2. Architekturen von DBS

- Schema-Architektur
- System-Architekturen
- Konkrete System-Architekturen
- Anwendungsarchitekturen

Schema-Architektur I

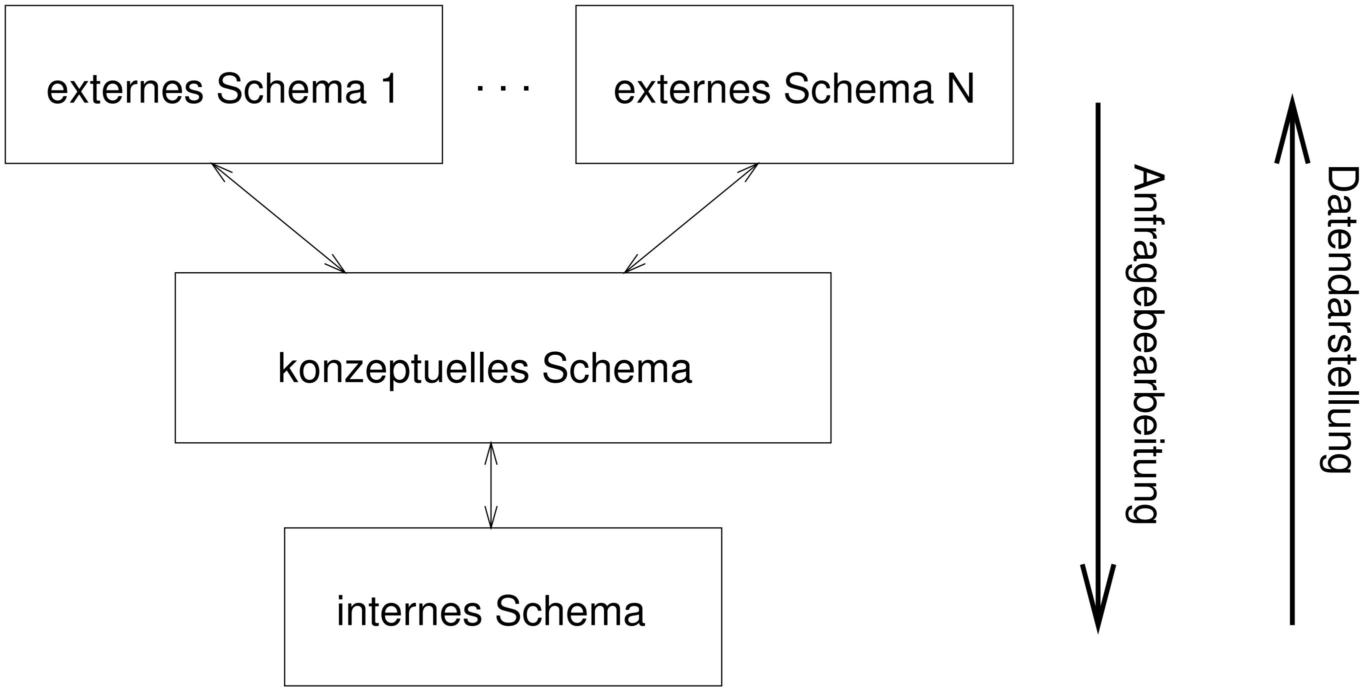
Zusammenhang zwischen

- Konzeptuellen Schema (Ergebnis der Datendefinition)
- Internen Schema (Festlegung der Dateiorganisationen und Zugriffspfade)
- Externen Schema (Ergebnis der Sichtdefinition)
- Anwendungsprogrammen (Ergebnis der Anwendungsprogrammierung)

Schema-Architektur II

- Trennung Schema — Instanz
  - Schema (Metadaten, Datenbeschreibungen)
  - Instanz (Anwenderdaten, Datenbankzustand oder -ausprägung)
- Datenbankschema besteht aus
  - internem, konzeptuellen, externen Schema und den Anwendungsprogrammen
  - im konzeptuellen Schema etwa:
    - Strukturbeschreibungen
    - Integritätsbedingungen
    - Autorisierungsregeln (pro Benutzer für erlaubte DB-Zugriffe)

Schema-Architektur III



Datenunabhängigkeit

- Stabilität der Benutzerschnittstelle gegen Änderungen
- physisch: Änderungen der Dateiorganisationen und Zugriffspfaden haben keinen Einfluß auf das konzeptuelle Schema
- logisch: Änderungen am konzeptuellen und gewissen externen Schemata haben keine Auswirkungen auf andere externe Schemata und Anwendungsprogramme
  - eventuell aber externe Schemata betroffen (Ändern von Attributen)
  - eventuell aber Anwendungsprogramme betroffen (Rekompilieren der Anwendungsprogramme, eventuell Änderungen nötig)
  - nötige Änderungen werden jedoch vom DBMS erkannt und überwacht

Ebenen-Architektur am Beispiel: Konzeptuelle Sicht

- Relationale Darstellung

AUTOR

Name	<u>Nr</u>	<u>BuchID</u> → BUCH.BuchID
Meier	1	4242
Schulze	2	3745
Ibsen	1	3745
...	...	....

BUCH

<u>BuchID</u>	Titel	Jahr	ISBN
4242	Datenbasen I	1993	3-452-12
3745	UNIX X	1998	1-424-11
...	...	...	...

Ebenen-Architektur am Beispiel: Externe Sicht I

- Daten in einer flachen Relation

TITEL

Name	Nr	Titel	Jahr	ISBN
Meier	1	Datenbasen I	1993	1-424-11
Schulze	2	UNIX X	1998	3-452-12
Ibsen	1	UNIX X	1998	3-452-12
...	...	...	...	...

Ebenen-Architektur am Beispiel: Externe Sicht II

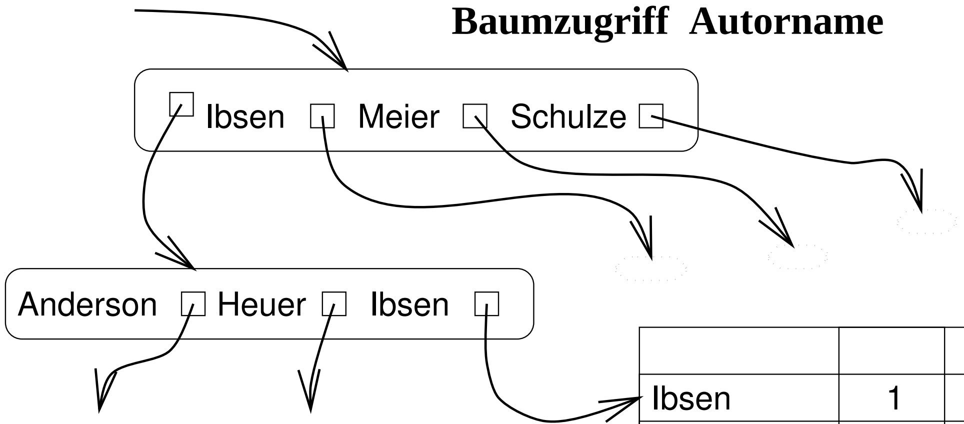
- Daten in einer hierarchisch aufgebauten Relation

TITEL

Autoren { Autor }	Titel	Jahr	ISBN
Meier	Datenbasen I	1993	1-424-11
Ibsen Schulze	UNIX X	1998	3-452-12
...	...	...	...

Ebenen-Architektur am Beispiel: Interne Darstellung

Baumzugriff Autorname



Ibsen	1	*
Jagellovsk	1	*
DeMonti	4	*
....	..	*

Hash-Tabelle Buchtitel

...	...
101101	-
101110	UNIX X
101111	-
110000	Datenbasen 1
110001	MZ4 antwortet nicht
...	...

System-Architekturen

- Beschreibung der Komponenten eines Datenbanksystems
- Standardisierung der Schnittstellen zwischen Komponenten

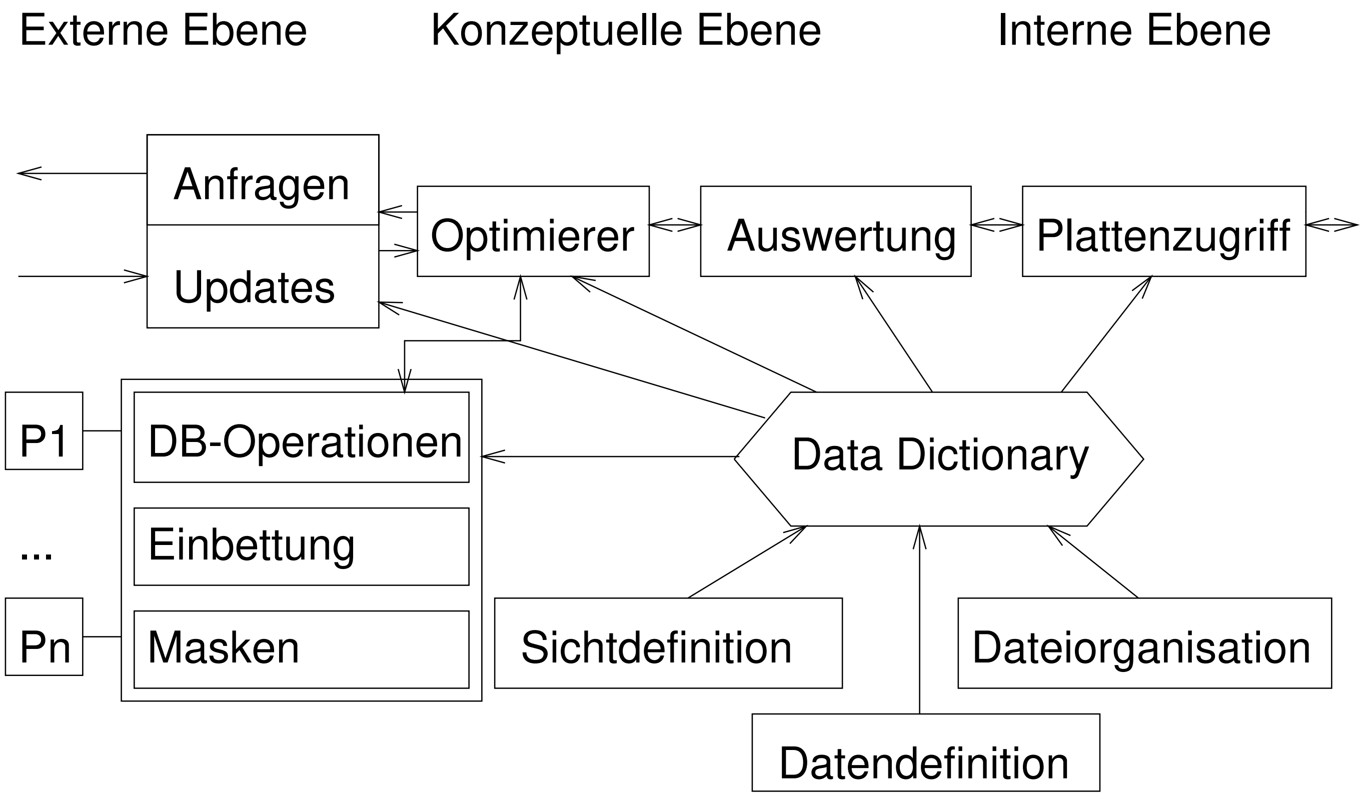
Architekturvorschläge

- ANSI-SPARC-Architektur
  → Drei-Ebenen-Architektur
- Fünf-Schichten-Architektur
  → beschreibt Transformationskomponenten

ANSI-SPARC-Architektur I

- 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 II



Klassifizierung der Komponenten

- Definitionskomponenten: Datendefinition, Dateiorganisation, Sichtdefinition
- Programmier-Komponenten: DB-Programmierung mit eingebetteten DB-Operationen
- Benutzerkomponenten: Anwendungsprogramme, Anfrage und Update interaktiv
- Transformationskomponenten: Optimierer, Auswertung, Plattenzugriffssteuerung
- Data Dictionary (Datenwörterbuch): Aufnahme der Daten aus Definitionskomponenten, Versorgung der anderen Komponenten

Fünf-Schichten-Architektur

- basierend auf Idee von Senko 1973
- Weiterentwicklung von Härder 1987
- Umsetzung im Rahmen des IBM-Prototyps *System R*
- genauere Beschreibung der Transformationskomponenten
  - schrittweise Transformation von Anfragen/Änderungen bis hin zu Zugriffen auf Speichermedien
  - Definition der Schnittstellen zwischen Komponenten

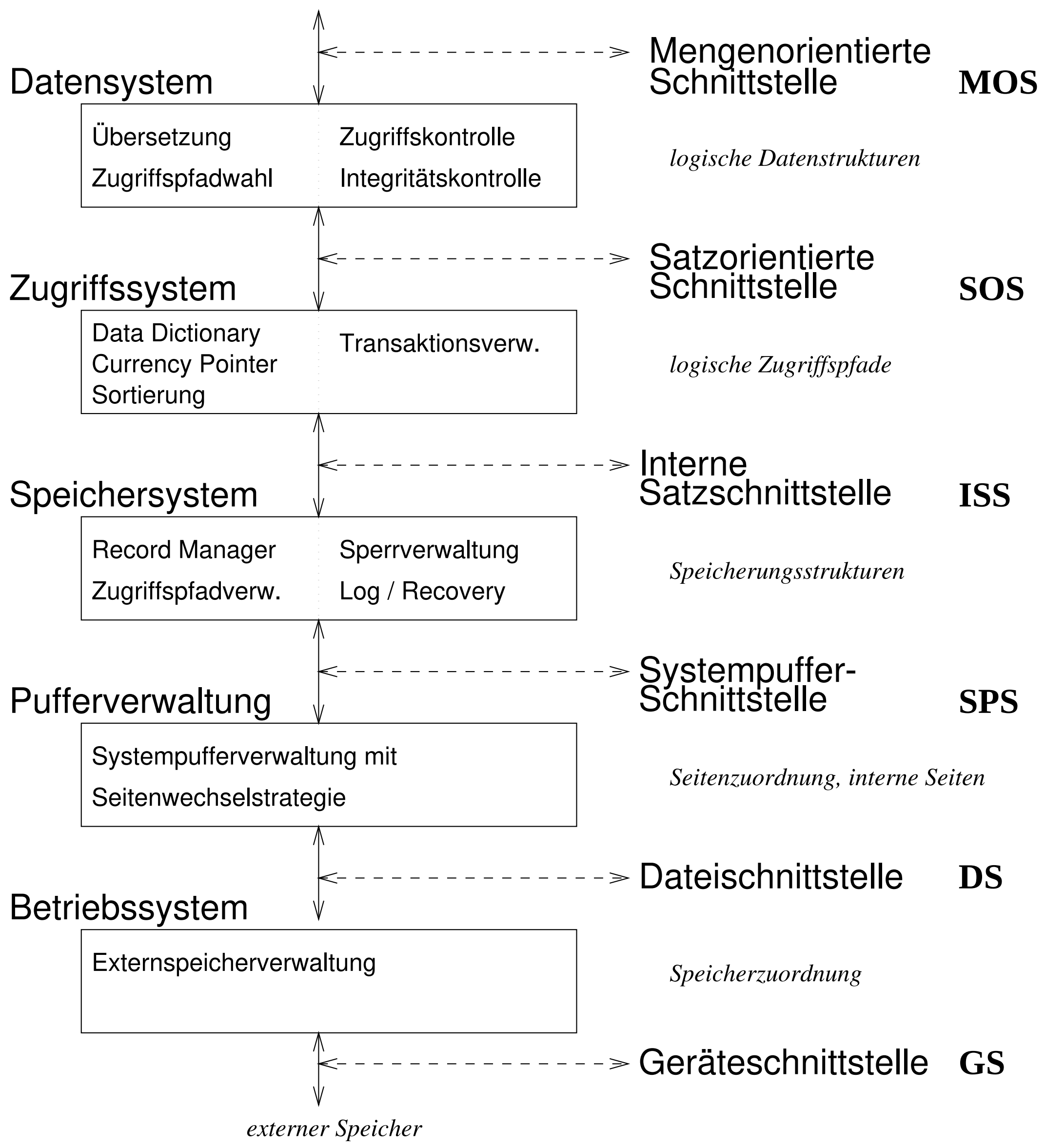
Fünf-Schichten-Architektur: Schnittstellen I

- mengenorientierte Schnittstelle:
  - deklarative DML auf Tabellen, Sichten, Zeilen
- satzorientierte Schnittstelle:
  - Sätze, logische Dateien, logische Zugriffspfade
  - navigierender Zugriff
- interne Satzschnittstelle
  - Sätze, Zugriffspfade
  - Manipulation von Sätzen und Zugriffspfaden

Fünf-Schichten-Architektur: Schnittstellen I

- Pufferschnittstelle
  - Seiten, Seitenadressen
  - Freigeben und Bereitstellen
- Datei- oder Seitenschnittstelle
  - Hole Seite, Schreibe Seite
- Geräteschnittstelle
  - Spuren, Zylinder
  - Armbewegungen

Fünf-Schichten-Architektur: Funktionen



Fünf-Schichten-Architektur: Objekte, Operationen

Objekte	Schicht / Schnittstelle	Operationen
Relationen *Sichten*	**MOS**	*SQL : select ... from ...* *QBE, QUEL, ...*
	Datensystem	
externe Sätze *Index-Strukturen* *Scans*	**SOS**	*FIND NEXT satz* *STORE satz*
	Zugriffssystem	
interne Sätze *Bäume* *Hash-Tabellen*	**ISS**	*Speichere internen Satz s* *INSERT in B-Baum*
	Speichersystem	
Segmente *Seiten*	**SPS**	*Bereitstellen Seite j* *Freigeben Seite j*
	Pufferverwaltung	
Dateien *Blöcke*	**DS**	*Lies Block k* *Schreibe Block k*
	Betriebssystem	
Zylinder *Spuren*	**GS**	*Treiber*

Einige konkrete Systeme

Architekturüberblicke über

- IMS (hierarchisch)
- UDS (Netzwerk)
- relationale Systeme
  - DB2
  - SQL/DS
  - ORACLE
  - INGRES
  - dBASE
  - MS-Access

IMS

IMS $\cong$ Information Management System (IBM, Mitte der 60er)

Versionen

- IMS/360
- IMS/VS (Virtual Storage)
  - IMS/VS/DB (Data Base); nur batch-orientiert
  - IMS/VS/DC (Data Communication); auch dialogorientiert
- unter Betriebssystem DOS/VSE

IMS II

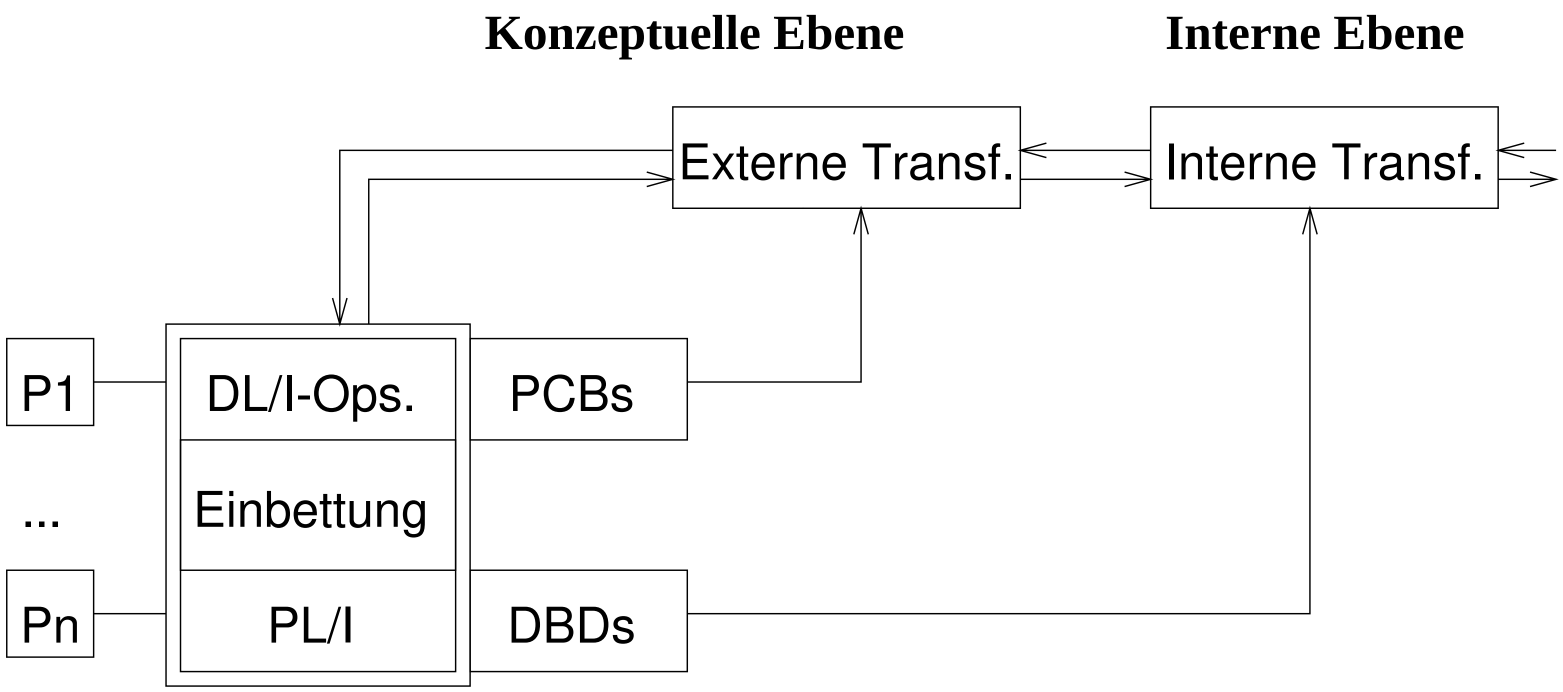
- Begriffseinordnung

Datenbank	$\equiv$	interne (physische) Datenbanken
internes Schema	$\equiv$	DBD (Data Base Description) für jede Datenbank
Sicht	$\equiv$	Menge logischer Datenbanken
Sichtdefinition	$\equiv$	PSB (Program Specification Block), bestehend aus mehreren PCBs (Program Communication Blocks)

  Alles im Zugriff des Anwendungsprogrammierers

- Sprache: DL/I (Data Language)
  - eingebettet in PL/I, COBOL oder System/370 Assembler
  - navigierend

IMS (Skizze)



Konzeptuelle Ebene
Interne Ebene
Externe Transf.
Interne Transf.
P1
...
Pn
DL/I-Ops.
Einbettung
PL/I
PCBs
DBDs

UDS

- UDS $\cong$ Universelles Datenbank-System (Siemens, 1976)
- Betriebssystem BS 2000 für Siemens 7.100 bis 7.500
- Kein Optimierer, sonst Systemaufbau entsprechend ANSI-SPARC
- DML-Einbettung in COBOL, Assembler, FORTRAN, PASCAL

Begriffseinordnung

DDL	$\equiv$	Schema-DDL
VDL	$\equiv$	Subschema-DDL
SSL	$\equiv$	SSL
IQL	$\equiv$	IQS

Relationale Systeme

Gemeinsame Merkmale

- Drei-Ebenen-Architektur nach ANSI-SPARC
- einheitliche Datenbanksprache (SQL; Structured Query Language)
- Einbettung dieser Sprache in kommerzielle Programmiersprachen
- diverse Werkzeuge für Definition, Anfrage und Darstellung von Daten; Entwurf von Datenbank-Anwendungsprogrammen; Benutzer-Interaktion
- kontrollierter Mehrbenutzerbetrieb, Datenschutz- und Datensicherheitsmechanismen

Relationale Systeme II

Beispiele

- DB2 (SQL/DS)

  Entwicklung System R $\rightarrow$ DB2 $\leadsto \Box\ \Box$

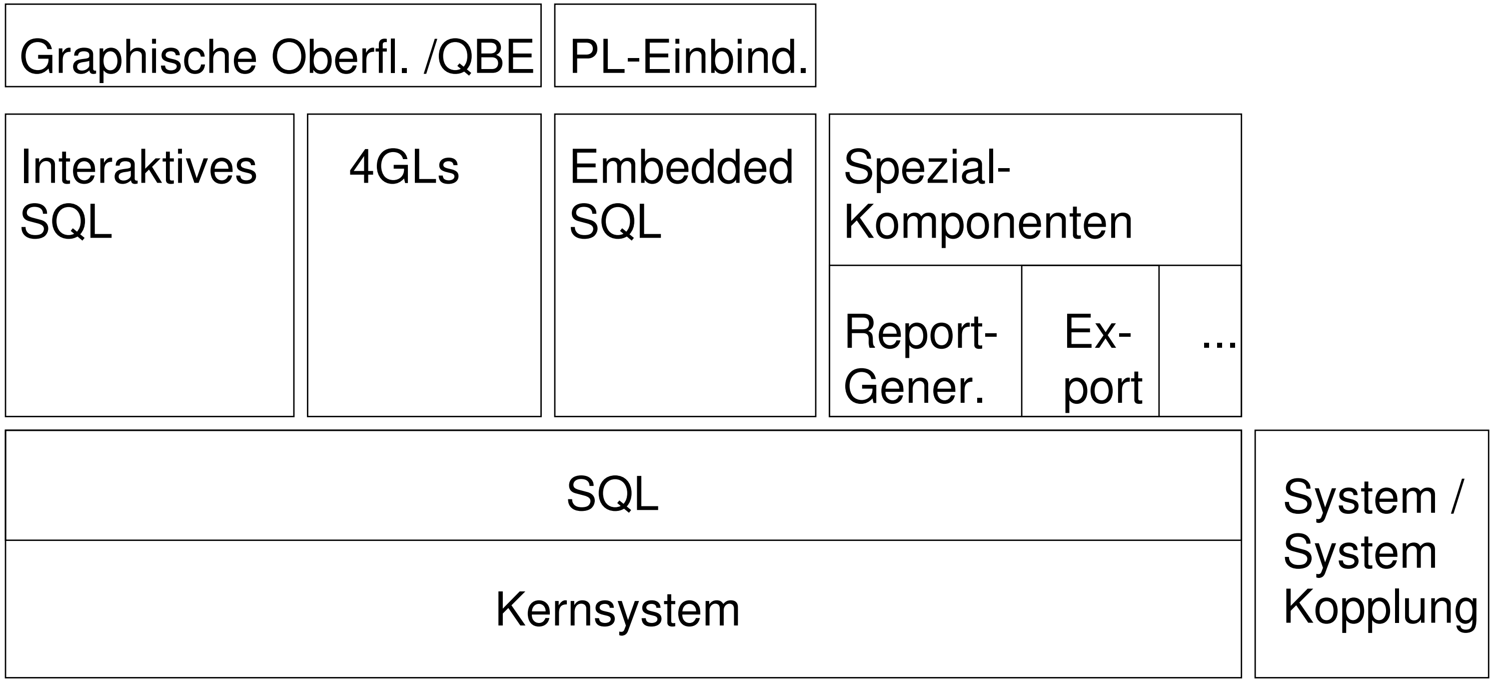
- ORACLE
- INGRES

Pseudo-RDBS (keine drei Ebenen, kein Mehrbenutzerbetrieb, ...)

- dBASE
- MS-Access

Anwendungs-Architekturen

Benutzerkomponenten



Abarbeiten eines DB-Anwendungsprogrammes

am Beispiel DB2

- Tabelle AUSLEIHE
- Spalten: NAME, INVENTARNR
- SQL-Anfrage:

```
SELECT NAME
FROM   AUSLEIHE
WHERE  INVENTARNR = 82;
```

soll in C-Programm eingebettet werden.

Abarbeiten eines DB-Anwendungsprogrammes II

- Aus Benutzer- / Anwendungsprogrammierer-Sicht besteht DB2 aus
  - Precompiler
  - Binder
  - Laufzeitüberwachungssystem
  - Dateiverwaltungssystem

DB2-Bausteine

- Precompiler für C, PL/I, COBOL, ...
  - Quellprogramm analysieren
  - Alle SQL-Statements durch Unterprogrammaufrufe ersetzen
  - Konstruktion eines Database Request Modules (DBRM) für jedes SQL-Statement, der Eingabe für Binder sein wird
- Binder
  - Ein oder mehrere DBRMs zu einem “Anwendungsplan” übersetzen
  - Der Anwendungsplan enthält ausführbaren Code zur Verwirklichung des ursprünglichen SQL-Statements
  - Der ausführbare Code enthält Betriebssystemaufrufe zum Dateiverwaltungssystem

DB2-Bausteine II

- Laufzeitüberwachungssystem
  - Überwacht die Ausführung des Anwendungsprogramms
  - Falls Anwendungsprogramm auf SQL-Statement stößt, wählt Laufzeitüberwachungssystem Teil eines passenden Anwendungsplanes aus
- Dateiverwaltungssystem
  - Verwirklicht schnellen Zugriff auf Daten
  - Z.B. Suchprozeduren, Updateoperationen, Indexverwaltung, . . .

Abarbeitung Anwendungsprogramm (Skizze)