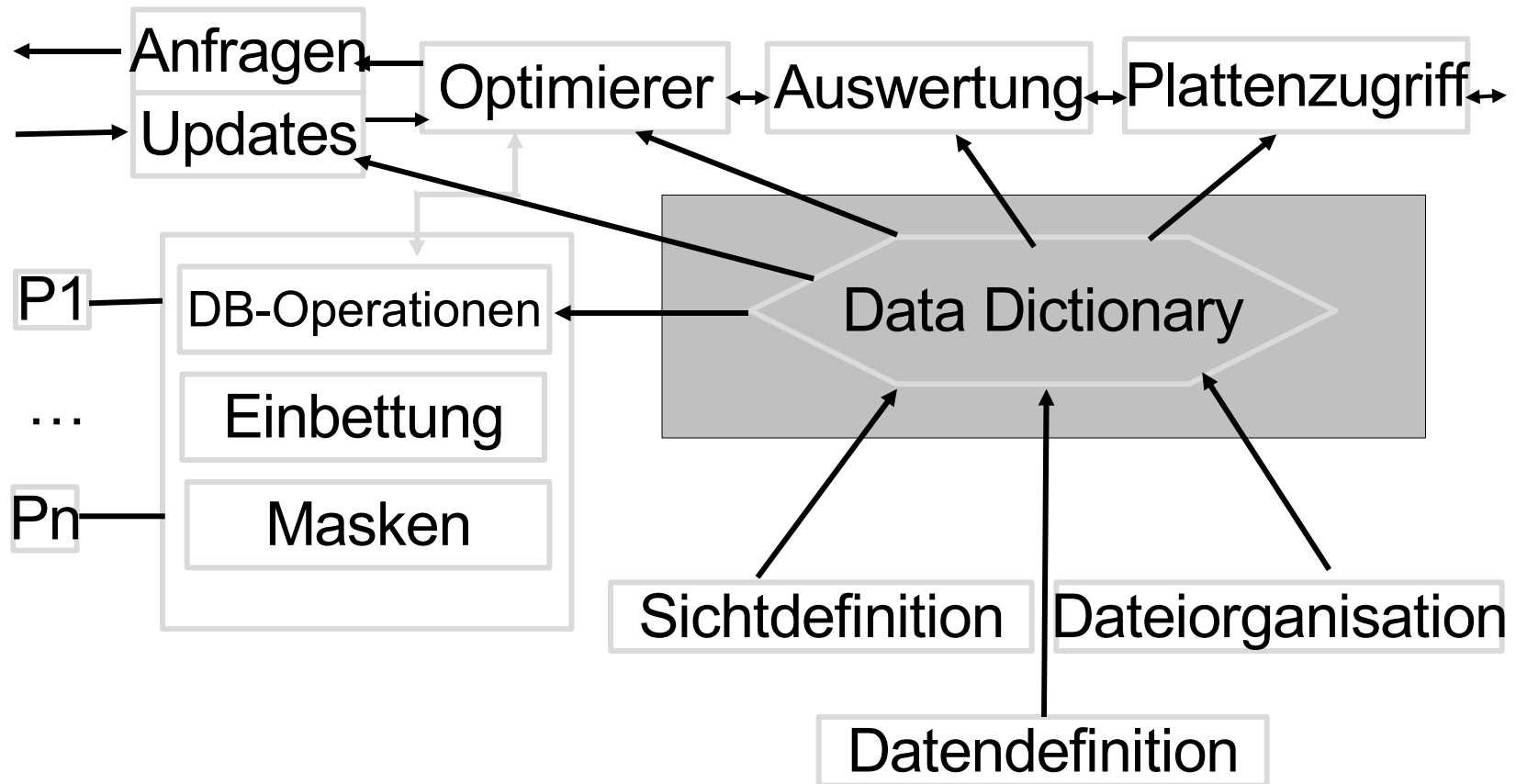
- **Transformationskomponenten:** Optimierer, Auswertung, Plattenzugriff



Einleitung
Welt
ohne DB
Datenbanken
Übersicht
Terminologie
Querying
DB –
Merkmale
Anwend.
Historie
Schema-
Arch.
- Einleit.
- [ANSI-
SPARC](#)
Schluß

ANSI-SPARC-Architektur (8)

- **Data Dictionary:** Verwaltung von Metadaten, Querschnittsfunktion



Einleitung
Welt
ohne DB
Datenbanken
Übersicht
Terminologie
Querying
DB –
Merkmale
Anwend.
Historie
Schema-
Arch.
- Einleit.
- [ANSI-
SPARC](#)
Schluß



Ausblick auf die nächsten Kapitel

Einleitung

Welt
ohne DB

Datenbanken
Übersicht

Terminologie

Querying

DB –
Merkmale

Anwend.

Historie

Schema-
Arch.

Schluß

- Datenbank-Definitionssprachen
- Modelle für den DB-Entwurf
- Relationale Anfragesprachen
- Datenbankentwurf
- Anfrageoptimierung
- Transaktionen in Datenbanken
- Anwendungsprogrammierung

