

1. Die Bedeutung des Datenmanagements
- 2. Datenbank-Architektur**
3. Modellierung und Entwurf von DB-Systemen
4. Relationale Algebra und Normalisierung
5. Definition und Abfrage von Datenbank-Systemen
6. Dateiorganisation und Zugriffsstrukturen
7. Optimierung von Anfragen
8. Transaktionen

### 1. Komponenten von DB-Systemen

1. *Datendefinition und Schema*
2. *Data Dictionary*
3. *Arbeitsweise eines Data Dictionary*
4. *Datenbankmanagementsystem (DBMS)*
5. *Dienstprogramme*
6. *Anwenderprogramme*

### 2. ANSI/SPARC-Architektur

1. *Das Drei-Ebenen-Konzept*
2. *Die Ebene des konzeptionelle (Daten-) Modells / Schemas*
3. *Die Ebene des internen (Daten-) Modells bzw. Schemas*
4. *Die Ebene des externe (Daten-) Modells / Schemas*
5. *Das Ebenen-Konzept zur Daten-Programm-Unabhängigkeit*
6. *Verwaltung der Ebenen durch das DBMS*

### 3. Aufgabenfelder

1. *Zugriffsformen*
2. *Aufgaben Entwickler*
3. *Aufgaben Administrator*
4. *Anwendertätigkeiten*
5. *Einordnung in das Zwiebelschalenmodell*

**Abbildung 2.1:** Schemadiagramm für die Datenbank von Abbildung 1.2.

### STUDENT

Name	StudentNumber	Class	Major
------	---------------	-------	-------

### COURSE

CourseName	CourseNumber	CreditHours	Department
------------	--------------	-------------	------------

### PREREQUISITE

CourseNumber	PrerequisiteNumber
--------------	--------------------

### SECTION

SectionIdentifier	CourseNumber	Semester	Year	Instructor
-------------------	--------------	----------	------	------------

### GRADE\_REPORT

StudentNumber	SectionIdentifier	Grade
---------------	-------------------	-------

Das **Datenbankschema** beschreibt die Struktur der Datenbank.

Quelle: Elmasri, Navathe, S. 42

## 2.1.1 Datendefinition und Schema

**Abbildung 1.2:** Beispiel einer Datenbank, in der Studentendatensätze mit Noten gespeichert werden.

STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	98	King
	92	CS1310	Fall	98	Anderson
	102	CS3320	Spring	99	Knuth
	112	MATH2410	Fall	99	Chang
	119	CS1310	Fall	99	Anderson
	135	CS3380	Fall	99	Stone

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PREREQUISITE	CourseNumber	PrerequisiteNumber
	CS3380	CS3320
	CS3380	MATH2410
	CS3320	CS1310

Die in der Datenbank zu einem bestimmten Zeitpunkt gespeicherten Daten werden als **Datenbankzustand** bezeichnet.

Quelle: Elmasri, Navathe, S. 21

- Die Struktur der aufzubauenden Datenbank muss in einem Datenmodell mit einer Datenbanksprache beschrieben werden.
- Die Beschreibung der verschiedenen konzeptionellen, externen und internen Aspekte erfolgt in entsprechender **DDL (Data Definition / Description Language, siehe später bei SQL)**
- Mit der DDL kann man ohne eine grafische Oberfläche eine relationale Datenbank definieren.

- **Data Dictionary (i.w.S.):**
  - Metadatenbank, in der die Schemata dokumentiert, gespeichert und den Benutzern zur Verfügung gestellt werden und die das vollständige Metawissen (Meta-Daten) über die Daten und das Datenmodell enthält.
  - im Regelfall zentral verwaltet
- **Data Dictionary (i.e.S.) / Datenkatalog:**
  - Eigentliche Beschreibung der Datenbank mit allen Merkmalsdefinitionen, Namen, Feldlängen usw.

## 2.1.2 Data Dictionary

VIEW_NAME	TEXT_LENGTH	TEXT
CUSTORDER	101	select custname,city,orderno,orderdate from customers,orders where customers.custno=orders.custno;



OWNER	TABLE_NAME	TABLE_TYPE
SMITH	ACCOUNT	TABLE
SMITH	CUSTOMERS	TABLE
SMITH	CUSTORDER	VIEW
SMITH	ORDERS	TABLE

### Statistiken

COLUMN_NAME	DATA_TYPE	DATA_LENGTH	NUM_DISTINCT	LOW_VALUE
ORDERNO	NUMBER	22	4	C102
CUSTNO	NUMBER	22	3	C102
ORDERDATE	DATE	7	4	77BF0A1E010101



INDEX_NAME	UNIQUENESS	BLEVEL	LEAF_BLOCKS	DISTINCT_KEYS	AVG_LEAF_BLOCKS_PER_KEY	AVG_DATA_BLOCKS_PER_KEY
ORD_CUSTNO	NONUNIQUE	0	1	3	1	1

externe Ebene

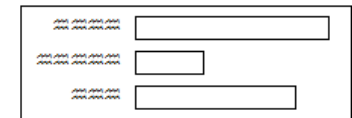


logische Ebene

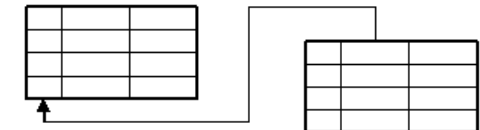


physische Ebene

Sichten (Ein- und Ausgabemasken)



Tabellen und Abhängigkeiten



Konfiguration / DB-Einrichtung

```
<table name="Artikel">
  <size> 5G <\size>
<\table>
```

## 2.1.3 Arbeitsweise eines Data Dictionary



Situation: Der Benutzer PAUL möchte sich alle Datensätze anzeigen lassen, die der folgenden Bedingung genügen:  
GEHALT > 3000



Benutzereingabe:

```
SQLPLUS> SELECT * FROM PERS WHERE GEHALT > 3000;  
/* Dieser Befehl wurde von PAUL eingegeben */
```



**DBMS** überprüft mittels interner SELECT-Befehle das Data-Dictionary u.a. auf folgendes:



## 2.1.3 Arbeitsweise eines Data Dictionary

→ a) Existiert eine Tabelle PERS, auf die der Benutzer PAUL Zugriff hat?

/\* wenn ja, dann b) sonst Fehlermeldung :  
TABLE or VIEW does not exist \*/

→ b) Was bedeutet '\*', aus welchen Spalten besteht die Tabelle PERS?

→ c) Existiert in der Tabelle PERS die Spalte GEHALT?

/\* wenn ja, dann d) sonst Fehlermeldung \*/

→ d) Gibt es einen INDEX für die Tabelle PERS über der Spalte GEHALT?

/\* wenn ja: INDEX-Zugriff sonst: Sequentieller Zugriff \*/

Wenn alle vier internen SELECT-Befehle abgearbeitet sind, stehen dem SYSTEM alle Informationen zur Verfügung, um das Ergebnis des externen, vom Benutzer PAUL eingegebenen Befehls auszuführen.

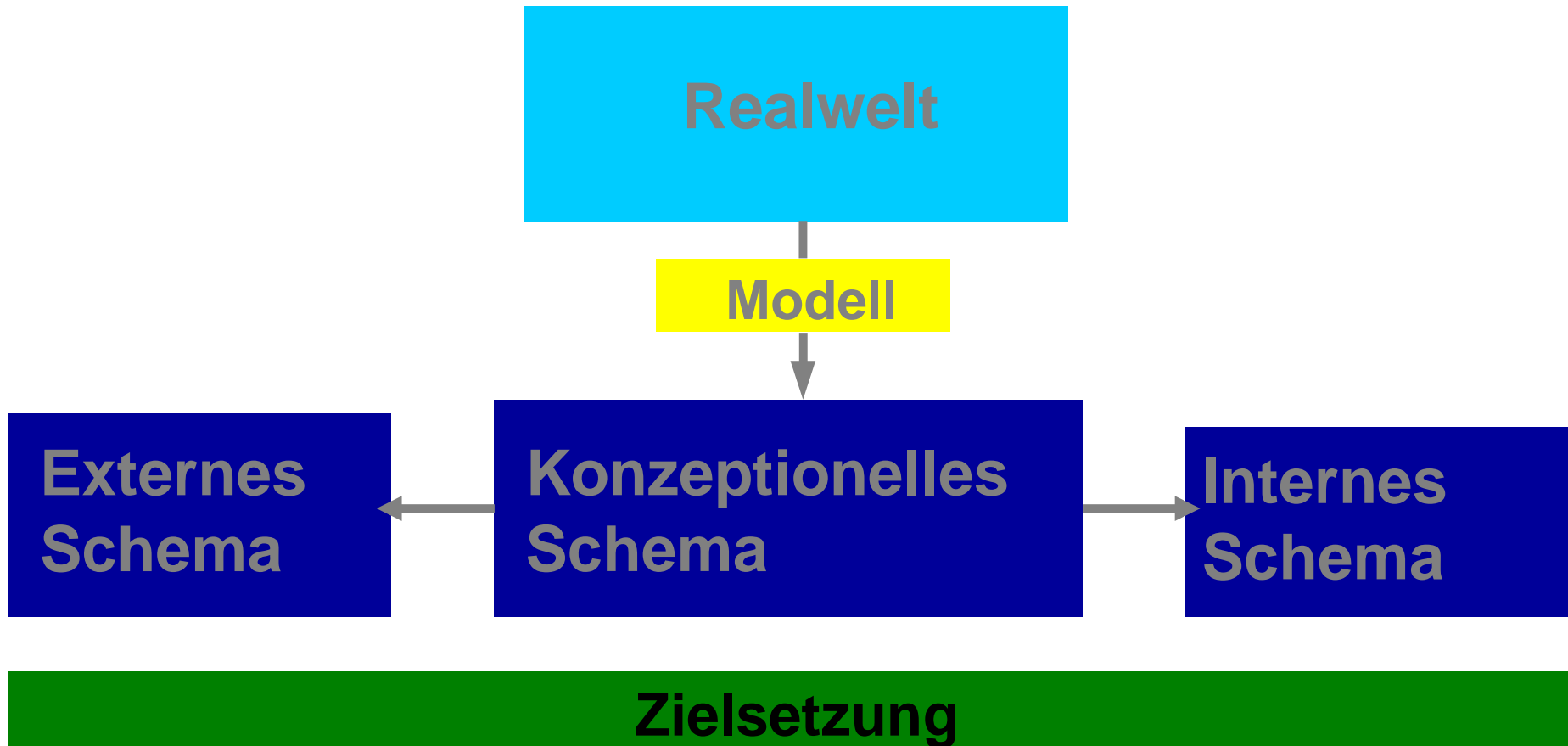
- Das Datenbankmanagementsystem (**DBMS**) ist gleichbedeutend mit Datenbankverwaltungssystem (DBVS).
- Es ist eine Software, die für den Betrieb des Datenbanksystems installiert und konfiguriert wird.
- Es legt das Datenbankmodell fest, speichert die Daten und stellt den Zugang zu diesen her.
- Das DBMS umfasst ein Paket von Systemroutinen für die üblichen Funktionen auf Datenbanken (*Schreiben, Suchen, Lesen, Löschen*), die den übersetzten Anwenderprogrammen zur Verfügung stehen.
- Es kann über Datenbanksprache zu Handlungen veranlasst werden.
- Es beinhaltet Anfrageoptimierungen, z.B. durch Indizes, und Anwendungsunterstützung, durch Bereitstellung von Triggern und Stored Procedures.

- Es verhindert ungewollte Redundanzen.
- Es stellt die Datenintegrität durch Constraints sicher. Constraints sind Regeln, die festlegen, wie Daten verändert werden dürfen, z.B. Foreign Key Constraint. **Fallen Ihnen noch weitere Anwendungen von Constraints ein?**
- Zuständig für die Verwaltung der Metadaten (Data Dictionary) und den Schutz gegen Datenverlust unerlaubten Zugriff.
- Mehrbenutzerbetrieb durch Transaktionskonzept: Transaktionen sperren Daten vorübergehend für den Zugriff durch andere Benutzer, bis eine Transaktion durch ein sinngemäßes “Commit” beendet wird. Danach werden die geänderten Daten wieder freigegeben.

Dienstprogramme sind System- oder Hilfsprogramme (engl. Utilities) zur Ausführung allgemeiner, systemnaher Aufgaben:

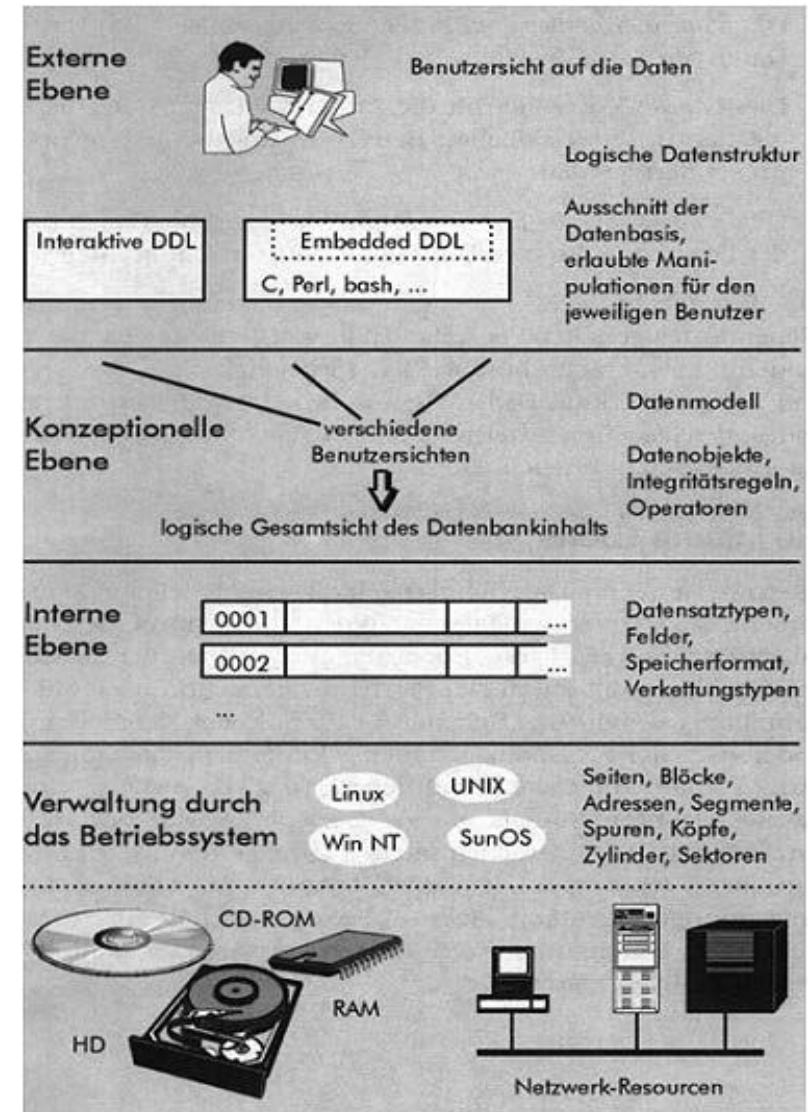
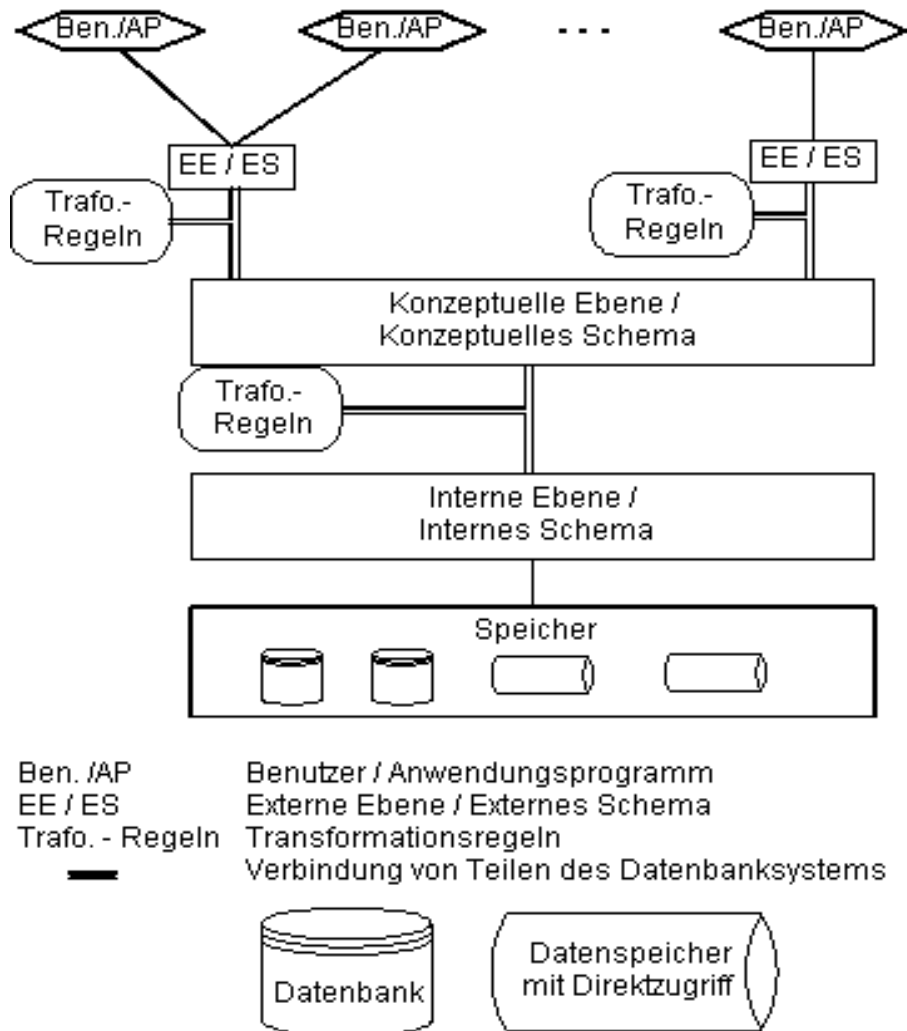
- Hilfsfunktionen beim Betrieb der Datenbank, besonders bei Reorganisation oder zur Vorbereitung und Ausführung einer Rekonstruktion im Schadensfall.
- Programme, um den gesamten Inhalt der Datenbank (inklusive Datenkataloge) auf Archivdatenträger zu schreiben und von dort wieder einzulesen.
- Bereitstellung von Betriebsstatistiken, die hinsichtlich der effektiven Nutzung des Datenbank-Systems aussagekräftig sind.

- Sogenannte „Front-Ends“ des Datenbank-Systems, die in einer DML (Data Manipulation Language) geschrieben werden und gegenüber herkömmlichen Programmen zusätzliche Lese- und Schreibbefehle als Verbindung zur Datenbank beinhalten.
- Die DML ist entweder eine selbständige Datenbank-Sprache oder eine höhere Programmiersprache, welche um spezielle Datenmanipulations-Prozeduren und/oder geeignete Sprachkonstrukte erweitert worden ist.
- Die Mehrheit der Benutzer verwendet übersetzte Anwenderprogramme, die den Umgang mit der Datenbank erleichtern sollen (z.B. Masken- und Report-Generatoren). Die auf Tool-Ebene getätigten Eingaben werden dabei in Befehle der Datenbank-Sprache umgesetzt.



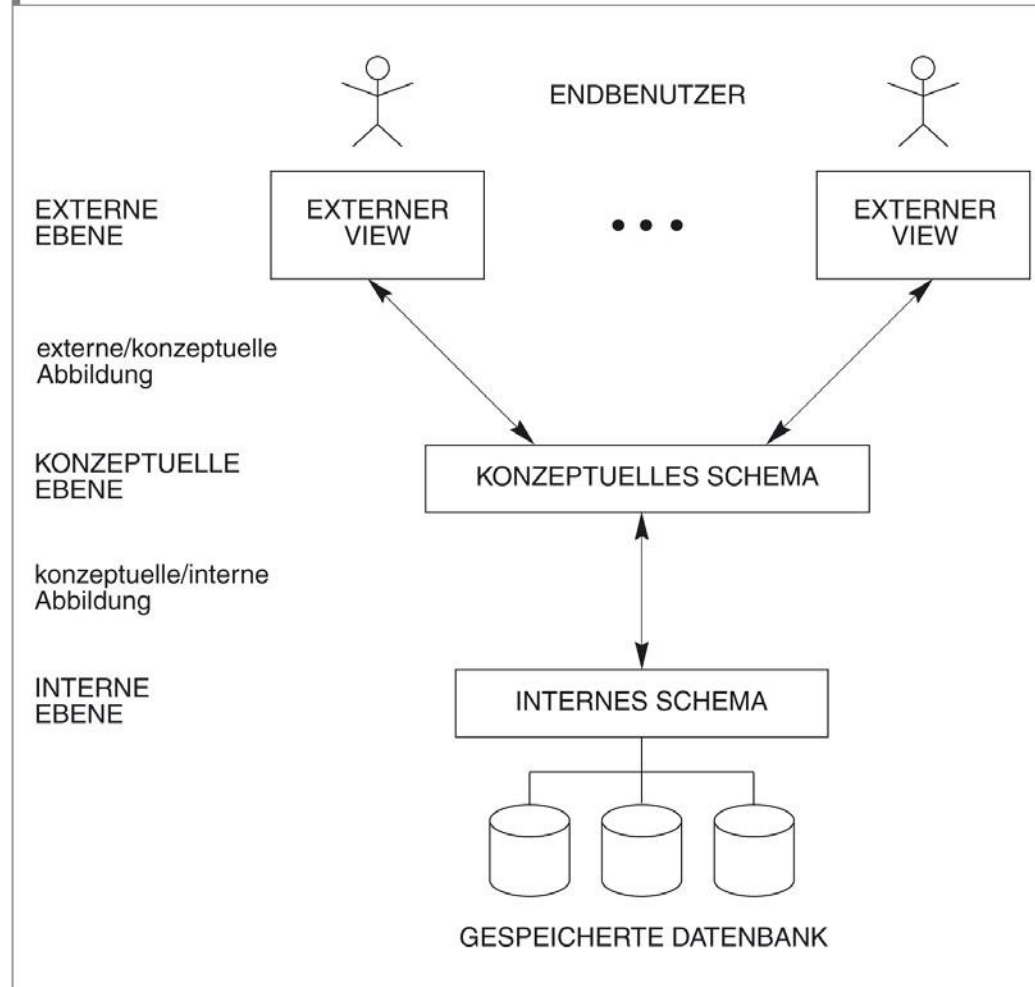
## 2.2.1 Das Drei-Ebenen-Konzept

3-Ebenen-Architektur nach ANSI/SPARC



## 2.2.1 Das Drei-Ebenen-Konzept

**Abbildung 2.2:** Darstellung der Drei-Schichten-Architektur.



Quelle: Elmasri, Navathe, S. 44



- **Konzeptionelles Modell**

- Beschreibung des Gesamtangebots aller Daten auf logischer Ebene (d.h. völlig unabhängig von Datenverarbeitungs-Gesichtspunkten):
  - Alle Daten
  - Beziehungen der Daten untereinander
- Designerziel ist eine vollständige und redundanzfreie Darstellung aller zu speichernden Informationen. Hier findet die Normalisierung des relationalen Datenbankschemas statt.
- stellt die „gemeinsame sprachliche Basis“ für die Kommunikation der an der Organisation von Datenverarbeitungsabläufen beteiligten Personen dar.

- Zuständigkeit für diese Ebene liegt beim Unternehmens-Administrator (Enterprise Administrator).
- Sie bildet die Grundlage für die Ableitung der bei der Datenspeicherung verwendeten physischen Datenstrukturen, dem Internen Modell sowie des dem Anwender bereitgestellten externen Modells.
- Sie wird mit Hilfe einer geeigneten Datenbeschreibungssprache (DDL) beschrieben; das Ergebnis heißt dann konzeptionelles Schema.
- Sie ändert sich in der Regel wesentlich langsamer als die Anforderungen an einzelne Anwendungen (d.h. die Programme) und stellt somit einen relativ stabilen Beziehungspunkt im Unternehmensmodell dar.

- **Internes Modell (auch physisches Modell)**
  - Sie stellt die physische Sicht der Datenbank dar.
  - Sie ist die Beschreibung, **wie** und **wo** die Daten in der Datenbank gespeichert werden (Aufbau der abgespeicherten Daten, deren Speicherorganisation, die Zugriffspfade, usw.).
  - Ihr Designziel ist der performante Zugriff auf die gespeicherten Daten, was auch durch eine bewusst in Kauf genommene Redundanz erreicht wird.
  - Zur Optimierung des Internen Modells werden z.B. statistische Infos über die Häufigkeit von Abfragen, von Zugriffen auf Tabellen etc. vom Datenbankadministrator ausgewertet.
  - Die Zuständigkeit für diese Ebene liegt beim DB-Administrator (Data Base Administrator).

- **Externes Modell**

- Beschreibung, welche der gespeicherten Daten ein Anwender/ Programmierer benötigt. Es gibt i. d. R. viele Externe Modelle / Schemata
- jedem Benutzer kann auf diese Weise ein ganz bestimmter Teil der Daten zur Verfügung gestellt werden („Filter“ über die physischen Dateien der Datenbank)
- stellt den Anwendungen und Benutzern individuelle Benutzersichten bereit (z.B. Formulare, Masken-Layouts, Listen, Schnittstellen)
- Auch als „Views“, „Benutzersichten“ oder „logische Dateien“ bezeichnet; im Zusammenhang mit Externem Schema wird auch von „Subschema“ gesprochen.

## 2.2.4 Die Ebene des externen (Daten-) Modells bzw. Schemas

- Der Umgang mit den Daten durch den Benutzer geschieht mit einer Datenmanipulationssprache (DML) oder Abfragesprache (QL: Query Language). Am verbreitetsten ist hier SQL.
- Die „Logische Datei“ bringt zum Ausdruck, dass ein Externes Modell keine „physische Daten-Datei“ darstellt. Hierbei können „neu errechnete“ Attribute (wie z.B. UMSATZ aus den physischen Feldern „PREIS“ und „MENGE“) angeboten werden, die in den Originaldaten gar nicht vorhanden sind.
- Die Zuständigkeit für diese Ebene liegt beim Anwendungs-Administrator (Application Administrator).

# Barrieren in Datenbank-Systemen gibt es zwischen:

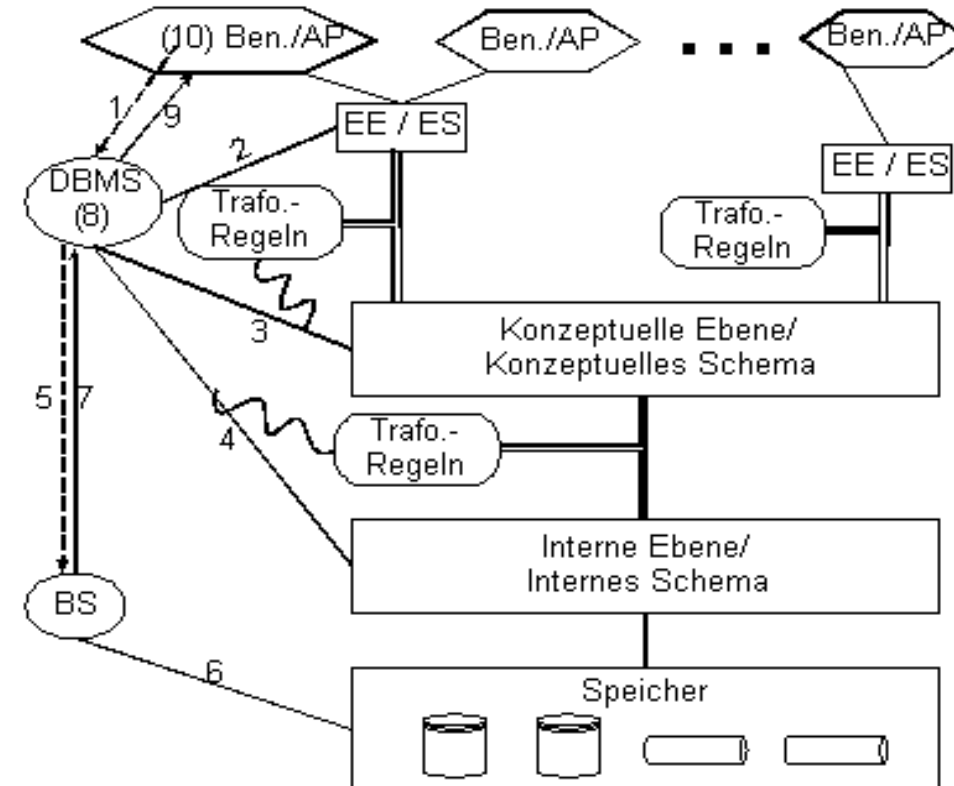
- ... der physikalischen Datenorganisation (internes Modell / Schema) und der globalen, konzeptionellen (logischen) Datenorganisation (konzeptionelles Modell / Schema).
- ... dem konzeptionellen Modell / Schema und der Sicht eines Anwenderprogramms auf die Daten (externes Modell / Schema).
- ... dem externen Modell eines beliebigen Anwenderprogramms x und dem externen Modell eines beliebigen anderen Anwenderprogramms.

### Gründe der vertikalen Trennung:

- Physische Datenunabhängigkeit durch Trennung der internen von der konzeptionellen. Physische Änderungen, z.B. des Speichermediums oder des Datenbankprodukts, wirken sich nicht auf die konzeptionelle Ebene aus.
- Logische Datenunabhängigkeit durch Trennung der konzeptionellen und der externen Ebene. Änderungen an der Datenbankstruktur (konzeptionelle Ebene) haben keine Auswirkungen auf die externe Ebene, also auf Masken- Layouts, Listen und Schnittstellen.

## 2.2.6 Verwaltung der Ebenen durch das DBMS

**ANSI/SPARC: Verwaltung der drei Ebenen**



DBMS      Datenbankmanagementsystem  
BS        Betriebssystem  
----->    Befehlsweitergabe  
----->    Daten- und Kontrollübergabe  
=====>    Ausführung einer Verarbeitung

Ben. /AP  
EE / ES  
Trafo. - Regeln

Benutzer / Anwendungsprogramm  
Externe Ebene / Externes Schema  
Transformationsregeln



1. Das DBMS empfängt den Befehl des Programms, einen Datensatz eines externen Modells zu lesen (z.B. MITARBEITER mit NAME = Müller).
2. Das DBMS besorgt die benötigten Definitionen des entsprechenden Datentyps aus dem externen Schema, welches das Programm benutzt (MITARBEITER: NAME, PERS, WOHNORT).
3. Das DBMS besorgt die entsprechenden Teile des konzeptionellen Schemas (unter Ausnutzung der Transformationsregeln) und stellt fest, welche konzeptionellen Datentypen und Beziehungen benötigt werden.
4. Das DBMS besorgt die benötigten Teile des internen Schemas (unter Ausnutzung der Transformationsregeln) und stellt fest, welche physischen Sätze zu lesen sind. Es bestimmt gegebenenfalls die auszunutzenden Zugriffspfade.

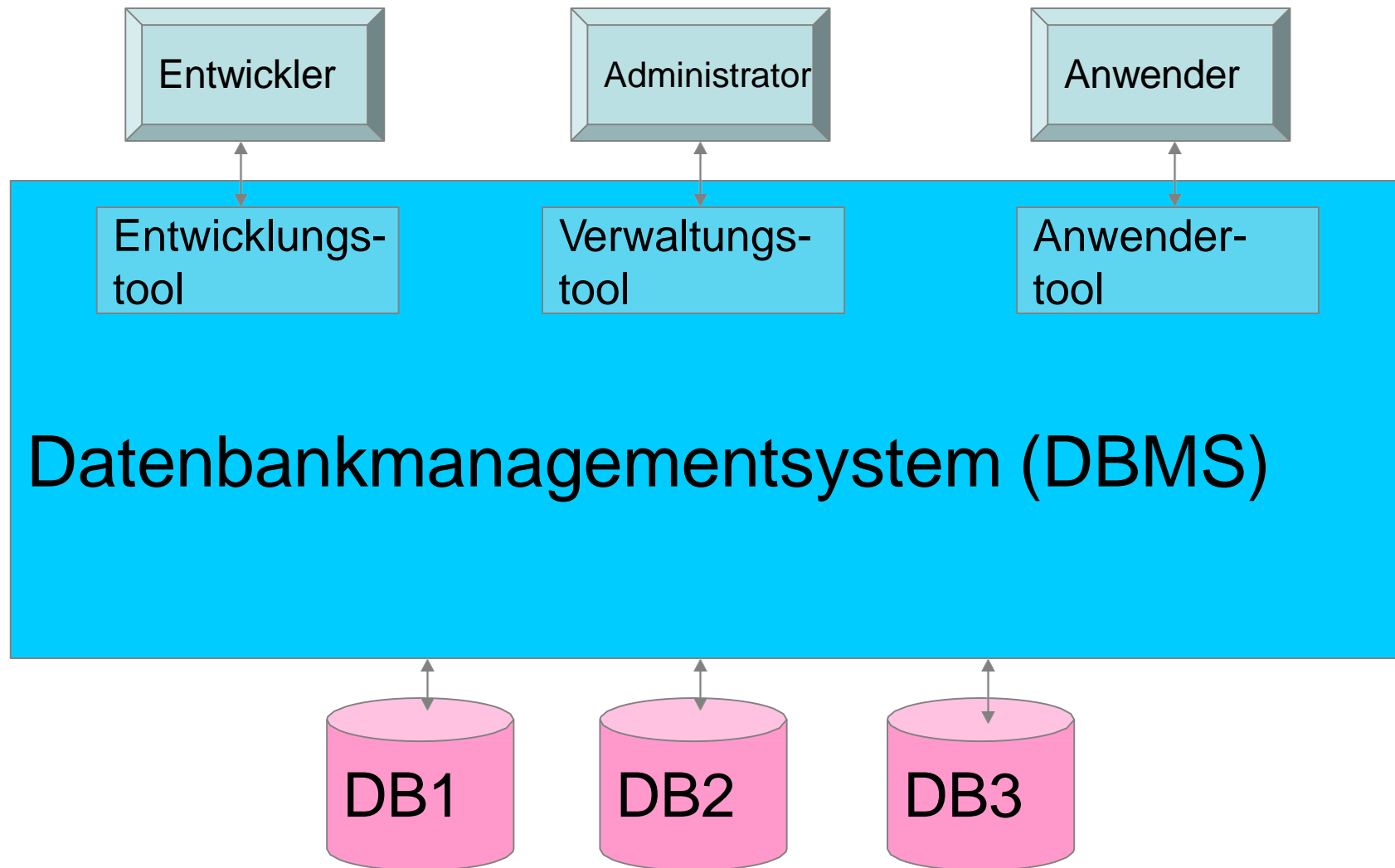
5. Das DBMS übergibt dem Betriebssystem die Nummern der zu lesenden Speicherblöcke.
6. Das Betriebssystem liest die angeforderten Blöcke aus dem Speicher.
7. Das Betriebssystem übergibt die verlangten Blöcke dem DBMS in einem Systempuffer.
8. Aufgrund der Transformationsregeln konzeptionelles/internes Modell und externes/konzeptionelles Modell stellt das DBMS aus den vorhandenen physischen Sätzen den verlangten externen Datensatz zusammen (Konvertierungen, Formatierungen).
9. Das DBMS übergibt den externen Datensatz dem Anwendungsprogramm in einem Kommunikationsbereich.
10. Das Anwendungsprogramm verarbeitet die vom DBMS übergebenen Daten.

## 2.3 Aufgabenfelder

Welche Aufgabenfelder können Sie sich im Zusammenhang mit einem DBMS vorstellen?

Denken Sie dabei an Personen, die mit dem System arbeiten und ihre möglichen Rollen. Wer ist das bei Ihnen in der Firma?

## 2.3.1 Zugriffsformen

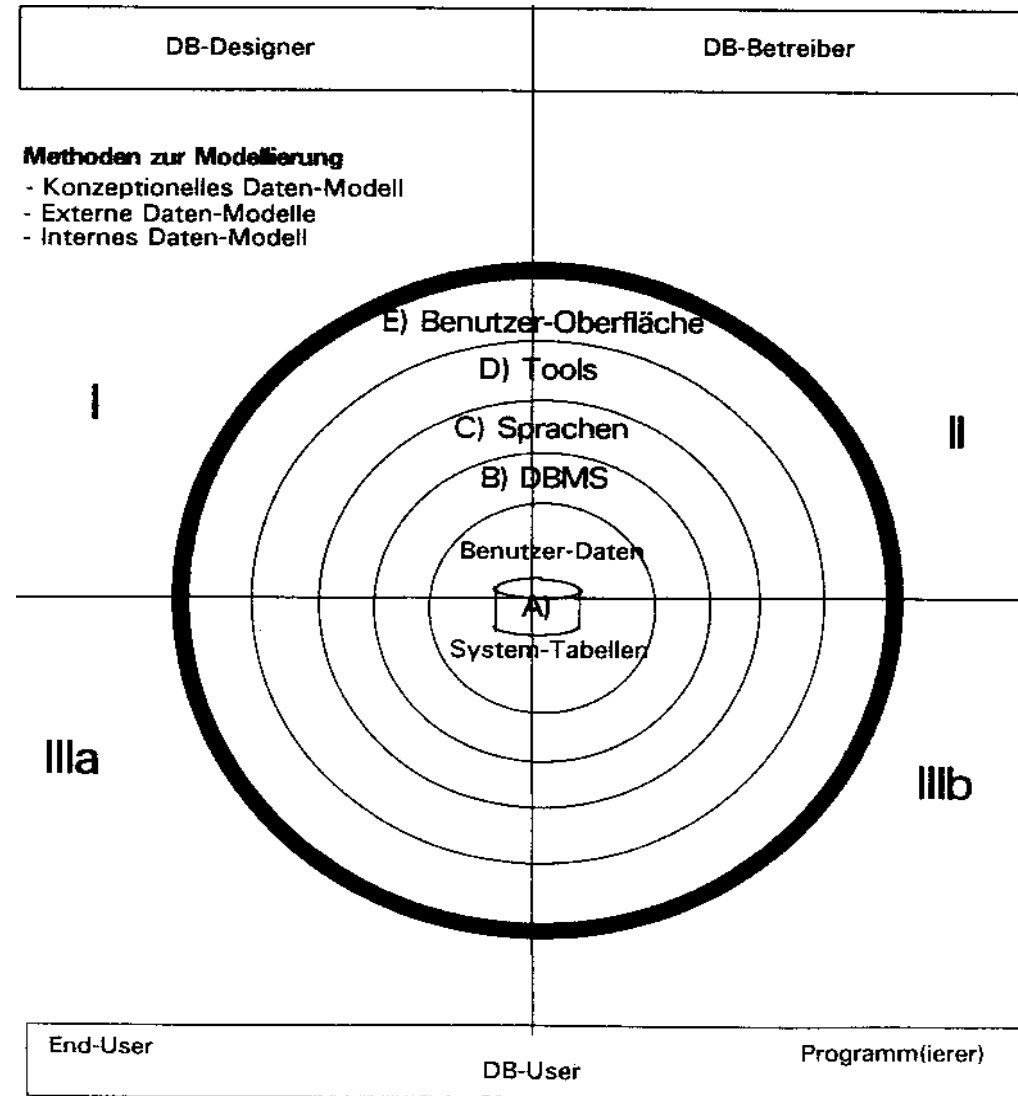


- **Entwickler entwerfen und erstellen eine Datenbank**
  - vom konzeptionellen Modell bis zum konzeptionellen Schema
  - vom internen Modell zum internen Schema
  - von den externen Modellen zu den externen Schemata
- **Datenmodellierung**
- **Konsistenzsicherung der Datenbasis**
- **DB anlegen, Tabellen mit Verknüpfungen definieren**
- **Anwendungsentwicklung**
  - Ansichten generieren
  - Formulare / Masken entwerfen
  - Standardberichte erstellen

- Implementation DBMS
- Datensicherung (Back-up)
- Benutzerverwaltung
- Tuning (Performance-Optimierung)
- Replikation im Rahmen verteilter DBs
- Gewährleistung des effizienten, laufenden Betriebes der DB
- Überwachung des Betriebs (Monitoring, Auditing)
- Existiert ein Gesamtverantwortlicher, der Aufgaben und Personen koordiniert, so wird dieser als Database oder Enterprise Administrator bezeichnet.

- Daten suchen und lesen
- Datenpflege (Aktualisierung der Daten)
  - neue Daten eingeben
  - vorhandene Daten verändern
  - vorhandene Daten löschen
- User: i.d.R. mit wenig Datenverarbeitungskenntnissen
- Programmierer (bzw. Anwendungsprogrammierer)
- Der Zugriff erfolgt über externe Modelle, die in ihrer Gesamtheit möglichst durch einen Application Administrator verantwortet werden.

## 2.3.5 Einordnung in das Zwiebelschalenmodell





**Vielen Dank für Ihre Aufmerksamkeit  
bisher. Wir befinden uns am Ende des  
zweiten Kapitels!**