2. Anforderungen und Beschreibungsmodelle

Inhalt

Nutzung von Dateisystemen
(DB-)Grundbegriffe
Anforderungen an DBS
Schichtenmodelle für DBS
Drei-Schema-Architektur
Dynamischer Ablauf von DB-Operationen

Nutzung von Dateisystemen

- **Dateisystem**
  - **Permanente Datenhaltung innerhalb von BS-Dateien**
  - **Betriebssystem / Dateisystem bietet Funktionen für**
    - Erzeugen / Löschen von Dateien
    - Zugriffsmöglichkeiten auf Blöcke/Sätze der Datei
    - einfache Operationen zum Lesen/Ändern/Einfügen/Löschen von Sätzen (dynamisches Wachstum)
  - **Probleme / Nachteile**
    - Datenredundanz und Inkonsistenz
    - Inflexibilität
    - Mehrbenutzerbetrieb, Fehlerfall
    - Integritätssicherung
    - Missbrauch der Daten
    - Verantwortlichkeit

Grundbegriffe (1)

- **Datenbank (DB) als Abbildung einer Miniwelt**
  - Vorgänge und Sachverhalte werden als gedankliche Abstraktionen (Modelle) der Miniwelt erfasst und als Daten (Repräsentationen von Modellen) in der Datenbank gespeichert
  - Daten beziehen sich nur auf solche Aspekte der Miniwelt, die für die Zwecke der Anwendung relevant sind
  - Eine DB ist integritätserhaltend (bedeutungstreu), wenn ihre Objekte Modelle einer gegebenen Miniwelt repräsentieren
- **Datenmodell und DB-Schema**
  - Datenmodell (Typen, Operatoren, Konsistenzbedingungen) legt Regeln fest, nach denen die Objekte von DBs (für die Repräsentation beliebiger Miniwelten) erzeugt und verändert werden (Konstruktionsregeln für die Zustandsräume der Modelle)
  - DB-Schema legt die Ausprägungen der Objekte fest, welche die DB für eine bestimmte Miniwelt einnehmen kann (Zustandsraum der Modelle einer Miniwelt)

Grundbegriffe (2)

- **Beschreibung und Handhabung der Daten**
  - Daten müssen interpretierbar sein
  - sie müssen bei allen am Austausch beteiligten Partnern (Systemen, Komponenten) die Ableitung derselben Information erlauben

Schema: **ANGESTELLTER**

Satztyp (Relation)

Ausprägungen

PNR	NAME	TAETIGKEIT	GEHALT	ALTER
496	PEINL	PFOERTNER	2100	63
497	KINZINGER	KOPIST	2800	25
498	MEYWEG	KALLIGRAPH	4500	56

  - Interpretierbarkeit der Daten muss zeitinvariant sein
  - Einsatzspektrum verlangt **generische Vorgehensweise**
    - Beschreibung der zulässigen DB-Zustände
    - Beschreibung der zulässigen Zustandsübergänge (generische Operatoren)

Grundbegriffe (3)

- **Anwendungsprogrammier-Schnittstelle (API)**
  - Operatoren zur Definition von Objekttypen (Beschreibung der Objekte)
    - DB-Schema: Welche Objekte sollen in der DB gespeichert werden?
  - Operatoren zum Aufsuchen und Verändern von Daten
    - AW-Schnittstelle: Wie erzeugt, aktualisiert und findet man DB-Objekte?
  - Definition von Integritätsbedingungen (*Constraints*)
    - Sicherung der Qualität: Was ist ein akzeptabler DB-Zustand?
  - Definition von Zugriffskontrollbedingungen
    - Maßnahmen zum Datenschutz: Wer darf was?

Anforderungen an ein DBS (1)

1. **Kontrolle über die operationalen Daten**
   - **Alle Daten können/müssen gemeinsam benutzt werden**
     - keine verstreuten privaten Dateien
     - Querauswertungen aufgrund inhaltlicher Zusammenhänge
     - symmetrische Organisationsformen (keine Bevorzugung einer Verarbeitungs- und Auswertungsrichtung)
     - Entwicklung neuer Anwendungen auf der existierenden DB
     - Erweiterung/Anpassung der DB (Änderung des Informationsbedarfs)
   - **Redundanzfreiheit (aus Sicht der Anwendung)**
     - keine wiederholte Speicherung in unterschiedlicher Form für verschiedene Anwendungen
     - Vermeidung von Inkonsistenzen
     - zeitgerechter Änderungsdienst, keine unterschiedlichen Änderungsstände
   - **Datenbankadministrator (DBA):** zentrale Verantwortung für die operationalen Daten

Anforderungen an ein DBS (2)

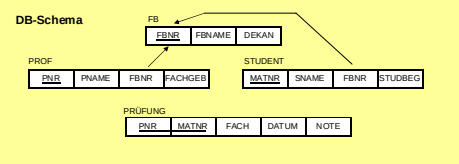
2. **Leichte Handhabbarkeit der Daten**
   - **Einfache Datenmodelle**
     - Beschreibung der logischen Aspekte der Daten
     - Benutzung der Daten ohne Bezug auf systemtechnische Realisierung
   - **Logische Sicht der Anwendung**
     - zugeschnitten auf ihren Bedarf
     - lokale Sicht auf die DB
   - **Leicht erlernbare Sprachen**
     - deskriptive Problemformulierung
     - hohe Auswahlmächtigkeit
     - Unterstützung der Problemlösung des Anwenders im Dialog

Anforderungen an ein DBS (3)

2. **Leichte Handhabbarkeit der Daten (Forts.)**
   - **Durchsetzung von Standards**
     - unterschiedliche DBS bieten einheitliche Schnittstelle
     - Portierbarkeit von Anwendungen
     - erleichterter Datenaustausch
   - **Erweiterung der Benutzerklassen**
     - Systempersonal
     - Anwendungsprogrammierer
     - anspruchsvolle Laien
     - parametrische Benutzer/ gelegentliche Benutzer

Anforderungen an ein DBS (4)

2. **Leichte Handhabbarkeit der Daten (Forts.)**
   - **Beispiel im Relationenmodell**



Anforderungen an ein DBS (5)

2. Leichte Handhabbarkeit der Daten (Forts.)

- **Beispiel im Relationenmodell (Forts.)**

FB

FBNR	FBNAME	DEKAN
FB 9	WIRTSCHAFTSWISS	4711
FB 5	INFORMATIK	2223

PROF

PNR	PNAME	FBNR	FACHGEB
1234	HÄRDER	FB 5	DATENBANKSYSTEME
5678	WEDEKIND	FB 9	INFORMATIONSSYSTEME
4711	MÜLLER	FB 9	OPERATIONS RESEARCH
6780	NEHMER	FB 5	BETRIEBSSYSTEME

STUDENT

MATNR	SNAME	FBNR	STUDBEG
123 766	COY	FB 9	1.10.95
225 332	MÜLLER	FB 5	15. 4.87
654 711	ABEL	FB 5	15.10.94
226 302	SCHULZE	FB 9	1.10.95
196 481	MAIER	FB 5	23.10.95
130 680	SCHMID	FB 9	1. 4.97

Ausprägungen

PRÜFUNG

PNR	MATNR	FACH	PDATUM	NOTE
5678	123 766	BWL	22.10.97	4
4711	123 766	OR	16. 1.98	3
1234	654 711	DV	17. 4.97	2
1234	123 766	DV	17. 4.97	4
6780	654 711	SP	19. 9.97	2
1234	196 481	DV	15.10.97	1
6780	196 481	BS	23.12.97	3

Anforderungen an ein DBS (6)

2. Leichte Handhabbarkeit der Daten (Forts.)

- **Beispiel im Relationenmodell (Forts.)**
  - **Deskriptive DB-Sprachen (wie SQL)**
    - hohes Auswahlvermögen und Mengenorientierung
    - leichte Erlernbarkeit auch für den DV-Laien
    - RM ist symmetrisches Datenmodell, d.h., es gibt keine bevorzugte Zugriffs- oder Auswertungsrichtung
  - Anfragebeispiele

I. Finde alle Studenten aus Fachbereich 5, die ihr Studium vor 1990 begonnen haben.

```
SELECT  *
FROM    STUDENT
WHERE   FBNR = 'FB5' AND STUDBEG < '1.1.95'
```


Anforderungen an ein DBS (7)

2. Leichte Handhabbarkeit der Daten (Forts.)

- **Beispiel im Relationenmodell (Forts.)**
  - Anfragebeispiele (Forts.)

II. Finde alle Studenten des Fachbereichs 5, die im Fach Datenverwaltung eine Note 2 oder besser erhalten haben.

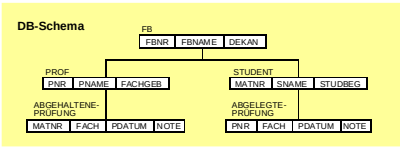
```
SELECT  *
FROM    STUDENT
WHERE   FBNR = 'FB5' AND MATNR IN
        (SELECT MATNR
        FROM    PRÜFUNG
        WHERE   FACH = 'DV' AND NOTE ≤ '2')
```

III. Finde die Durchschnittsnoten der DV-Prüfungen für alle Fachbereiche mit mehr als 1000 Studenten.

```
SELECT  S.FBNR, AVG (P.NOTE)
FROM    PRÜFUNG P, STUDENT S
WHERE   P.FACH = 'DV' AND P.MATNR = S.MATNR
GROUP BY S.FBNR
HAVING (SELECT COUNT(*)
        FROM    STUDENT T
        WHERE   T.FBNR = S.FBNR) > 1000
```

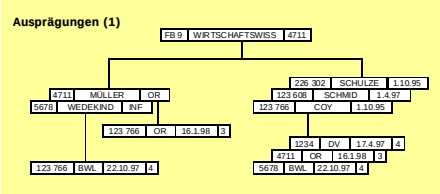
Anforderungen an ein DBS (8)

2. **Leichte Handhabbarkeit der Daten (Forts.)**
   - **Beispiel im Hierarchiemodell**



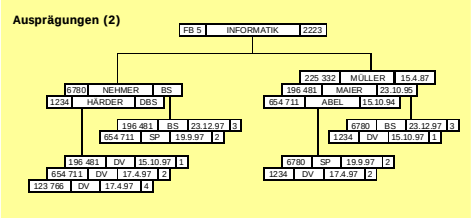
Anforderungen an ein DBS (9)

2. **Leichte Handhabbarkeit der Daten (Forts.)**
   - **Beispiel im Hierarchiemodell (Forts.)**



Anforderungen an ein DBS (10)

2. **Leichte Handhabbarkeit der Daten (Forts.)**
   - **Beispiel im Hierarchiemodell (Forts.)**



Anforderungen an ein DBS (11)

2. **Leichte Handhabbarkeit der Daten (Forts.)**
   - **Beispiel im Hierarchiemodell (Forts.)**
     - **Prozedurale DB-Sprachen**
       - Programmierer als Navigator
       - satzweiser Zugriff über vorhandene Zugriffspfade
       - bei hierarchischen Datenmodellen
       - unnatürliche Organisation bei komplexen Beziehungen (n:m)
       - Auswertungsrichtung ist vorgegeben

Anforderungen an ein DBS (12)

2. **Leichte Handhabbarkeit der Daten (Forts.)**
   - **Beispiel im Hierarchiemodell (Forts.)**
     - **Anfragebeispiele**

I. Finde alle Studenten aus Fachbereich 5, die ihr Studium vor 1990 begonnen haben.

```
                  GU FB (FBNR = 'FB5');
NEXT_STUDENT:     GNP STUDENT (STUDBEG < '1.1.90');
                  IF END_OF_PARENT THEN EXIT;
                  PRINT STUDENT RECORD;
                  GOTO NEXT_STUDENT;
```


Anforderungen an ein DBS (13)

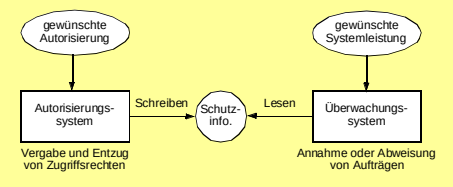
2. **Leichte Handhabbarkeit der Daten (Forts.)**
   - **Beispiel im Hierarchiemodell (Forts.)**
     - **Anfragebeispiele (Forts.)**

II. Finde alle Studenten des Fachbereichs 5, die im Fach Datenverwaltung eine Note 2 oder besser erhalten haben.

```
                  GU FB (FBNR = 'FB5');
NEXT_STUDENT:     GNP STUDENT;
                  IF END_OF_PARENT THEN EXIT;
                  GNP ABGELEGTE_PRÜFUNG
                        (FACH = 'DV') AND (NOTE ≤ '2');
                  IF END_OF_PARENT THEN GOTO
                        NEXT_STUDENT;
                  PRINT STUDENT RECORD;
                  GOTO NEXT_STUDENT;
```

Anforderungen an ein DBS (14)

3. **Kontrolle der Datenintegrität**
   - **Automatisierte Zugriffskontrollen (Datenschutz)**
     - separat für jedes Datenobjekt
     - unterschiedliche Rechte für verschiedene Arten des Zugriffs
     - **Idealziel**: „least privilege principle“



Anforderungen an ein DBS (15)

3. **Kontrolle der Datenintegrität (Forts.)**
   - **Erhaltung der logischen Datenintegrität (system enforced integrity)**
     - Beschreibung der „Richtigkeit“ von Daten durch Prädikate und Regeln
     - „Qualitätskontrollen“ bei Änderungsoperationen
     - aktive Maßnahmen des DBS erwünscht (ECA-Regeln)

Anforderungen an ein DBS (16)

3. **Kontrolle der Datenintegrität (Forts.)**
   - **Transaktionskonzept** (Durchsetzung der ACID-Eigenschaften)

May all your transactions commit and never leave you in doubt. (J. Gray)

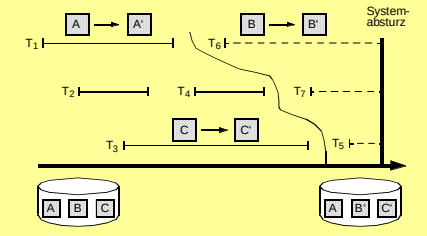
   - „Schema-Konsistenz (**C**) aller DB-Daten wird bei Commit erzwungen
   - ACID impliziert Robustheit, d. h., DB enthält nur solche Zustände, die explizit durch erfolgreich abgeschlossene TA erzeugt wurden
     - **D**auerhaftigkeit (Persistenz): Effekte von abgeschlossenen TA gehen nicht verloren
     - **A**tomarität (Resistenz): Zustandsänderungen werden entweder, wie in der TA spezifiziert, vollständig durchgeführt oder überhaupt nicht
   - Im Mehrbenutzerbetrieb entsteht durch nebenläufige TA ein Konkurrenzverhalten (concurrency) um gemeinsame Daten, d. h., TA geraten in Konflikt
     - **I**solationseigenschaft: TA-Konflikte sind zu verhindern oder aufzulösen

Anforderungen an ein DBS (17)

3. **Kontrolle der Datenintegrität (Forts.)**
   - **Erhaltung der physischen Datenintegrität**
     - Periodisches Erstellen von Datenkopien
     - Führen von Änderungsprotokollen für den Fehlerfall (Logging)
     - Bereitstellen von Wiederherstellungsalgorithmen im Fehlerfall (Recovery)
       - **Garantie** nach erfolgreichem Neustart: jüngster transaktionskonsistenter DB-Zustand
   - **Notwendigkeit des kontrollierten Mehrbenutzerbetriebs**
     - logischer Einbenutzerbetrieb für jeden von n parallelen Benutzern (Leser + Schreiber)
     - geeignete Synchronisationsmaßnahmen zur gegenseitigen Isolation
     - angepasste Synchronisationseinheiten (z. B. Sperrgranulate) mit abgestuften Zugriffsmöglichkeiten
       - **Ziel**: möglichst geringe gegenseitige Behinderung

Anforderungen an ein DBS (18)

3. **Kontrolle der Datenintegrität (Forts.)**
   - **Zu: Erhaltung der physischen Datenintegrität**



Anforderungen an ein DBS (19)

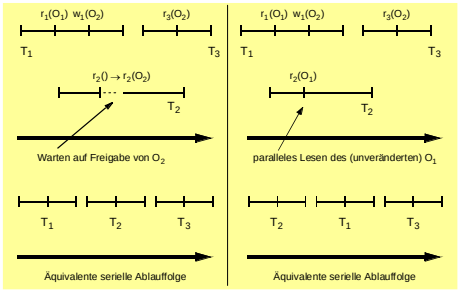
3. **Kontrolle der Datenintegrität (Forts.)**
   - **Zu: Erhaltung der physischen Datenintegrität (Forts.)**
     - **DBMS garantiert physische Datenintegrität**
       - bei jedem Fehler (z. B. Ausfall des Rechners, Absturz des Betriebssystems oder des DBMS, Fehlerhaftigkeit einzelner Transaktionsprogramme) wird eine „korrekte" Datenbank rekonstruiert
       - nach einem (Teil-)Absturz ist immer der jüngste transaktionskonsistente Zustand der DB zu rekonstruieren, in dem alle Änderungen von Transaktionen enthalten sind, die vor dem Zeitpunkt des Fehlers erfolgreich beendet waren (T_1 bis T_4) und sonst keine
       - automatische Wiederherstellung nach Neustart des Systems
     - **Maßnahmen beim Wiederanlauf (siehe auch Beispiel)**
       - Ermittlung der beim Absturz aktiven Transaktionen (T_5, T_6, T_7)
       - Rücksetzen (UNDO) der Änderungen der aktiven Transaktionen in der Datenbank (B' → B)
       - Wiederholen (REDO) der Änderungen von abgeschlossenen Transaktionen, die vor dem Absturz nicht in die Datenbank zurückgeschrieben waren (A → A')

Anforderungen an ein DBS (20)

3. **Kontrolle der Datenintegrität (Forts.)**
   - **Zu: Notwendigkeit der Kontrolle des Mehrbenutzerbetriebs**
     - Beim **logischen Einbenutzerbetrieb** hat jede der parallel aktiven Transaktionen den ‚Eindruck', als liefe sie alleine ab, d. h., logisch bilden alle Transaktionen eine serielle Ablauffolge
     - **Synchronisationskomponente** des DBMS umfasst alle Maßnahmen zur Sicherstellung der Ablaufintegrität (Isolation der parallelen Transaktionen)
     - **Formale Definition:** Eine parallele Ablauffolge von Transaktionen ist genau dann korrekt synchronisiert, wenn es eine zu dieser Ablauffolge äquivalente (bezüglich ihrer Lese- und Schreibabhängigkeiten (r, w)) serielle Ablauffolge gibt, so dass jede Transaktion T_i in der seriellen Reihenfolge dieselben Werte liest und schreibt wie im parallelen Ablauf. (Dabei ist jede Permutation der T_i-Folge gleichermaßen zulässig, siehe Beispiel)

Anforderungen an ein DBS (21)

3. **Kontrolle der Datenintegrität (Forts.)**
   - **Zu: Notwendigkeit der Kontrolle des Mehrbenutzerbetriebs (Forts.)**



Anforderungen an ein DBS (22)

4. **Leistung und Skalierbarkeit**
   - **DBS-Implementierung gewährleistet**
     - **Effizienz** der Operatoren (möglichst geringer Ressourcenverbrauch)
     - **Verfügbarkeit** der Daten (Redundanz, Verteilung usw.)
   - **Ausgleich von Leistungsanforderungen, die im Konflikt stehen**
     - globale Optimierung durch den DBA (Rolle des internen Schemas)
     - ggf. Nachteile für einzelne Anwendungen
   - **Effizienz des Datenzugriffs**
     - Zugriffsoptimierung durch das DBS, nicht durch den Anwender
     - Anlegen von Zugriffspfaden durch den DBA, Auswahl idealerweise durch das DBS

Anforderungen an ein DBS (23)

4. **Leistung und Skalierbarkeit (Forts.)**
   - **Leistungsbestimmung**
     - Maßzahlen für Leistung
       - Durchsatz: Anzahl abgeschlossener TA pro Zeiteinheit (meist Sekunde)
       - Antwortzeit: Zeitbedarf für die Abwicklung einer TA
     - Rolle von Benchmarks: TPC-C, TPC-H, TPC-W, TPC-R, . . .
   - **Skalierbarkeit**
     - Software- und Hardware-Architektur sollen hinsichtlich des DBS-Leistungsverhaltens automatisch durch Hinzufügen von Ressourcen (CPU's, Speicher) skalieren
       - Scaleup: bei Wachstum der Anforderungen (DB-Größe, Transaktionslast)
       - Speedup: zur Verringerung der Antwortzeit

Transaction Processing Council: www.tpc.org

Anforderungen an ein DBS (24)

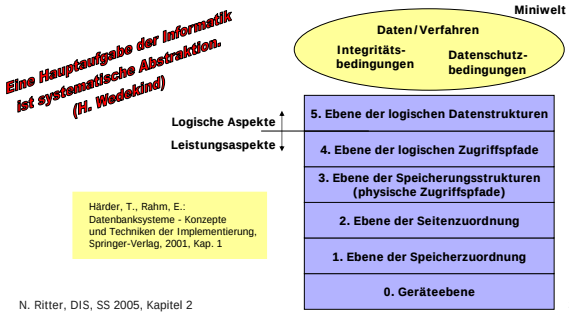
5. **Hoher Grad an Daten-Unabhängigkeit**
   - **Konventionelle Anwendungsprogramme (AP) mit Dateizugriff**
     - Nutzung von Kenntnissen der Datenorganisation und Zugriffstechnik
     - gutes Leistungsverhalten, aber . . . ?
   - **Datenabhängige Anwendungen sind äußerst unerwünscht**
     - Rolle des Datenmodells: Vergleiche relationales und hierarchisches Datenmodell
     - Verschiedene Anwendungen brauchen verschiedene Sichten auf dieselben Daten
     - Änderungen im Informationsbedarf sowie bei Leistungsanforderungen erzwingen Anpassungen bei Speicherungsstrukturen und Zugriffsstrategien
   - deshalb: *möglichst starke Isolation der APs von den Daten*
     sonst: extremer Wartungsaufwand für die APs

Anforderungen an ein DBS (25)

5. **Hoher Grad an Daten-Unabhängigkeit (Forts.)**
   - **Realisierung verschiedener Arten von *Daten-Unabhängigkeit*:**
     - **Minimalziel:** physische Daten-Unabhängigkeit (durch das BS/DBS)
       - **Geräte**unabhängigkeit
       - **Speicherungsstruktur**-Unabhängigkeit
     - logische Daten-Unabhängigkeit (vor allem durch das Datenmodell!)
       - **Zugriffspfad**-Unabhängigkeit
       - **Datenstruktur**-Unabhängigkeit

Schichtenmodelle für DBS (1)

- Ebenen beim Entwurf eines DBS



Härder, T., Rahm, E.: Datenbanksysteme - Konzepte und Techniken der Implementierung, Springer-Verlag, 2001, Kap. 1

Schichtenmodelle für DBS (2)

- Ziel: Architektur eines daten-unabhängigen DBS
  - **Systementwurf**
    - Beschreibungs- und Kommunikationstechniken notwendigerweise informal (es gibt keine Architekturlehre für den Aufbau großer SW-Systeme)
    - angemessene Abstraktion erforderlich
    - Evolution des Systems: Kontrolle der Abhängigkeiten!

„Die durch Abstraktion entstandenen Konstrukte der Informatik als Bedingungen möglicher Information sind zugleich die Bedingungen der möglichen Gegenstände der Information in den Anwendungen“ (H. Wedekind in Anlehnung an eine Aussage Kants aus der „Kritik der reinen Vernunft“)

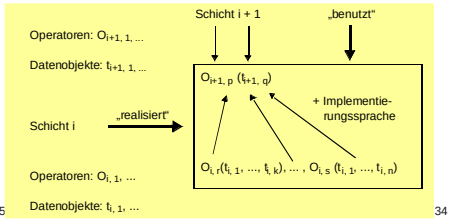
Vereinfacht ausgedrückt: Informatiker erfinden (konstruieren) abstrakte Konzepte; diese ermöglichen (oder begrenzen) wiederum die spezifischen Anwendungen.

Schichtenmodelle für DBS (3)

- Ziel: Architektur eines daten-unabhängigen DBS (Forts.)
  - **Aufbau in Schichten**
    - „günstige Zerlegung“ des DBS in „nicht beliebig viele“ Schichten
    - optimale Bedienung der Aufgaben der darüber liegenden Schicht
    - Implementierungs-unabhängige Beschreibung der Schnittstellen
  - **Empfohlene Konzepte**
    - Geheimnisprinzip (*Information Hiding*)
    - hierarchische Strukturierung
    - generische Auslegung der Schnittstellen: nur bestimmte Objekttypen mit charakteristischen Operationen sind vorgegeben, jedoch nicht ihre anwendungsbezogene Spezifikation und Semantik

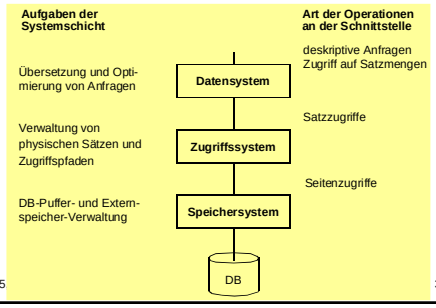
Schichtenmodelle für DBS (4)

- Ziel: Architektur eines daten-unabhängigen DBS (Forts.)
  - **Aufbauprinzip**
    - ‚Benutzt'-Relation: A benutzt B, wenn A B aufruft und die korrekte Ausführung von B für die vollständige Ausführung von A notwendig ist
    - Anzahl der Schichten: Entwurfskomplexität pro Schicht fällt, Laufzeitaufwand des DBS steigt mit wachsendem n



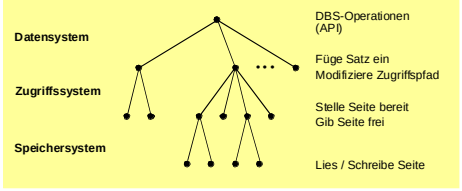
Schichtenmodelle für DBS (5)

- Ziel: Architektur eines daten-unabhängigen DBS (Forts.)
  - **Vereinfachtes Schichtenmodell**



Schichtenmodelle für DBS (6)

- Ziel: Architektur eines daten-unabhängigen DBS (Forts.)
  - **Vereinfachtes Schichtenmodell (Forts.)**



Schichtenmodelle für DBS (7)

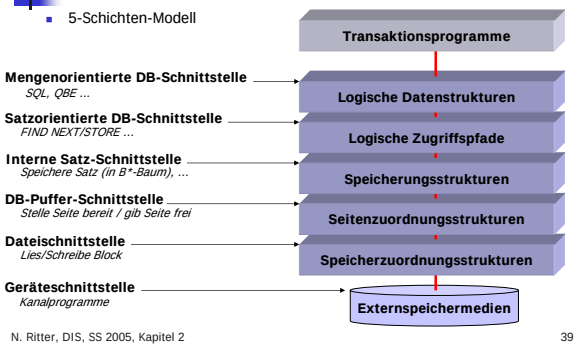
- Ziel: Architektur eines daten-unabhängigen DBS (Forts.)
  - Vorteile als Konsequenzen der Nutzung **hierarchischer Strukturen** und **der „benutzt“-Relation**
    - höhere Ebenen (Systemkomponenten) werden einfacher, weil sie tiefere Ebenen (Systemkomponenten) benutzen können
    - Änderungen auf höheren Ebenen sind ohne Einfluss auf tieferen Ebenen
    - höhere Ebenen können abgetrennt werden, tiefere Ebenen bleiben trotzdem funktionsfähig
    - tiefere Ebenen können getestet werden, bevor die höheren Ebenen lauffähig sind

Schichtenmodelle für DBS (8)

- Ziel: Architektur eines daten-unabhängigen DBS (Forts.)
  - Jede Hierarchieebene kann als abstrakte oder virtuelle Maschine aufgefasst werden
    - Programme der Schicht i benutzen als abstrakte Maschine die Programme der Schicht i-1, die als Basismaschine dient
    - abstrakte Maschine der Schicht i dient wiederum als Basismaschine für die Implementierung der abstrakten Maschine der Schicht i+1
  - Eine abstrakte Maschine entsteht aus der Basismaschine durch Abstraktion
    - einige Eigenschaften der Basismaschine werden verborgen
    - zusätzliche Fähigkeiten werden durch Implementierung höherer Operationen für die abstrakte Maschine bereitgestellt
  - Programme einer bestimmten Schicht können die der nächst tieferen Schicht genau so benutzen, als sei die untere Schicht Hardware

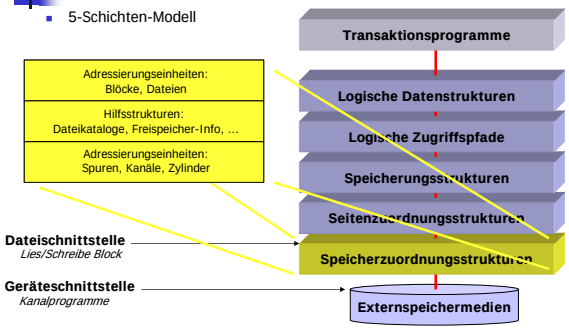
Schichtenmodelle für DBS (9)

- 5-Schichten-Modell



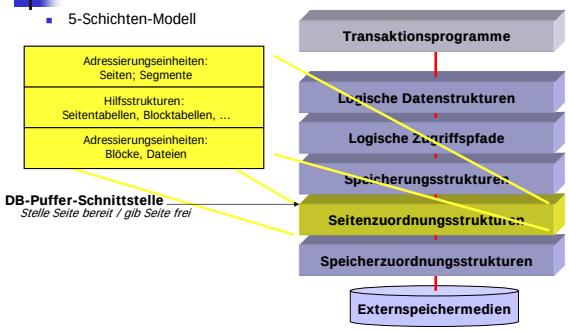
Schichtenmodelle für DBS (10)

- 5-Schichten-Modell



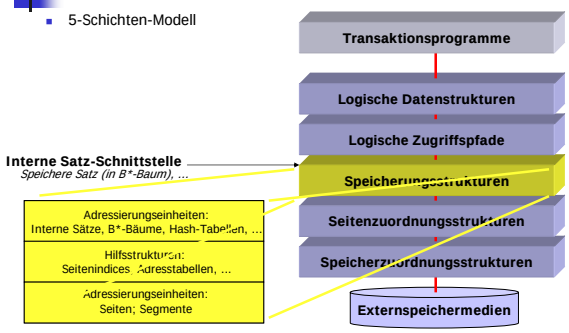
Schichtenmodelle für DBS (10)

- 5-Schichten-Modell



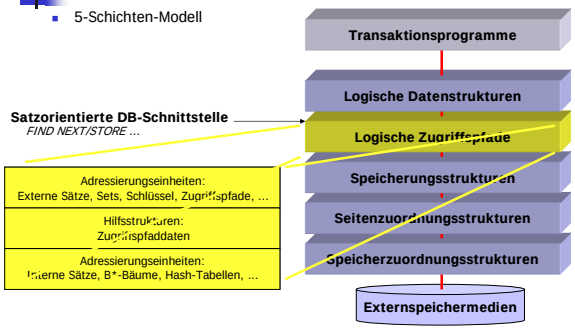
Schichtenmodelle für DBS (10)

- 5-Schichten-Modell



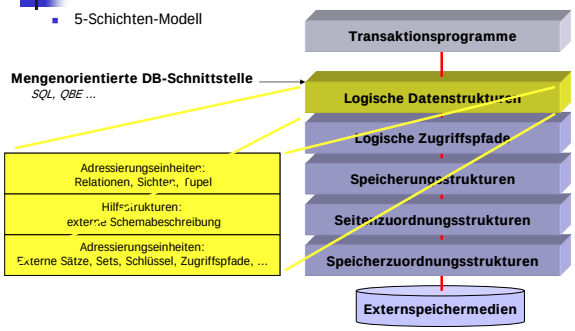
Schichtenmodelle für DBS (10)

- 5-Schichten-Modell

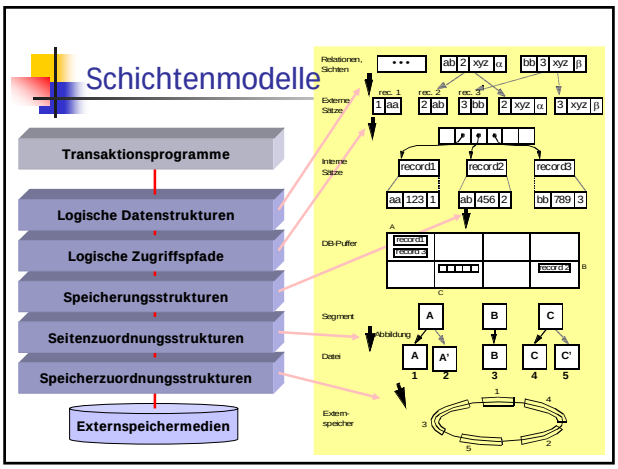


Schichtenmodelle für DBS (10)

- 5-Schichten-Modell



Schichtenmodelle



Schichtenmodelle für DBS (12)

- (5) Schichten repräsentieren Grade der Daten-Unabhängigkeit

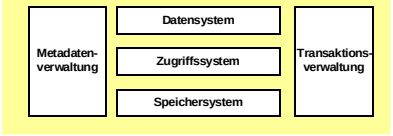
Benutzung von:	Was wird verborgen?
Mengenorientierte DB-Schnittstelle	Positionsanzeiger und explizite Beziehungs-konstrukte im Schema
Satzorientierte DB-Schnittstelle	Zahl und Art der physischen Zugriffspfade; interne Satzdarstellung
Interne Satzschnittstelle	DB-Pufferverwaltung; Recovery-Vorkehrungen
DB-Pufferschnittstelle	Dateiabbildung, Recovery-Unterstützung durch das BS
Dateischnittstelle	Technische Eigenschaften und Betriebs-details der externen Speichermedien

Schichtenmodelle für DBS (13)

- Weitere Komponenten in der DBS-Architektur
  - Entwurfsziel: *DBS sollen von ihrem Aufbau und ihrer Einsatzorientierung her in hohem Maße generische Systeme sein. Sie sind so zu entwerfen, dass sie flexibel durch Parameterwahl und ggf. durch Einbindung spezieller Komponenten für eine vorgegebene Anwendungsumgebung konfigurierbar sind.*
  - Metadaten
    - Metadaten enthalten Informationen über die zu verwaltenden Daten
    - sie beschreiben also diese Daten (Benutzerdaten) näher hinsichtlich Inhalt, Bedeutung, Nutzung, Integritätsbedingungen, Zugriffskontrolle usw.
    - die Metadaten lassen sich unabhängig vom DBVS beschreiben (siehe internes, konzeptionelles und externes Schema)
    - dadurch erfolgt das „Zuschneidern eines DBS" auf eine konkrete Einsatzumgebung; die Spezifikation, Verwaltung und Nutzung von Metadaten bildet die Grundlage dafür, dass DBS hochgradig „generische" Systeme sind
    - Metadaten fallen in allen DBS-Schichten an
    - Metadatenverwaltung, DB-Katalog, Data-Dictionary-System, DD-System, ...

Schichtenmodelle für DBS (14)

- Weitere Komponenten in der DBS-Architektur
  - Transaktionsverwaltung
    - Realisierung der ACID-Eigenschaften (Synchronisation, Logging/Recovery, Integritätssicherung)
  - Überblick

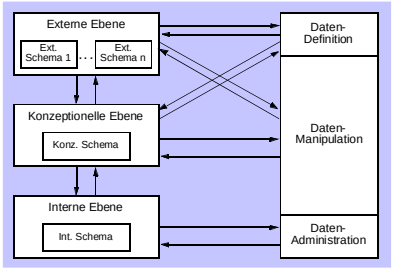


Schichtenmodelle für DBS (15)

- Drei-Schema-Architektur nach ANSI-SPARC
  - bisher Realisierungssicht (5-Schichtenmodell), nun Benutzungssicht

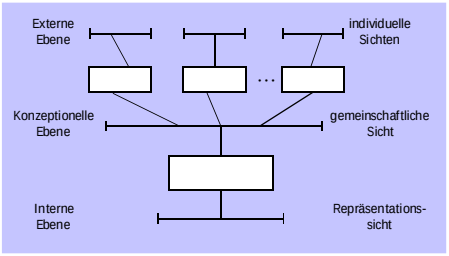
Tsichritzis, D. C., Klug, A.:
The ANSI/X3/Sparc DBMS Framework
Report of the Study Group on
Database Management Systems,
in: Information Systems 3:3,
1978, 173-191

ANSI: American National Standards
Institute, SPARC-Komitee:
Study Group on Database
Management Systems,
http://www.ansi.org



Schichtenmodelle für DBS (16)

- Drei-Schema-Architektur nach ANSI-SPARC (Forts.)



Schichtenmodelle für DBS (17)

- Drei-Schema-Architektur nach ANSI-SPARC (Forts.)
  - Konzeptionelles Schema
    - (zeitvariante) globale Struktur; neutrale und redundanzfreie Beschreibung in der Sprache eines spezifischen Datenmodells
    - Beschreibungssprache: DDL (*Data Definition Language*)
  - Externes Schema
    - Definition von zugeschnittenen Sichten auf Teile des konzeptionellen Schemas für spezielle Anwendungen (Benutzer)
    - Sichtenbildung
      - Anpassung der Datentypen an die der Wirtssprache (DBS ist „multi-lingual")
      - Zugriffsschutz
      - Reduktion der Komplexität
  - Internes Schema
    - legt physische Struktur der DB fest (Satzformate, Zugriffspfade etc.)
    - Beschreibungssprache: SSL (*Storage Structure Language*)

Schichtenmodelle für DBS (18)

- Drei-Schema-Architektur nach ANSI-SPARC (Forts.)
  - Beispiel

Extern (PL/1)

```
DCL   1 PERS,
      2 PNR CHAR(6),
      3 GEH FIXED BIN(31)
      4 NAME CHAR(20)
```

Extern (COBOL)

```
01    PERSC.
      02   PERSNR PIC X(6).
      02   ABTNR PIC X(4).
```

Konzeptionelles Schema

```
ANGESTELLTER
        ANG_NUMMER      CHARACTER   (6)
        NAME            CHARACTER   (20)
        ABT_NUMMER      CHARACTER   (4)
        GEHALT          NUMERIC     (5)
```

Internes Schema

```
SPEICHERUNG_PERS   LENGTH=40
        PREFIX     TYPE=BYTE(6), OFFSET=0
        PNR        TYPE=BYTE(6), OFFSET=6, INDEX=PNR
        NAME       TYPE=BYTE(20), OFFSET=12
        ANR        TYPE=BYTE(4), OFFSET=32
        GEHALT     TYPE=FULLWORD, OFFSET=36
```


Schichtenmodelle für DBS (19)

- Dynamischer Ablauf (Forts.)
  - Interne Bearbeitungsschritte
    1. **SELECT * FROM PERS' WHERE PNR = '12345'**
    2. Vervollständigen der Verarbeitungsinformation aus Konzeptionellem und Internem Schema; Ermittlung der Seiten# (z. B. durch Hashing)
    3. Zugriff auf DB-Puffer: erfolgreich oder
    4. Zugriff auf DB über DB-Pufferverwaltung/Betriebssystem
    5. Durchführen des E/A-Auftrages
    6. Ablegen der Seite im DB-Puffer
    7. Übertragen in Arbeitsbereich
    8. Statusinformation: Return-Code, Cursor-Info
    9. Manipulation mit Anweisungen der Programmiersprache

Zusammenfassung

- DBS-Charakteristika
  - Zentralisierte Verwaltung der operationalen Daten (Rolle des DBA)
  - Adäquate Schnittstellen (Datenmodell und DB-Sprache)
  - Datenkontrolle, insbes. zentrale Kontrolle der Datenintegrität und kontrollierter Mehrbenutzerbetrieb
  - Leistung und Skalierbarkeit
  - Hoher Grad an Daten-Unabhängigkeit
- Beschreibungsmodelle für ein DBS
  - Schichtenmodell
  - Drei-Schema-Architektur
- Programmierschnittstellen (siehe Schichtenmodell)
  - mengenorientierte DB-Schnittstelle (relationale DBS)
  - satzorientierte DB-Schnittstelle (hierarchische und netzwerkartige DBS)
  - interne Satzschnittstelle (zugriffsmethodenorientierte Programmierschnittstelle)