



**Hasso
Plattner
Institut**

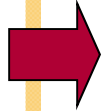
IT Systems Engineering | Universität Potsdam

Datenbanksysteme I
Historie, Begriffe und Architektur

13.4.2011

Felix Naumann

2



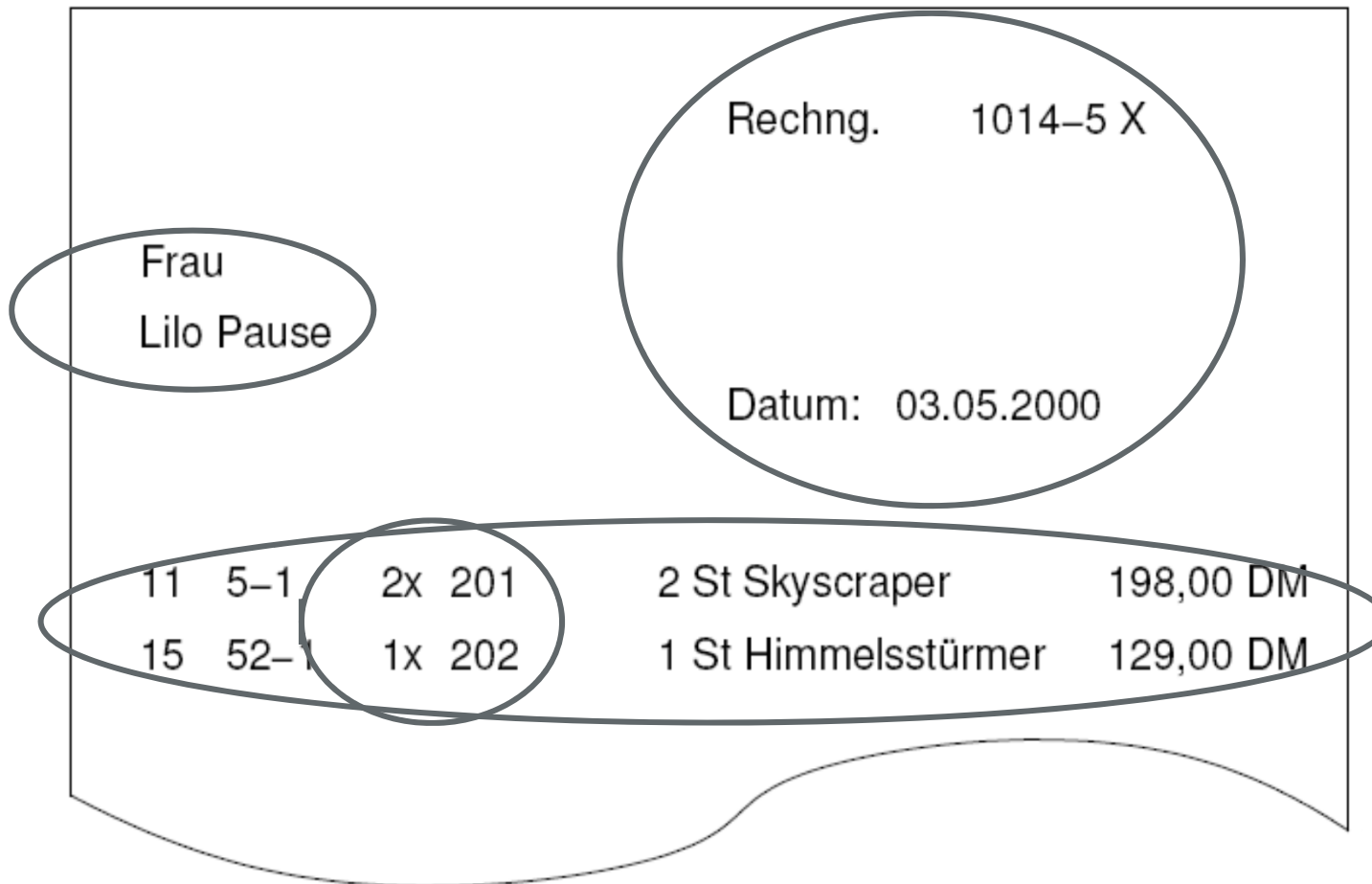
- Motivation
- Historie
- Architektur
- Datenunabhängigkeit
- Einsatzgebiete
- Ausblick



Folien basierend z.T. auf Foliensatz Prof. Sattler, TU Ilmenau

Beispiel: Rechnung

3



Daten in Tabellenform

4

- **Kundentabelle** speichert relevante Informationen des Kunden (Namen, Kontostand, Adresse etc.).
 - Kunde wird über die Kundennummer identifiziert.
- **Produkttable** mit Informationen zu Produktnamen, Lagerort (die ersten beiden Angaben auf der Rechnung), Preis, vorhandener Lagerbestand (nicht auf der Rechnung).
 - Identifikation erfolgt durch eine Produktnummer.
- Weitere Tabelle enthält **Rechnungs- und Lieferungsdaten** einzelner Rechnungen.
- Zur Vereinfachung werden **Rechnungspositionen** in einer separaten Tabelle gespeichert.

Beispiel – Tabellen

5

KUNDE

Knr	Vorname	Name
103	Lilo	Pause

RECHNUNG

RNr	Datum	Kunde	Status
1014	03.05.2000	103	geliefert

PRODUKT

PNr	Bezeichnung	Preis
201	Skyscraper	99.00
202	Himmelsstürmer	129.00

POSITION

RNr	PNr	Anzahl
1014	201	2
1014	202	1

Ohne Datenbanken: Datenredundanz

6

- 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 Bausteine
- Daten sind redundant (mehrfach gespeichert)
 - Verschwendung von Speicherplatz
 - „Vergessen“ von Änderungen
 - Inkonsistente Änderungen
 - keine zentrale, „genormte“ Datenhaltung

Ohne Datenbanken: Keine Effizienz

7

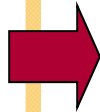
- Andere Software-Systeme können große Mengen von Daten nicht effizient verarbeiten.
- Mehrere Benutzer oder Anwendungen können nicht parallel auf den gleichen Daten arbeiten, ohne sich zu stören.
- Anwendungsprogrammierer / Benutzer können Anwendungen nicht programmieren / benutzen, ohne...
 - ...interne Darstellung der Daten und
 - Speichermedien oder Rechner...
- ...zu kennen (Datenunabhängigkeit nicht gewährleistet).
- Datenschutz und Datensicherheit sind nicht gewährleistet.

Die gesamte Basis- und Anwendungssoftware arbeitet auf denselben Daten (z.B. Adressen und Artikel werden nur einmal gespeichert).

- Datenbanksysteme können große Datenmengen effizient verwalten (Anfragesprachen, Optimierung, interne Ebene).
- Benutzer können parallel auf Datenbanken arbeiten (Transaktionskonzept).
- Datenunabhängigkeit durch 3-Schichten-Konzept
- Datenschutz (kein unbefugter Zugriff) und Datensicherheit (kein ungewollter Datenverlust) werden vom System gewährleistet.

9

- Motivation
- Historie
- Architektur
- Datenunabhängigkeit
- Einsatzgebiete
- Ausblick



Historie

10

• Datensammlungen... und Speichertechnologie

- Herman Hollerith: „punch card tabulating machine“
- Firmen – Vorläufer der IBM
 - Tabulating Machine Corp - 1896
 - Computing-Tabulating-Recording Company (C-T-R) - 1911
- International Business Machines Corporation (IBM) - 1924



Quelle: Prof. Freytag, Ringvorlesung 2005

Historie – Plattentechnologie

11

- Erste Platte: RAMAC 350 der IBM – 1955/56
 - „Random Access Method of Accounting and Control“
 - Entwickelt in San Jose, CA als sog. „bootleg“Projekt
 - Reynold B. Johnson (1906-1998): Technischer Leiter



Quelle: Prof. Freytag, Ringvorlesung 2005

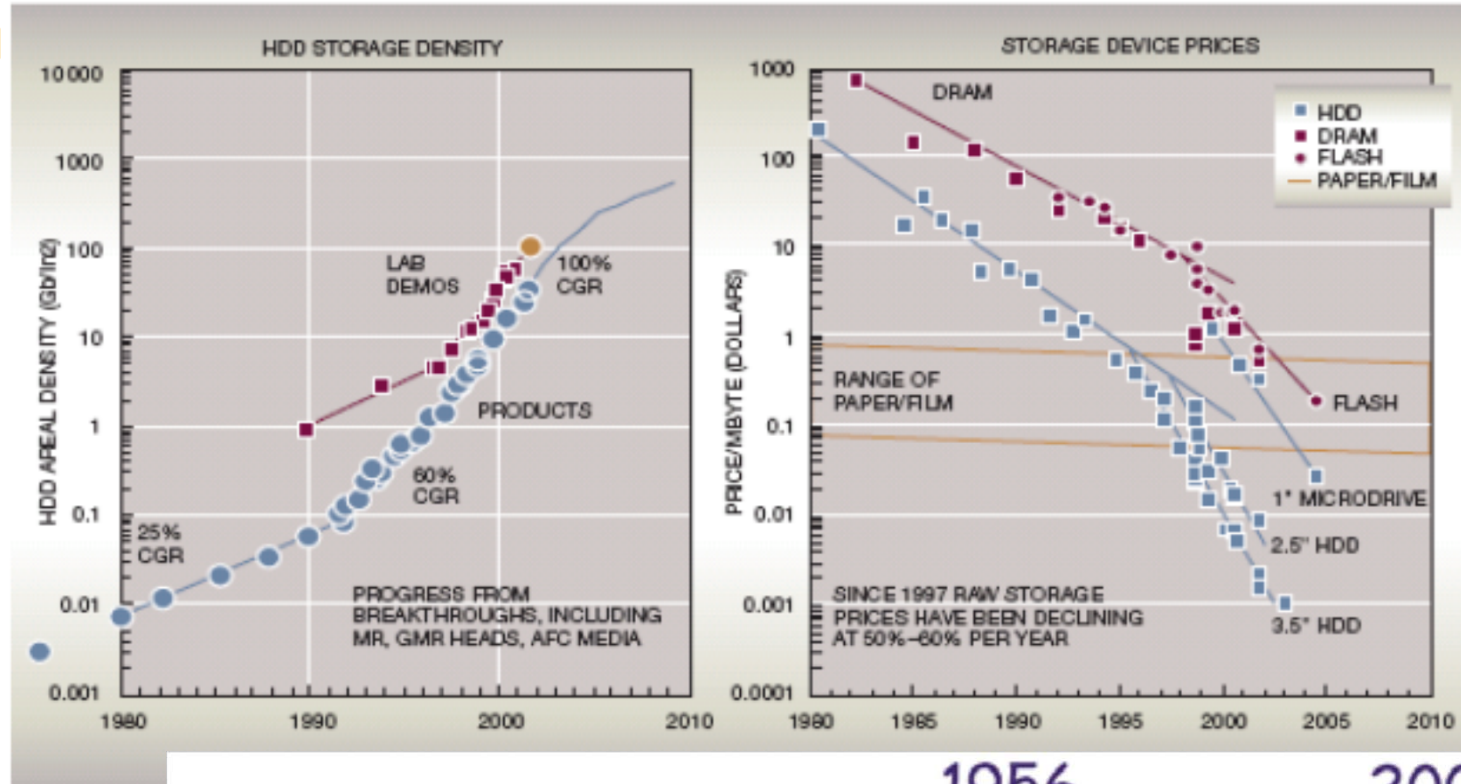


Bilder des Magnetic Disk Heritage Centers, Santa Clara, CA

Leistungsvergleich

- HDD Dichte: Verdopplung alle 18 Monate (Moore's Law)

12



Cost per megabyte
Areal density
Access time

	1956	2005
Cost per megabyte	\$10,000	\$0.001
Areal density	1000 bits/in ²	80 GB/in ²
Access time	1 sec.	4 msec.

Entwicklungslinien (Überblick)

13

- 60er: DBS basierend auf hierarchischem Modell, Netzwerkmodell
 - Zeigerstrukturen zwischen Daten
 - Schwache Trennung interne / konzeptuelle Ebene
 - Navigierende DML
 - Trennung DML / Programmiersprache
- 70er und 80er: Relationale Datenbanksysteme
 - Daten in Tabellenstrukturen
 - 3-Ebenen-Konzept
 - Deklarative DML
 - Trennung DML / Programmiersprache
- 80er und 90er
 - Immer größer
 - Immer kleiner
 - Objektorientierte DBs
- Heute
 - Spezialisierung auf neue Daten-Arten

Historie – Datenbanksysteme

14

- Anfang 60er Jahre
 - elementare Dateien
 - anwendungsspezifische Datenorganisation (geräteabhängig, redundant, inkonsistent)
 - Integrated Data Store (IDS) von General Electric
- Ende 60er Jahre
 - Dateiverwaltungssysteme (SAM, ISAM)
 - mit Dienstprogrammen (Sortieren) (geräteunabhängig, aber redundant und inkonsistent)
 - IBM: Information Management System (IMS)
 - ◇ Noch heute auf Mainframes im Einsatz
 - ◇ > 1.000.000.000 \$ Umsatz

Historie – Datenbanksysteme

15

Einsatzgebiete damals (und heute)

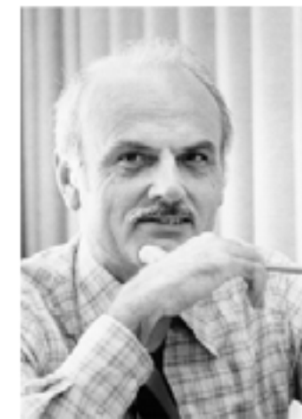
- Viele kleine Datensätze, viele Anfragen und Updates (OLTP)
- Flugbuchungssysteme
 - Reservierungen (Sitz, Mahlzeit, ...)
 - Flüge (Start, Ziel, Flugzeug, ...)
 - Tickets (Preise, Verfügbarkeit, ...)
 - Besonders wichtig: Simultaner Zugriff
- Banksysteme
 - Kunden, Konten, Kredite
 - Besonders wichtig: Konsistenz (z.B. Geldausgabe am Automaten)
- Warenwirtschaftssystem
 - Buchführung, Personalwesen, Steuern

Historie von Relationalen DBMS

16

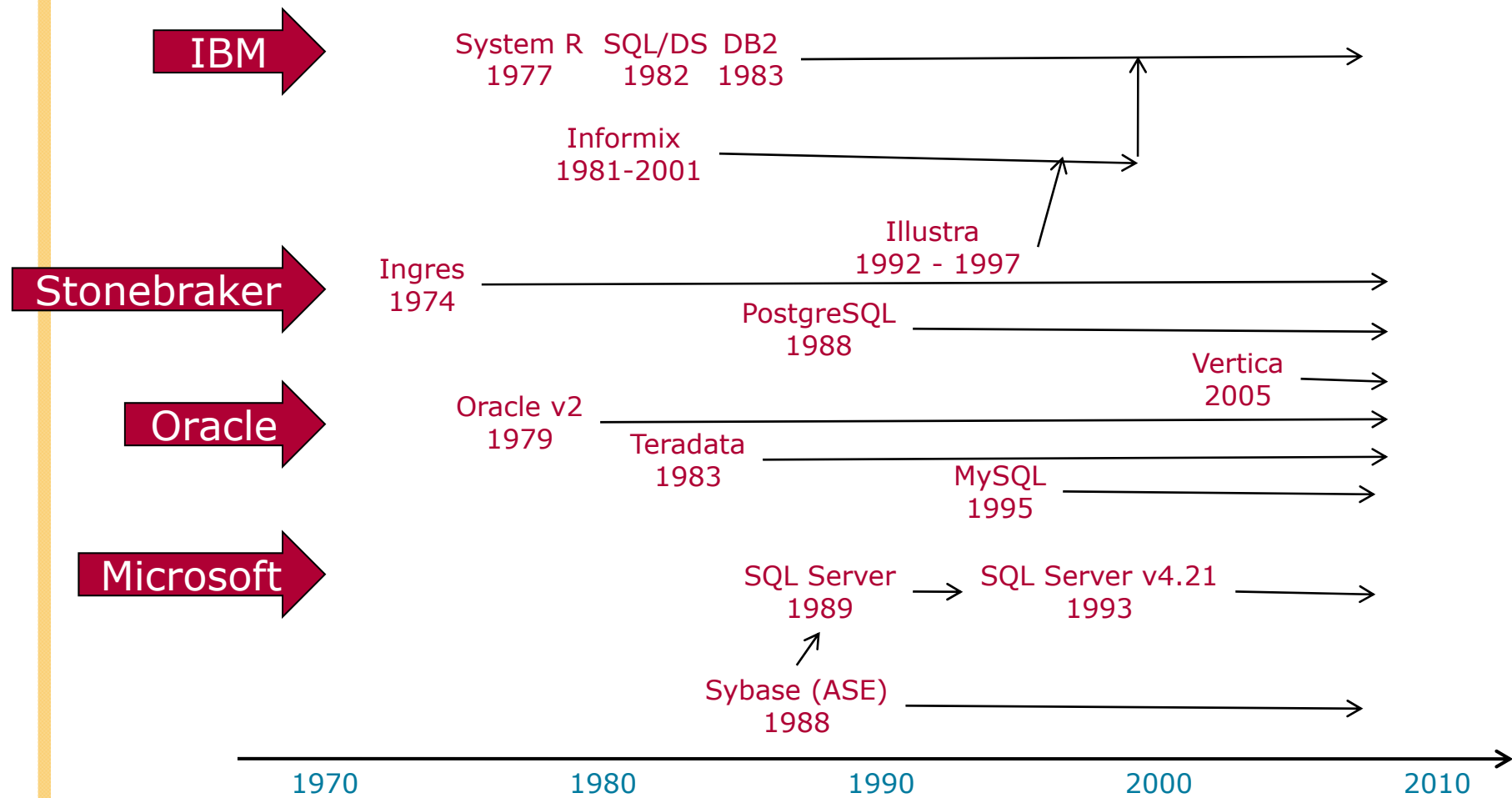
- 70er: Datenbanksysteme (Geräte- und Datenunabhängigkeit, redundanzfrei, konsistent)

- 1970: Ted Codd (IBM)
 - Relationenmodell als konzeptionelle Grundlage relationaler DBS
- 1974: System R (IBM)
 - erster Prototyp eines RDBMS
 - ca. 80.000 LOC (PL/1, PL/S, Assembler), ca. 1,2 MB Codegröße
 - Anfragesprache SEQUEL
 - erste Installation 1977
- 1975: University of California at Berkeley (UCB)
 - Ingres
 - Anfragesprache QUEL
 - Vorgänger von Postgres, Sybase, . . .
- 1979: Oracle Version 2



RDBMS Timeline

17



Entwicklungslinien 80er und 90er

19

- Immer kleinere Systeme
 - DBMS auch auf kleinen Rechnern
 - Oft schon teil der Standardinstallation
- Immer größere Systeme
 - Gigabyte, Terabyte
 - Große, komplexe (Multimedia-) Objekte
 - Tertiäre Speicher (Magnetbänder, DVDs)
 - Verteilung und Parallelität
- Objektorientierte Datenbanksysteme
 - Daten in komplexeren Objektstrukturen (Trennung Objekt und seine Daten)
 - Deklarative oder navigierende DML
 - Oft integrierte Datenbankprogrammiersprache
 - Oft keine vollständige Ebenentrennung

Entwicklungslinien Heute

20

Unterstützung für spezielle Anwendungen

- **Multimediatatenbanken**
 - Multimediale Objekte (Bilder, Audio, Video)
- **XML-Datenbanken**
 - Semistrukturierte Daten (XML-Dokumente)
- **Verteilte Datenbanken**
 - Daten auf verschiedene Rechnerknoten
- **Föderierte Datenbanken, Multidatenbanken, Mediatoren**
 - Integration von Daten aus heterogenen Quellen (Datenbanken, Dateien, Web-Quellen)
 - Informationsintegration
- **Mobile Datenbanken**
 - Datenverwaltung auf Kleinstgeräten (PDA, Handy, . . .)
- **Search!**

Weitere Themen

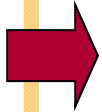
21

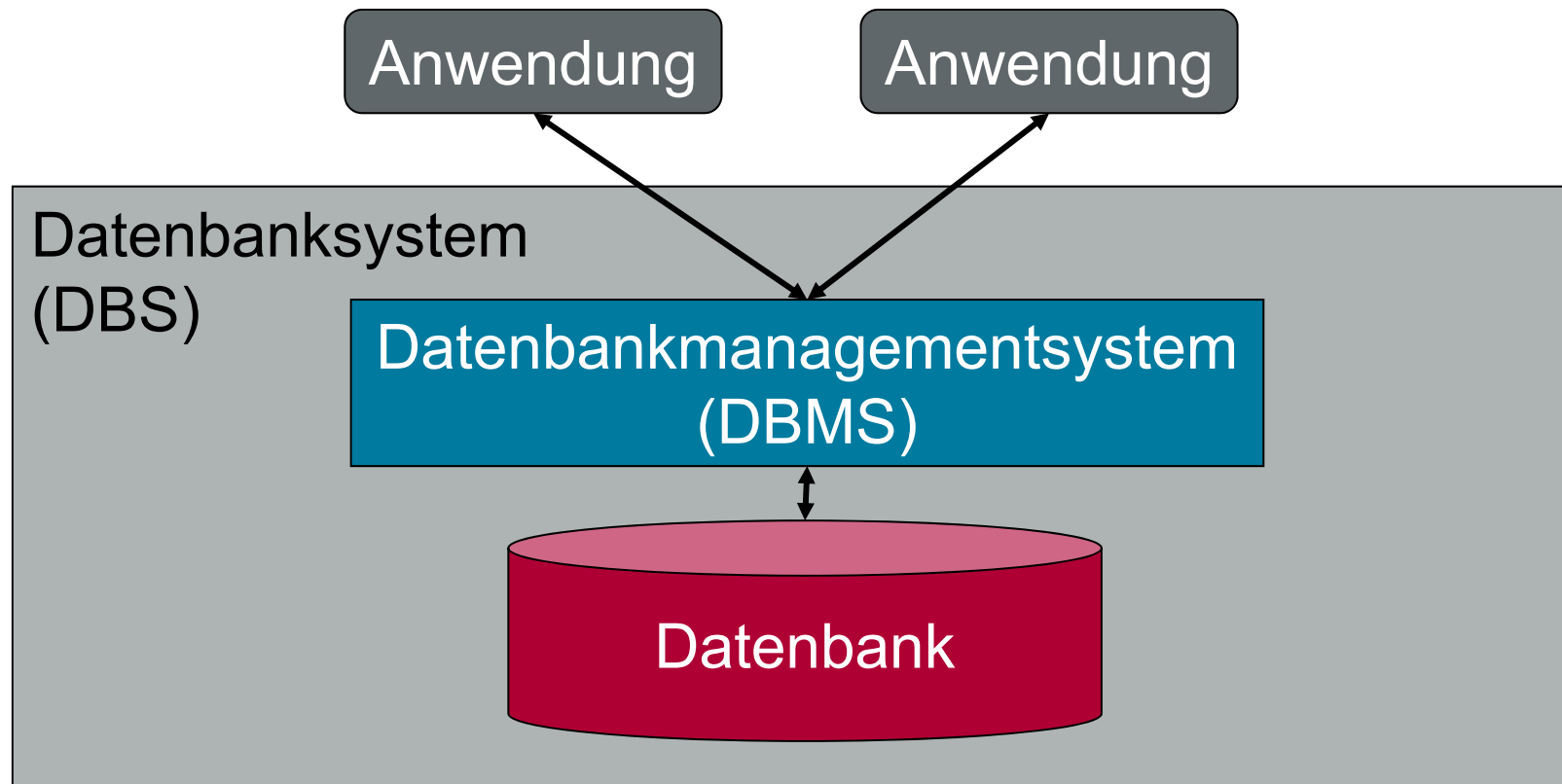
- Datenbankforschung und -entwicklung
 - Information Retrieval / Search
 - Data Warehouses
 - Spatial and temporal databases
 - Data Mining
 - Datenströme
 - P2P
 - ...

- Mehr dazu im Juli

22

- Motivation
- Historie
- Architektur
- Datenunabhängigkeit
- Einsatzgebiete
- Ausblick





Anforderungen an DBMS nach Codd 1982

24

- Integration
 - einheitliche, nichtredundante Datenverwaltung
- Operationen
 - Definieren, Speichern, Abfragen, Ändern
 - Deklarativ
- Katalog
 - Zugriffe auf Datenbankbeschreibungen im *Data Dictionary* (Metadaten)
- Benutzersichten
 - Verschiedene Anwendungen, Zugriffskontrolle, Umstrukturierung
- Integritätssicherung
 - Korrektheit und Konsistenz des Datenbankinhalts

Anforderungen an DBMS nach Codd 1982

25

- **Datenschutz**
 - Ausschluss nicht-autorisierter Zugriffe
- **Transaktionen**
 - mehrere DB-Operationen als Funktionseinheit
- **Synchronisation**
 - parallele Transaktionen koordinieren
- **Datensicherung**
 - Wiederherstellung von Daten nach Systemfehlern
 - Persistenz
 - Große Datenmengen, Effizienz

System Katalog (Beispiel DB2)

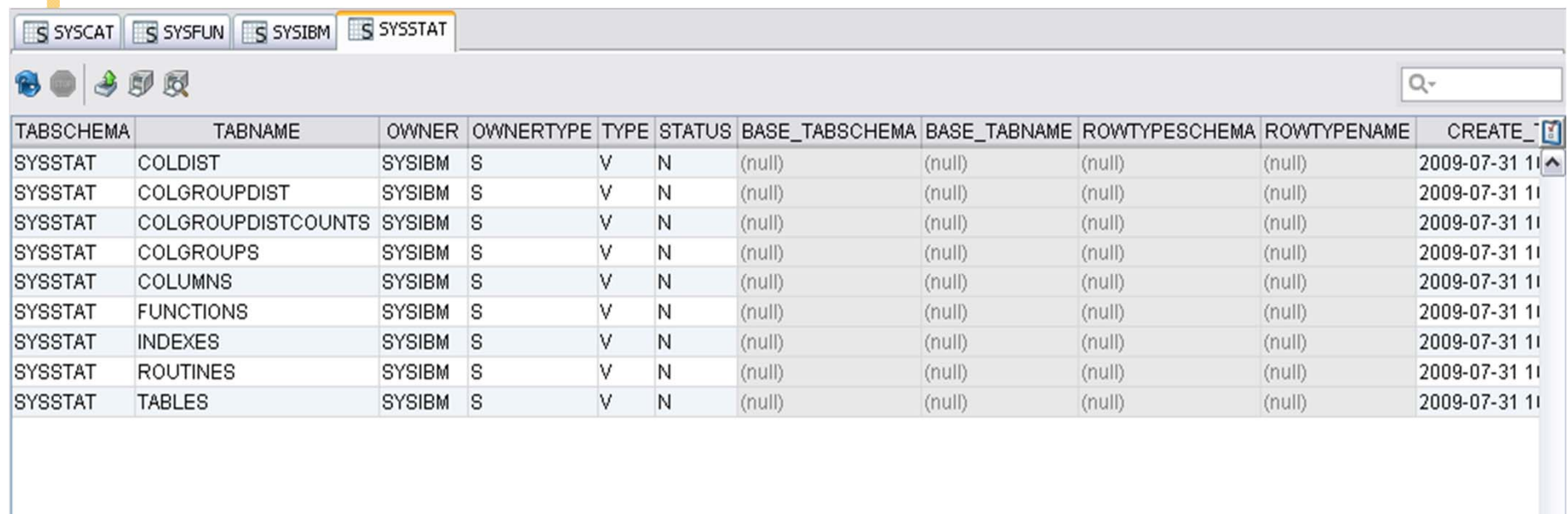
26

TABSCHEMA	TABNAME	OWNER	OWNERTYPE	TYPE	STATUS	BASE_TABSCHEMA	BASE_TABNAME	ROWTYPESCHEMA	ROWTYPENAME
SYSCAT	ATTRIBUTES	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	AUDITPOLICIES	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	AUDITUSE	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	BUFFERPOOLDBPARTITIONS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	BUFFERPOOLNODES	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	BUFFERPOOLS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	CASTFUNCTIONS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	CHECKS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	COLAUTH	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	COLCHECKS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	COLDIST	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	COLGROUPCOLS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	COLGROUPDIST	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	COLGROUPDISTCOUNTS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	COLGROUPS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	COLIDENTATTRIBUTES	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	COLLATIONS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	COOPTIONS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	COLUMNS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	COLUSE	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	CONSTDEP	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	CONTEXTATTRIBUTES	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	CONTEXTS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	DATAPARTITIONEXPRESSION	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	DATAPARTITIONS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	DATATYPES	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	DBAUTH	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	DBPARTITIONGROUPDEF	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	DBPARTITIONGROUPS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	EVENTMONITORS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	EVENTS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	EVENTTABLES	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	FULLHIERARCHIES	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	FUNCDEP	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	FUNCMAPOPTIONS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)
SYSCAT	FUNCMAPPARMOPTIONS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)

Fe

System Katalog (Beispiel DB2)

27



TABSHEMA	TABNAME	OWNER	OWNERTYPE	TYPE	STATUS	BASE_TABSCHEMA	BASE_TABNAME	ROWTYPESHEMA	ROWTYPENAME	CREATE_
SYSSTAT	COLDIST	SYSIBM	S	V	N	(null)	(null)	(null)	(null)	2009-07-31 11
SYSSTAT	COLGROUPDIST	SYSIBM	S	V	N	(null)	(null)	(null)	(null)	2009-07-31 11
SYSSTAT	COLGROUPDISTCOUNTS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)	2009-07-31 11
SYSSTAT	COLGROUPS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)	2009-07-31 11
SYSSTAT	COLUMNS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)	2009-07-31 11
SYSSTAT	FUNCTIONS	SYSIBM	S	V	N	(null)	(null)	(null)	(null)	2009-07-31 11
SYSSTAT	INDEXES	SYSIBM	S	V	N	(null)	(null)	(null)	(null)	2009-07-31 11
SYSSTAT	ROUTINES	SYSIBM	S	V	N	(null)	(null)	(null)	(null)	2009-07-31 11
SYSSTAT	TABLES	SYSIBM	S	V	N	(null)	(null)	(null)	(null)	2009-07-31 11

System Katalog (Beispiel DB2)

28

SELECT * FROM SYSCAT.TABLES ORDER BY TABSCHEMA

1:47 INS Auto Commit: ON Untitled*

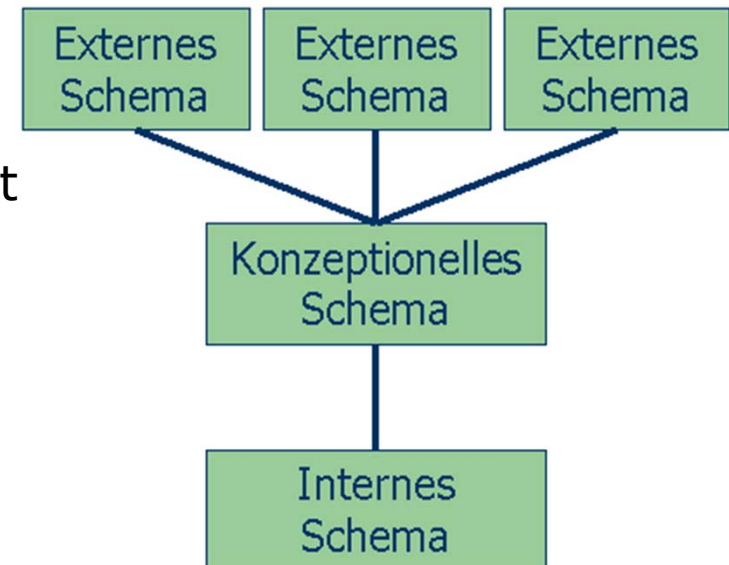
1: SELECT * FROM SYSCAT.TABLES OR...

	TABSCHEMA	TABNAME	OWNER	OWNERTYPE	TYPE	STATUS	BASE_TABSCHE
1	DBLP	DBLPUBS	HUMMER	U	T	N	(null)
2	DBLP	DBLPAUTHORS	HUMMER	U	T	N	(null)
3	DBLP	APY2	HUMMER	U	S	C	(null)
4	SYSCAT	ATTRIBUTES	SYSIBM	S	V	N	(null)
5	SYSCAT	AUDITPOLICIES	SYSIBM	S	V	N	(null)
6	SYSCAT	AUDITUSE	SYSIBM	S	V	N	(null)
7	SYSCAT	BUFFERPOOLNODES	SYSIBM	S	V	N	(null)
8	SYSCAT	CASTFUNCTIONS	SYSIBM	S	V	N	(null)
9	SYSCAT	CHECKS	SYSIBM	S	V	N	(null)
10	SYSCAT	COLAUTH	SYSIBM	S	V	N	(null)
11	SYSCAT	COLCHECKS	SYSIBM	S	V	N	(null)
12	SYSCAT	COLDIST	SYSIBM	S	V	N	(null)
13	SYSCAT	COLGROUPCOLS	SYSIBM	S	V	N	(null)
14	SYSCAT	COLGROUPDISTCOUNTS	SYSIBM	S	V	N	(null)
15	SYSCAT	COLLATIONS	SYSIBM	S	V	N	(null)
16	SYSCAT	COOPTIONS	SYSIBM	S	V	N	(null)
17	SYSCAT	COLUMNS	SYSIBM	S	V	N	(null)
18	SYSCAT	COLIDENTATTRIBUTES	SYSIBM	S	V	N	(null)
19	SYSCAT	COLUSE	SYSIBM	S	V	N	(null)
20	SYSCAT	BUFFERPOOLDBPARTITIONS	SYSIBM	S	V	N	(null)
21	SYSCAT	CONTEXTATTRIBUTES	SYSIBM	S	V	N	(null)

Schichtenmodell für Schemata

29

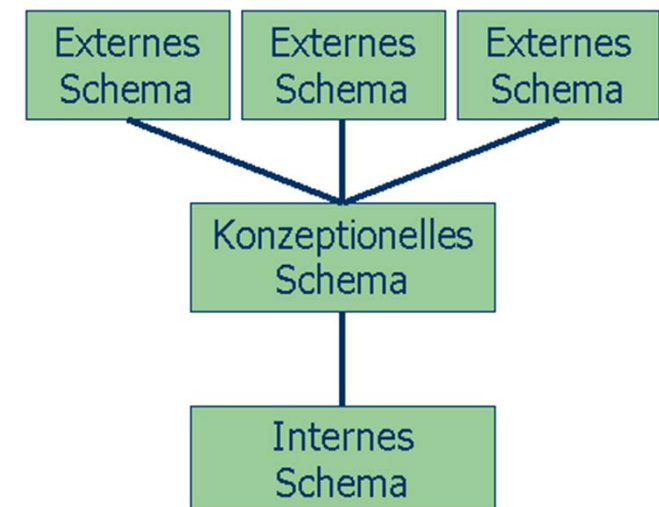
- Interne (physische) Schicht
 - Speichermedium (Tape, Festplatte)
 - Speicherort (Zylinder, Block)
- Konzeptionelle (logische) Schicht
 - Unabhängig von physischer Schicht
 - Definiert durch Datenmodell
 - Stabiler Bezugspunkt für interne und externe Schichten
- Externe (logische) Schicht
 - Anwendungsprogramme
 - Nur auf die relevanten Daten
 - Enthält Aggregationen und Transformationen



Schichtenmodell – Objekte

30

- Interne (physische) Schicht
 - Dateien, Blocks
- Konzeptionelle (logische) Schicht
 - Schema
 - ◇ Relationen, Attribute, Typen, Integritätsbedingungen
- Externe (logische) Schicht
 - „Views“ (Sichten)
 - Anwendungen



Systemarchitekturen

31

- Beschreibung der Komponenten eines Datenbanksystems
- Standardisierung der Schnittstellen zwischen Komponenten
- Architekturvorschläge
 - ANSI-SPARC-Architektur
 - ◇ Drei-Ebenen-Architektur
 - Fünf-Schichten-Architektur
 - ◇ beschreibt Transformationskomponenten im Detail
 - ◇ in weiterführenden Büchern und Vorlesungen

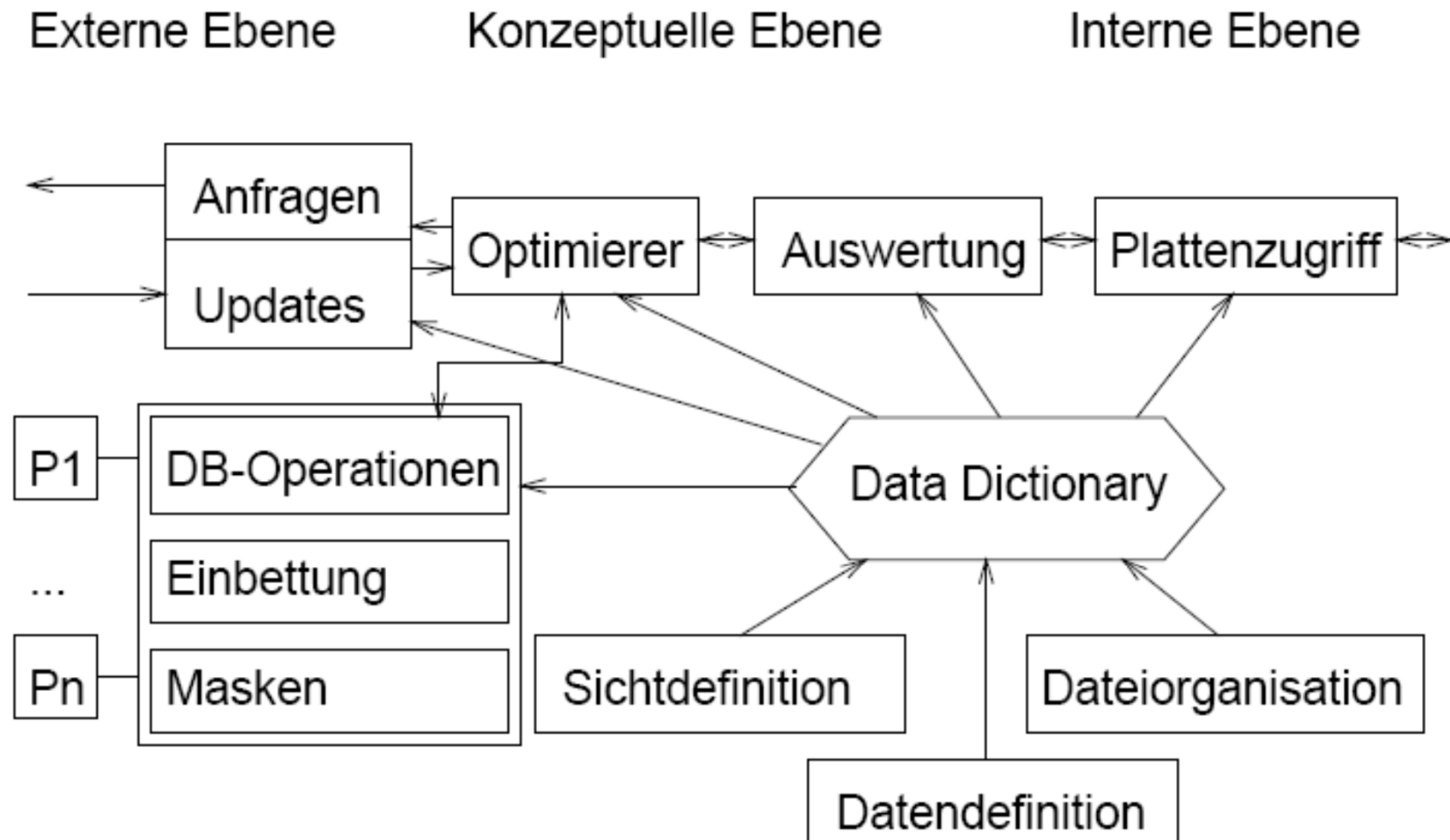
ANSI-SPARC-Architektur

32

- ANSI: American National Standards Institute
- SPARC: Standards Planning and Requirement Committee
- Vorschlag von 1978
- 3-Schichten-Architektur verfeinert
 - Interne Ebene / Betriebssystem verfeinert
 - Mehrere Interaktive und Programmier-Komponenten
 - Schnittstellen bezeichnet und normiert

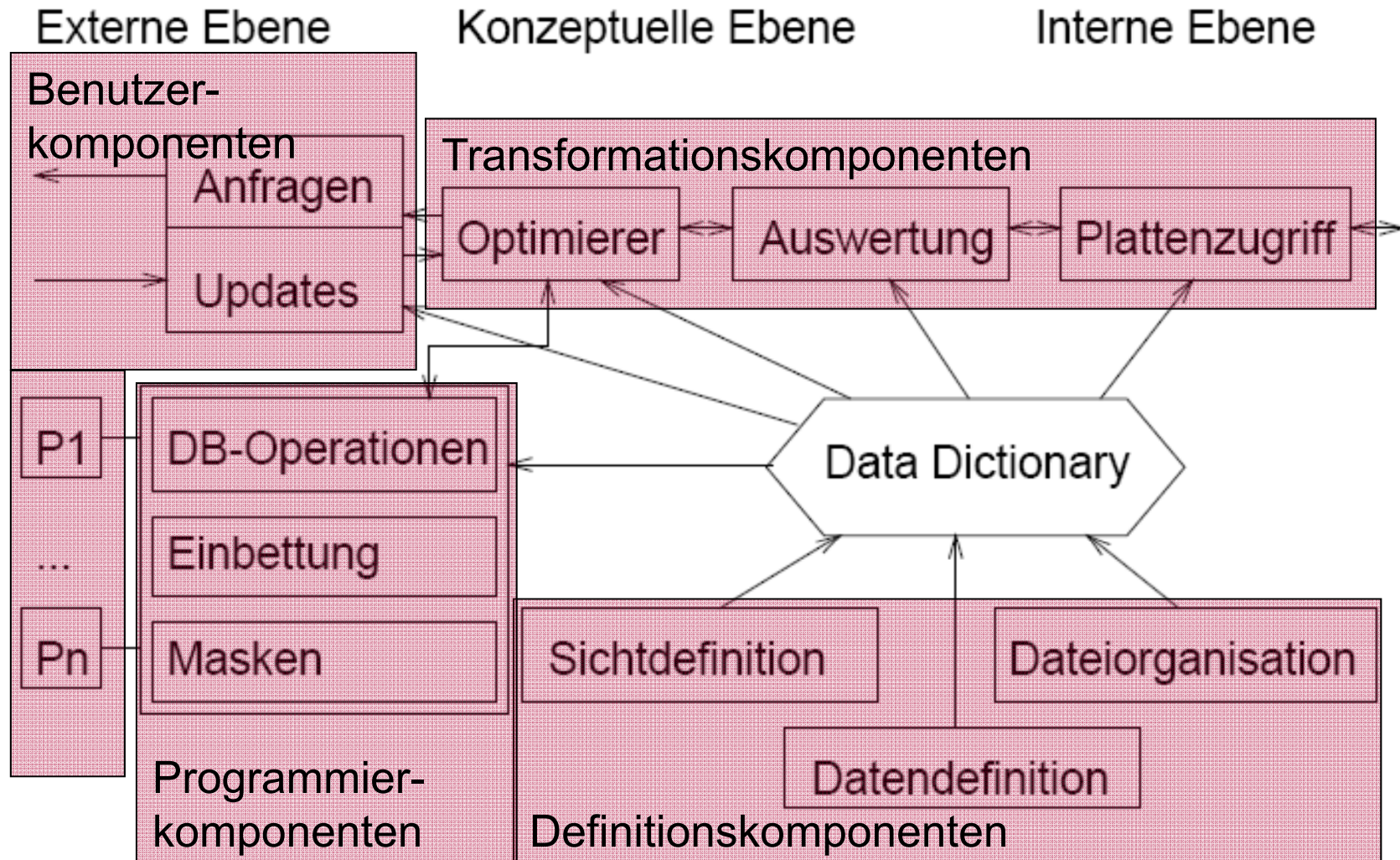
ANSI-SPARC-Architektur

33



ANSI-SPARC-Architektur

34

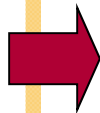


ANSI-SPARC-Komponenten

35

- Definitionskomponenten
 - DDL, Sichten, Dateiorganisation, Indizes
- Programmierkomponenten
 - Entwicklungsumgebung und Programmiersprache
 - Integration von DB-Operationen
- Benutzerkomponenten
 - Anfrageinterface für Experten
 - DB-Anwendungen für Laien
- Transformationskomponenten
 - Anfrageausführung und Darstellung der Ergebnisse
- Data Dictionary
 - Metadaten (in relationalen Systemtabellen)

- Motivation
- Historie
- Architektur
- Datenunabhängigkeit
- Einsatzgebiete
- Ausblick



Datenunabhängigkeit

37

Entkopplung von Benutzer- und Implementierungssicht

- Trennung von Modellierungssicht und interner Speicherung
- Portierbarkeit
- Tuning vereinfachen
- Standardisierte Schnittstellen
- Stabilität der Benutzerschnittstelle gegen Änderungen

Datenunabhängigkeit

38

Physische Datenunabhängigkeit

- Auch: *Implementierungsunabhängigkeit*
- Änderungen der Dateiorganisationen und Zugriffspfade haben keinen Einfluss auf das konzeptuelle Schema.

Logische Datenunabhängigkeit

- Auch: *Anwendungsunabhängigkeit*
- Änderungen am konzeptuellen und gewissen externen Schemata haben keine Auswirkungen auf andere externe Schemata und Anwendungsprogramme.

3-Schichten am Beispiel

39

- Konzeptionelle Sicht

Produkt

<u>Prodlid</u>	Bezeichnung	Preis	Herstld → Hersteller.Herstld
201	Skyscraper	99.0	901
203	Rainbow Hopper	45.0	902
205	2Hi4U	129.0	902

Hersteller

<u>Herstld</u>	LName
901	Flattermann GmbH
902	Dragon.com

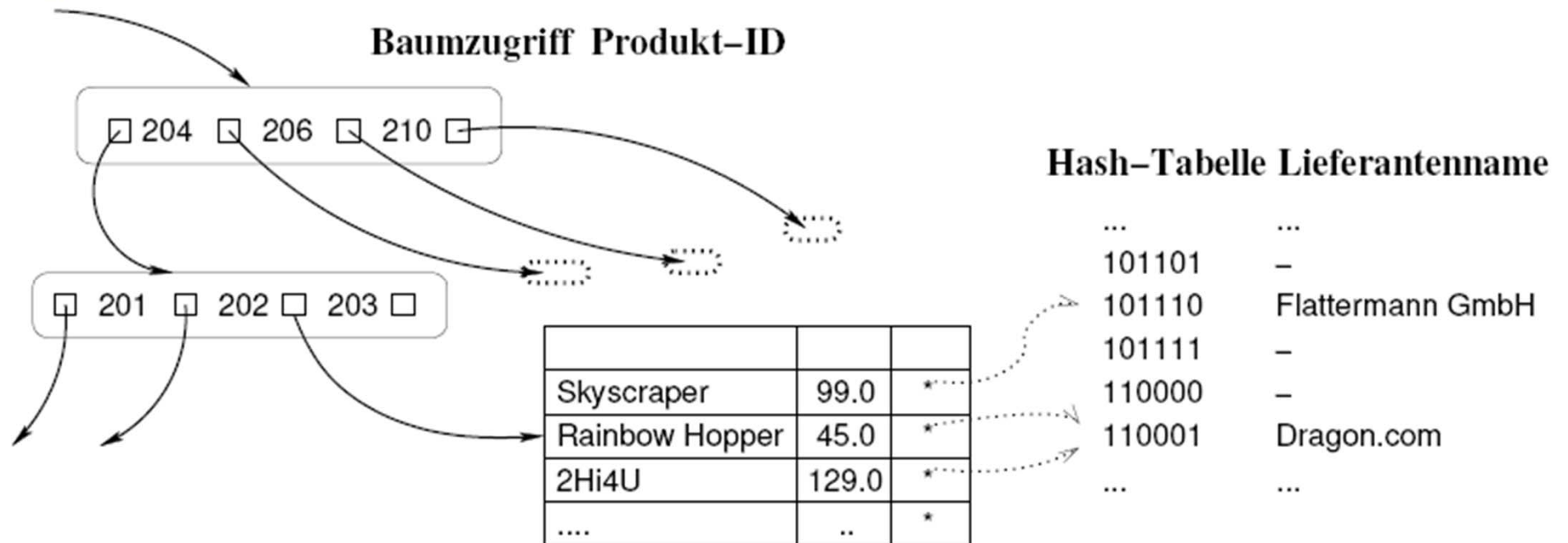
- Externe Sicht

<u>Prodlid</u>	Bezeichnung	Preis	LName
201	Skyscraper	99.0	Flattermann GmbH
203	Rainbow Hopper	45.0	Dragon.com
205	2Hi4U	129.0	Dragon.com

3-Schichten am Beispiel

40

■ Interne Sicht



41

- Motivation
- Historie
- Architektur
- Datenunabhängigkeit
- Einsatzgebiete
- Ausblick



Aktuelle DBMS (state of the art)

42

- Drei-Ebenen-Architektur nach ANSI-SPARC
- Einheitliche Datenbanksprache
 - SQL: Structured Query Language
- Einbettung dieser Sprache in kommerzielle Programmiersprachen
 - Embedded SQL, Dynamic SQL
 - JDBC, ODBC, SQLJ
- Diverse Werkzeuge für die Definition, Anfrage und Darstellung von Daten und den Entwurf von Datenbank-Anwendungsprogrammen und der Benutzer-Interaktion
- Kontrollierter Mehrbenutzerbetrieb, Zugriffskontrolle und Datensicherheitsmechanismen

Datenbankgrößen

43

- Sloan Digital Sky Survey 40 TB
 - Himmelsdaten (Bilder und Objektinformationen); bis 2004
- WalMart Data Warehouse 24 TB
 - Produktinfos (Verkäufe etc.) von 2.900 Märkten;
 - 50.000 Anfragen/Woche
- US Library of Congress 10-20 TB
 - nicht digitalisiert
- Indexierbares WWW (1999) 6 TB
 - ca. 800 Mill. Dokumente
- Microsofts TerraServer 3,5 TB
 - unkomprimierte Bilder/Karten (komprimiert: ca. 1 TB);
 - 174 Mill. Tupel

2005 TopTen Award Winners

Winter Corporation recognizes these organizations and their vendors for their achievements in the 2005 TopTen Program.

[List of all the winners](#) [Frequently Asked Questions](#)

Pick a TopTen Award Category:

Metric: Norm. Data Volume **Platform:** All Environments **Usage:** DW

Norm. Data Volume, All Environments, DW *

Company/Organization	Norm. Data Volume (GB)	DBMS	Platform	Architecture	DBMS Vendor	System Vendor	Storage Vendor
AT&T	330,644	Daytona	UNIX	Federated/SMP	AT&T	HP	HP
AT&T	93,468	Daytona	UNIX	Federated/SMP	AT&T	Sun	Sun
Amazon.com	28,184	Oracle RAC	Linux	Centralized/Cluster	Oracle	HP	HP
Nielsen Media Research	17,969	Sybase IQ	UNIX	Centralized/SMP	Sybase	Sun	EMC
Yahoo!	17,014	Oracle	UNIX	Centralized/SMP	Oracle	Fujitsu Siemens	EMC
Amazon.com	14,849	Oracle RAC	Linux	Centralized/Cluster	Oracle	HP	HP
UBS AG	14,177	Oracle	UNIX	Centralized/SMP	Oracle	Sun	EMC
China Telecom Corporation Co.,Ltd. GuangZhou Research Institute	13,241	Sybase IQ	UNIX	Centralized/SMP	Sybase	Sun	Sun
USDA	11,876	SQL Server	Windows	Centralized/SMP	Microsoft	HP	EMC
Reliance Infocomm Ltd	11,500	Oracle	UNIX	Centralized/SMP	Oracle	Sun	EMC
Cellcom	10,345	Oracle RAC	UNIX	Centralized/Cluster	Oracle	HP	EMC

Normalized Data Volume estimates of the total volume of data managed by the DBMS in GB.

- http://www.wintercorp.com/VLDB/2005_TopTen_Survey/TopTenWinners_2005.asp

Military Grapples With Information Overload (InformationWeek, July 9, 2009)

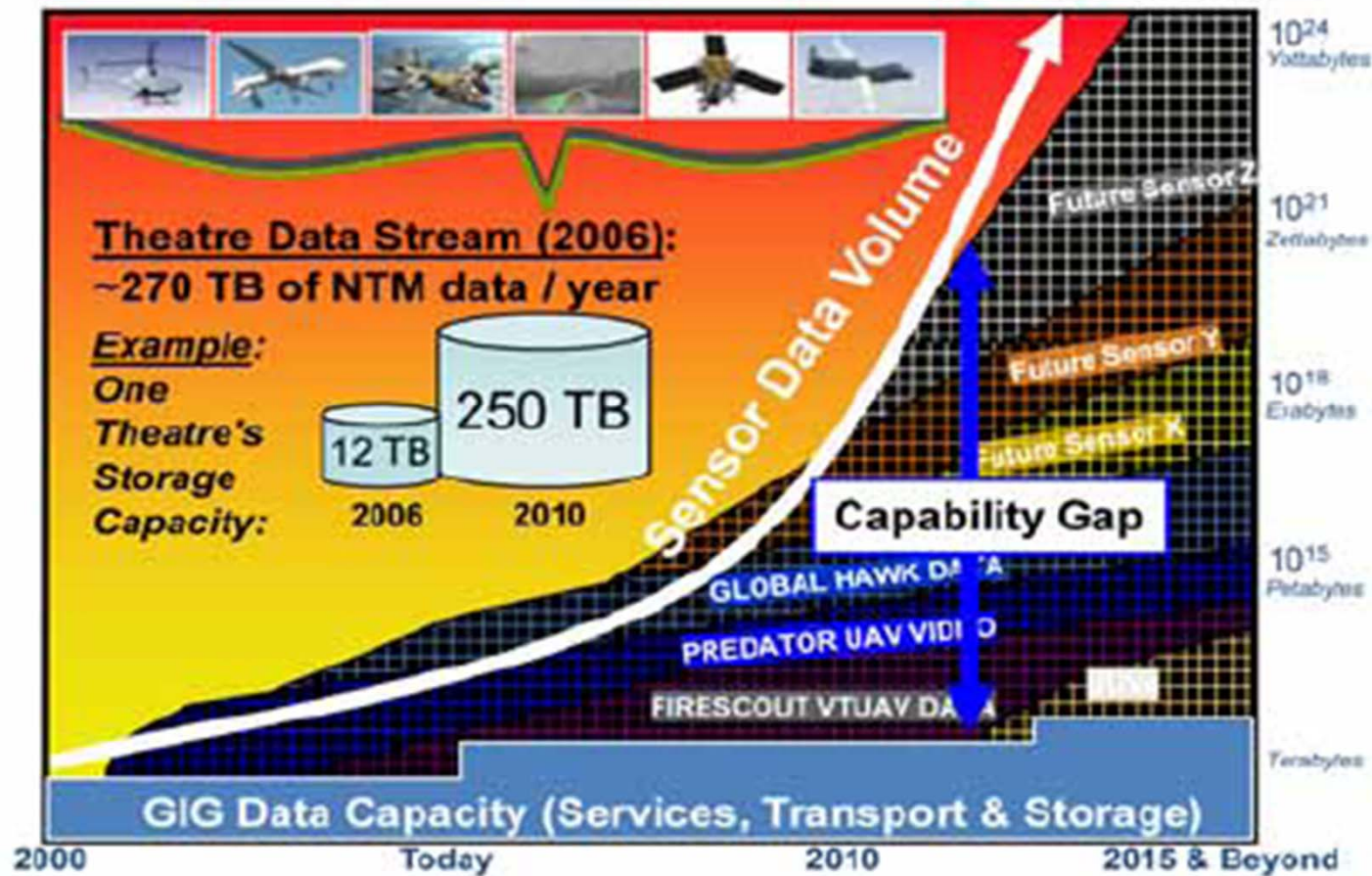
45

- "As the sensors associated with the various surveillance missions improve, the data volumes are increasing with a projection that sensor data volume could potentially increase to the level of Yottabytes (10^{24} Bytes) by 2015," the report says.
 - Referring to Data Analysis Challenges, JSR-08-142, JASON, The MITRE Corp, 12/08

- Using 1TB drives, this would require 1 trillion (10^{12}) drives!

Military Projection of Sensor Data Volume (later refuted)

46



Datenbankgrößen

47

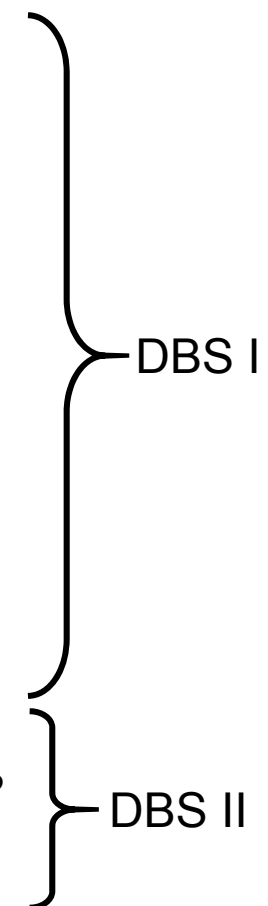
- SAP R/3-Installation der Deutschen Telekom AG (1998)
 - Financial Accounting: Rechnungen, Zahlungsaufforderungen, Lastschriften, Mahnungen etc.
- 15 SAP R/3-Systeme; jedes
 - verarbeitet 200.000 Rechnungen, 12.000 Mahnungen, 10.000 Änderungen von Kundendaten pro Tag
 - bis zu jeweils 1000 Nutzer gleichzeitig
 - über 13.000 Datenbanktabellen
- Hardware: 51 Unix Enterprise Server, 34 EMC-Speichersysteme (30 TB), 68 Magnetbandsysteme für Backup (Backup in 2h)

- Motivation
- Historie
- Architektur
- Datenunabhängigkeit
- Einsatzgebiete
- Ausblick



Datenbank – die großen Themen

49

- Datenbankdesign
 - Wie entwickelt man eine nützliche DB?
 - Was wird darin gespeichert?
 - Wie sind die Informationen strukturiert und miteinander verbunden?
 - Datenbankprogrammierung
 - Wie drückt man Anfragen und Updates aus?
 - Wie verwendet man andere DBMS Fähigkeiten (Transaktionen, ...)
 - Wie kombiniert man DBs mit herkömmlicher Programmierung
 - Implementierung von Datenbanksystemen
 - Wie entwickelt man ein DBMS (im Gegensatz zu einer DB)?
 - Wie macht man Operationen effizient?
- 
- DBS I
- DBS II