

Transaktionale Informationssysteme

1. DB-Grundbegriffe und TA-Konzept

Norbert Ritter

Datenbanken und Informationssysteme
vsis-www.informatik.uni-hamburg.de

© N. Ritter

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 integritätserschützend (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	Ausprägungen				
ANGESTELLTER	PNR	NAME	TAETIGKEIT	GEHALT	ALTER
Satztyp (Relation)	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 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 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 DBS (3)

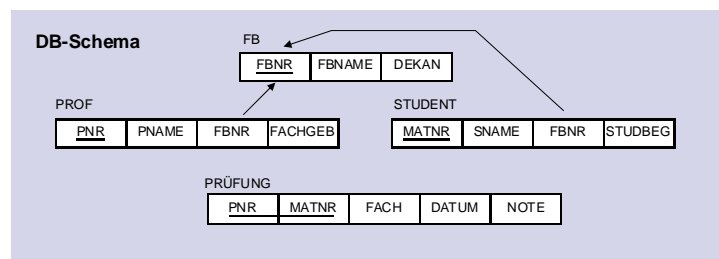
2. Leichte Handhabbarkeit der Daten (Forts.)

- **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 DBS (4)

2. Leichte Handhabbarkeit der Daten (Forts.)

- **Beispiel im Relationenmodell**
 - **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



Anforderungen an DBS (5)

2. Leichte Handhabbarkeit der Daten (Forts.)

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

- 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';
```

- 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');
```

Anforderungen an DBS (6)

2. Leichte Handhabbarkeit der Daten (Forts.)

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

- Anfragebeispiele (Forts.)

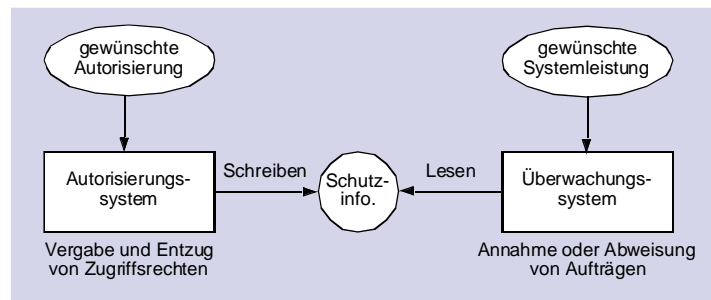
- III. Finde die Durchschnittsnoten der DV-Prüfungen für alle Fachbereiche mit mehr als 1000 Studenten.

```
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 DBS (7)

3. Kontrolle der Datenintegrität

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



Anforderungen an DBS (8)

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 DBS (9)

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
 - **Dauerhaftigkeit (Persistenz):** Effekte von abgeschlossenen TA gehen nicht verloren
 - **Atomarität (Resistenz):** Zustandsänderungen werden entweder, wie in der TA spezifiziert, vollständig durchgeführt oder überhaupt nicht
- Im Mehrbenutzerbetrieb entsteht durch nebenläufige TA ein Konkurrenzverhalten um gemeinsame Daten, d. h., TA geraten in Konflikt
 - **Isolationseigenschaft:** TA-Konflikte sind zu verhindern oder aufzulösen

Anforderungen an DBS (10)

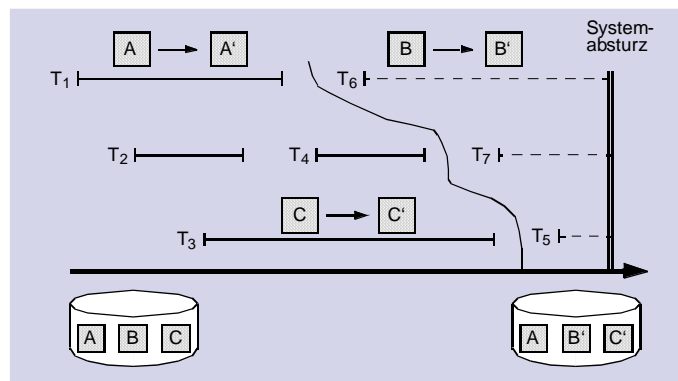
3. Kontrolle der Datenintegrität (Forts.)

- **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 DBS (11)

3. Kontrolle der Datenintegrität (Forts.)

- **Zu: Erhaltung der physischen Datenintegrität**



Anforderungen an DBS (12)

3. Kontrolle der Datenintegrität (Forts.)

- **Zu: Erhaltung der physischen Datenintegrität (Forts.)**
 - **DBMS garantiert physische Datenintegrität (siehe Beispiel)**
 - 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 (T₁ bis T₄) und sonst keine
 - automatische Wiederherstellung nach Neustart des Systems
 - **Maßnahmen beim Wiederanlauf (siehe Beispiel)**
 - Ermittlung der beim Absturz aktiven Transaktionen (T₅, T₆, T₇)
 - 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 DBS (13)

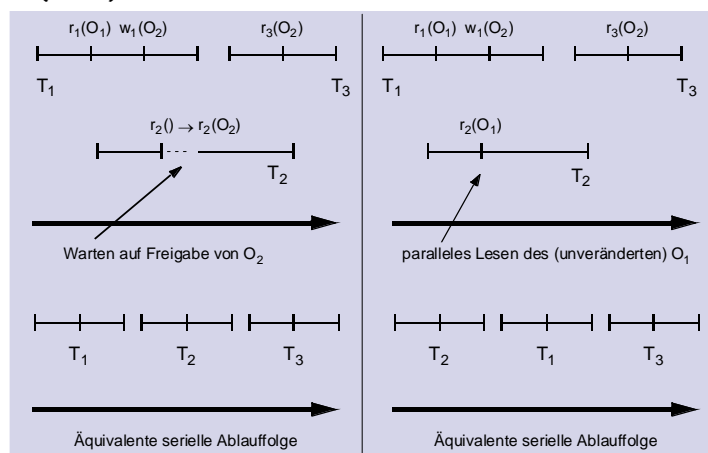
3. Kontrolle der Datenintegrität (Forts.)

- Zu: Notwendigkeit der Kontrolle des Mehrbenutzerbetriebs**
 - Beim **logischen Einbenutzerbetrieb** hat jede der parallel aktiven Transaktionen den ‚Eindruck‘, als lief 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 DBS (14)

3. Kontrolle der Datenintegrität (Forts.)

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



Anforderungen an DBS (15)

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 DBS (16)

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 DBS (17)

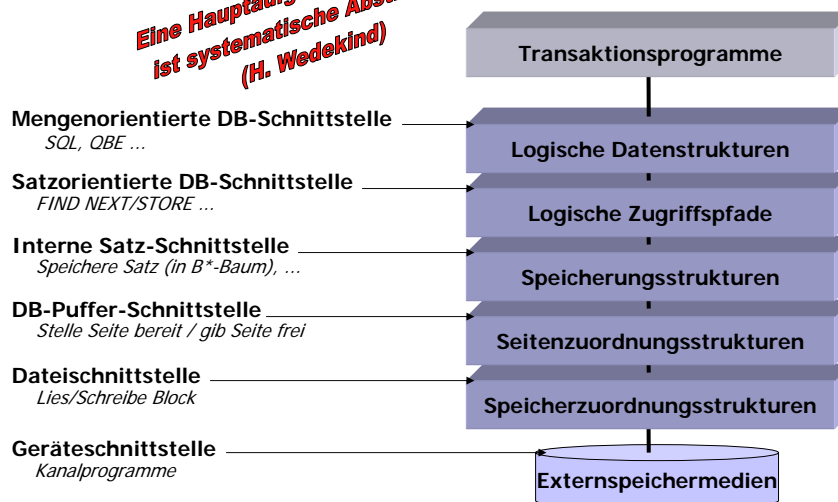
5. Hoher Grad an Daten-Unabhängigkeit

- **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

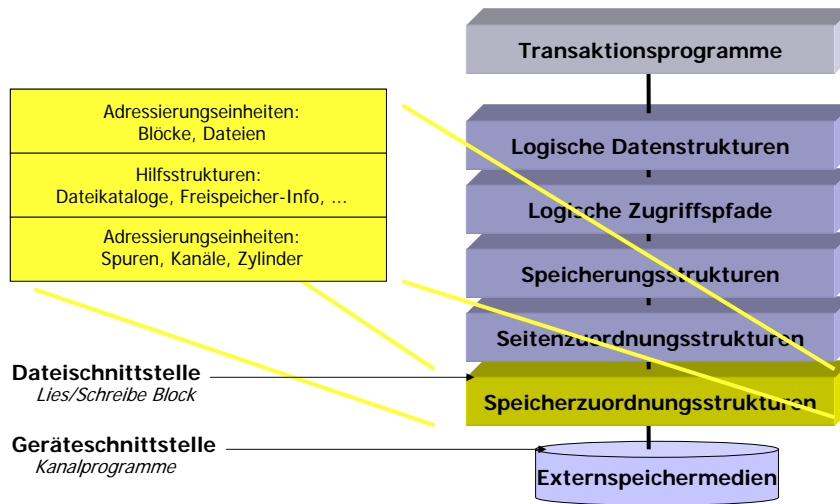
5-Schichtenmodell (1)

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

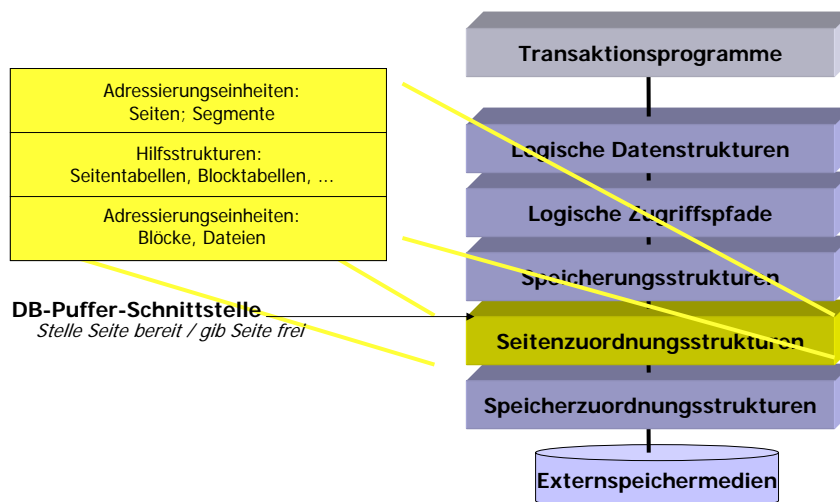
*Eine Hauptaufgabe der Informatik
ist systematische Abstraktion.
(H. Wedekind)*



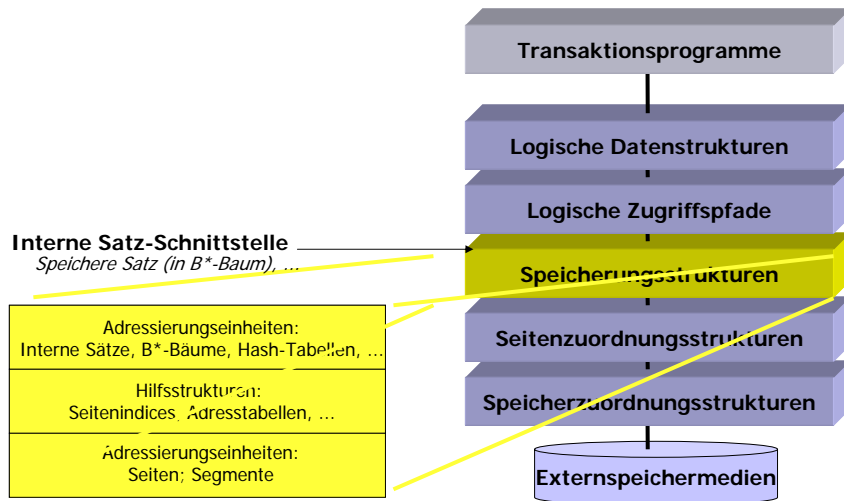
5-Schichtenmodell (2)



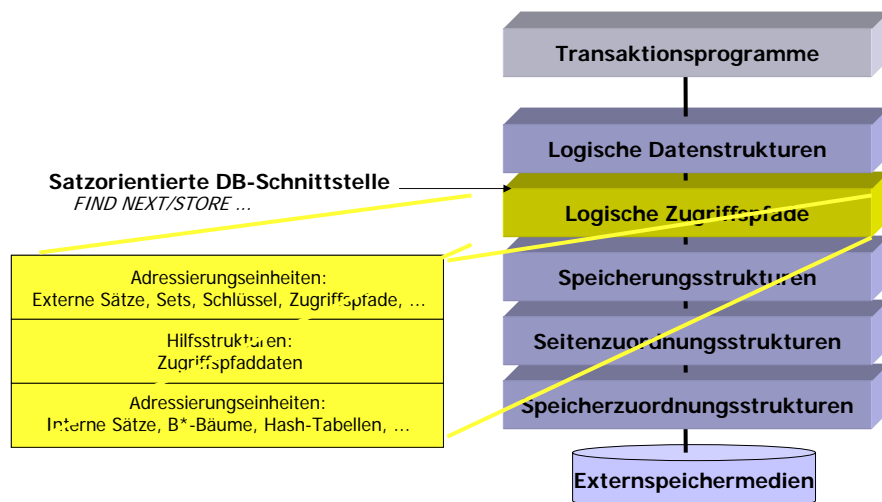
5-Schichtenmodell (3)



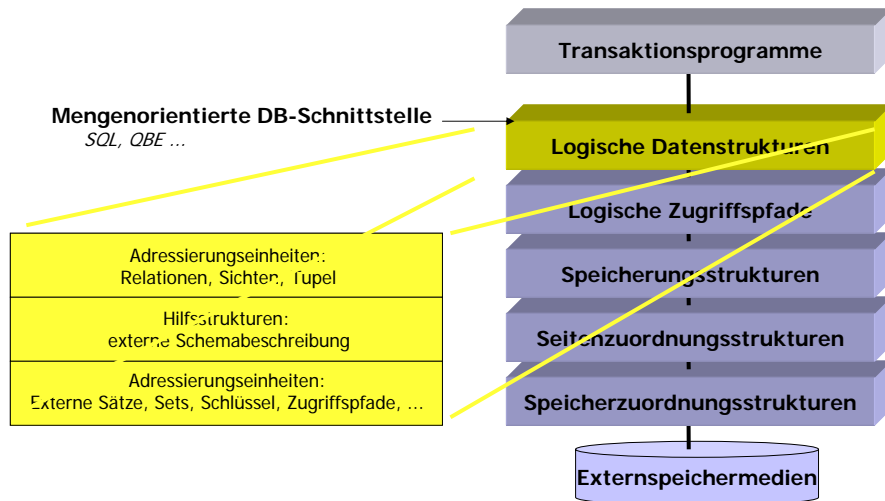
5-Schichtenmodell (4)



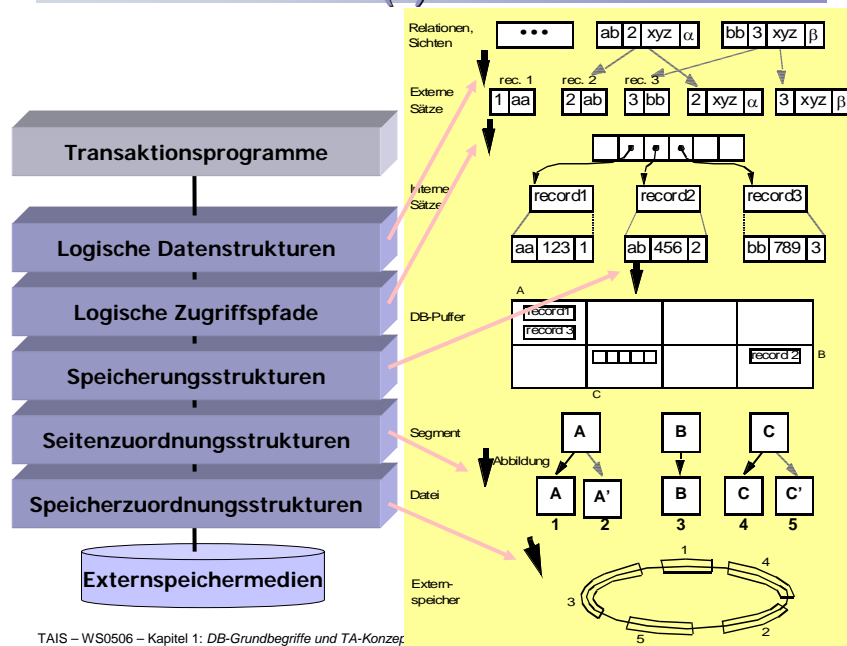
5-Schichtenmodell (5)



5-Schichtenmodell (6)



5-Schichtenmodell (7)



5-Schichtenmodell (8)

- Schichten repräsentieren Grade der Datenunabhängigkeit

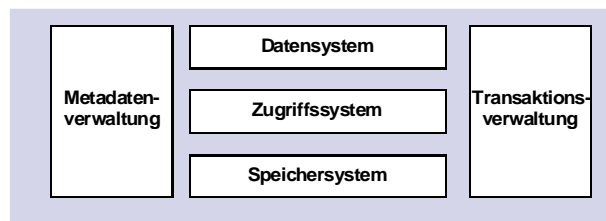
Benutzung von:	Was wird verborgen?
Mengenorientierte DB-Schnittstelle	Positionsanzeiger und explizite Beziehungskonstrukte im Schema
Satzorientierte DB-Schnittstelle	Zahl und Art der physischen Zugriffspfade; interne Satzdarstellung
Interne Satzschnittstelle	DB-Pufferverwaltung; Recovery-Vorkehrungen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

5-Schichtenmodell (9)

- 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 „Zuschneiden 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, ...

5-Schichtenmodell (10)

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



TA-Konzept (1)

- Gefährdung der DB-Konsistenz

	<i>Korrektheit der Abbildungshierarchie</i>	<i>Übereinstimmung zwischen DB und Miniwelt</i>
<i>Durch das Anwendungs- programm</i>	Mehrbenutzeranomalien Synchronisation	Unzulässige Änderungen Integritätsüberwachung des DBVS TA-orientierte Verarbeitung
<i>Durch das DBVS und die Betriebsumgebung</i>	Fehler auf den Externspeichern Fehlertolerante Implementierung Archivkopien (Backup)	Undefinierter DB-Zustand nach einem Systemausfall TA-orientierte Fehlerbehandlung

TA-Konzept (2)

- Ablaufkontrollstruktur: Transaktion (TA)
 - *Eine Transaktion ist eine ununterbrechbare Folge von DML-Befehlen, die die Datenbank von einem logisch konsistenten in einen (neuen) logisch konsistenten Zustand überführt.*
 - Beispiel eines TA-Programms:

```
BOT
UPDATE Konto
...
UPDATE Schalter
...
UPDATE Zweigstelle
...
INSERT INTO Ablage (...)
COMMIT
```

TA-Konzept (3)

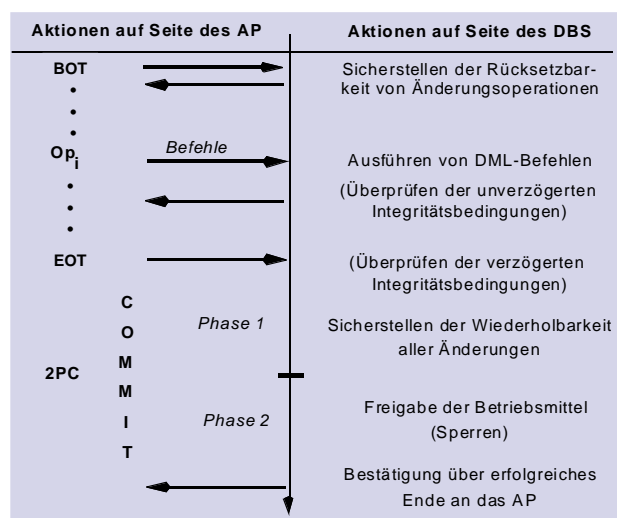
- Ablaufkontrollstruktur: Transaktion (Forts.)
 - **ACID**-Eigenschaften von Transaktionen
 - **Atomicity (Atomarität)**
 - TA ist kleinste, nicht mehr weiter zerlegbare Einheit
 - Entweder werden alle Änderungen der TA festgeschrieben oder gar keine („alles-oder-nichts“-Prinzip)
 - **Consistency**
 - TA hinterlässt einen konsistenten DB-Zustand, sonst wird sie komplett (siehe Atomarität) zurückgesetzt
 - Zwischenzustände während der TA-Bearbeitung dürfen inkonsistent sein
 - Endzustand muss alle definierten Integritätsbedingungen erfüllen

TA-Konzept (4)

- Ablaufkontrollstruktur: Transaktion (Forts.)
 - **ACID**-Eigenschaften von Transaktionen (Forts.)
 - **Isolation**
 - Nebenläufig (parallel, gleichzeitig) ausgeführte TA dürfen sich nicht gegenseitig beeinflussen
 - Parallele TA bzw. deren Effekte sind nicht sichtbar (logischer Einbenutzerbetrieb)
 - **Durability (Dauerhaftigkeit)**
 - Wirkung erfolgreich abgeschlossener TA bleibt dauerhaft in der DB
 - TA-Verwaltung muss sicherstellen, dass dies auch nach einem Systemfehler (HW- oder System-SW) gewährleistet ist
 - Wirkung einer erfolgreich abgeschlossenen TA kann nur durch eine sog. kompensierende TA aufgehoben werden

TA-Konzept (5)

- Schnittstelle zwischen Anwendungsprogramm (AP) und DBS

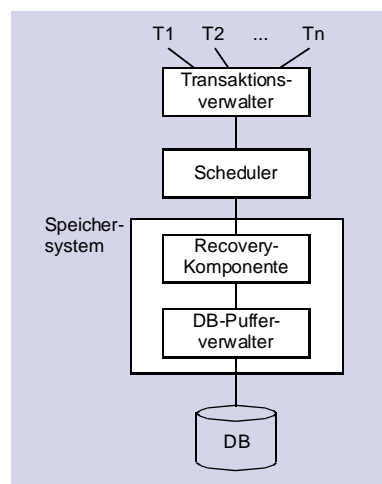


TA-Konzept (6)

- Wesentliche Abstraktionen (aus Sicht der DB-Anwendung) zur Gewährleistung einer ‚fehlerfreien Sicht‘ auf die Datenbank im logischen Einbenutzerbetrieb
 - Alle Auswirkungen auftretender Fehler bleiben der Anwendung verborgen (*failure transparency*)
 - Es sind keine anwendungsseitigen Vorkehrungen zu treffen, um Effekte der Nebenläufigkeit beim DB-Zugriff auszuschließen (*concurrency transparency*)
- TA-Verwaltung
 - koordiniert alle DBS-seitigen Maßnahmen, um ACID zu garantieren
 - besitzt zwei wesentliche Komponenten
 - Synchronisation
 - Logging und Recovery
 - kann zentralisiert oder verteilt (z.B. bei VDBS) realisiert sein
 - soll Transaktionsschutz für heterogene Komponenten bieten

TA-Konzept (7)

- Abstraktes Architekturmodell (für das Read/Write-Modell auf Seitenbasis)

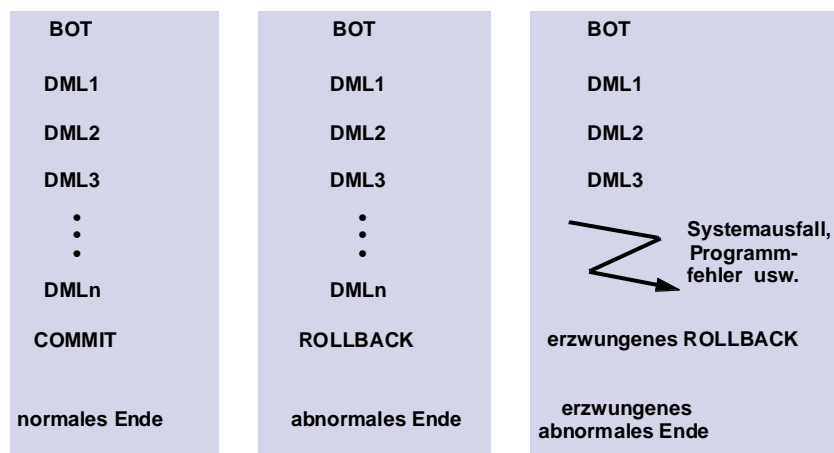


TA-Konzept (8)

- Komponenten (vgl. Architekturmodell vorangegangene Folie)
 - **Transaktionsverwalter**
 - Verteilung der DB-Operationen in VDBS und Weiterreichen an den Scheduler
 - zeitweise Deaktivierung von TA (bei Überlast)
 - Koordination der Abort- und Commit-Behandlung
 - **Scheduler** (Synchronisation)
kontrolliert die Abwicklung der um DB-Daten konkurrierenden TA
 - **Recovery**-Komponente
sorgt für die Rücksetzbarkeit/Wiederholbarkeit der Effekte von TA
 - **DB-Pufferverwalter**
stellt DB-Seiten bereit und gewährleistet persistente Seitenänderungen

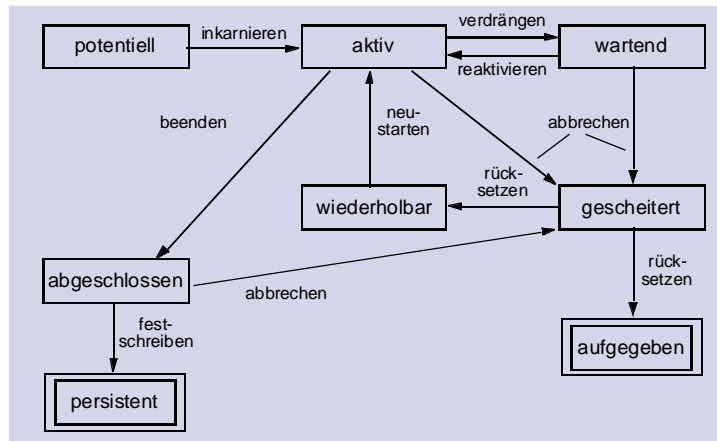
TA-Konzept (9)

- Transaktionsablauf
 - Mögliche Ausgänge einer Transaktion



TA-Konzept (10)

- Zustände einer Transaktion
 - Zustandsübergangsdiagramm – Kontrolle durch TA-Verwaltung



TA-Konzept (11)

- Zustände einer Transaktion (Forts.)
 - **potentiell**
 - TAP wartet auf Ausführung
 - Beim Start werden, falls erforderlich, aktuelle Parameter übergeben
 - **aktiv**
 - TA konkurriert um Betriebsmittel und führt Operationen aus
 - **wartend**
 - Deaktivierung bei Überlast
 - Blockierung z.B. durch Sperren
 - **abgeschlossen**
 - TA kann sich (einseitig) nicht mehr zurücksetzen
 - TA kann jedoch noch scheitern (z.B. bei Konsistenzverletzung)
 - **persistent** (Endzustand)
 - Wirkung aller DB-Änderungen werden dauerhaft garantiert

TA-Konzept (12)

- Zustände einer Transaktion (Forts.)
 - **gescheitert**
 - Vielfältige Ereignisse können zum Scheitern ein TA führen (siehe Fehlermodell, Verklemmung usw.)
 - **wiederholbar**
 - Gescheiterte TA kann ggf. (mit denselben Eingabewerten) erneut ausgeführt werden
 - **aufgegeben** (Endzustand)

Zusammenfassung (1)

- DBS-Charakteristika
 - Zentralisierte Verwaltung der operationalen Daten (Rolle des DBA)
 - Adäquate Schnittstellen (Datenmodell und DB-Sprache)
 - Datenkontrolle, insbes. zentrale Kontrolle der Datenintegrität, kontrollierter Mehrbenutzerbetrieb und Fehlerbehandlung
 - Leistung und Skalierbarkeit
 - Hoher Grad an Daten-Unabhängigkeit
- 5-Schichtenmodell als Beschreibungsmodell für ein DBS
 - Ausblick: Synchronisationsverfahren/Recovery-Mechanismen arbeiten auf
 - Seiten-Ebene
 - Objekt-Ebene

Zusammenfassung (2)

- Transaktionsparadigma (ACID)
 - Verarbeitungsklammer für die Einhaltung von semantischen Integritätsbedingungen
 - Verdeckung von (erwarteten) Fehlerfällen (*failure isolation*)
 - Logging/Recovery
 - Verdeckung der Nebenläufigkeit (*concurrency isolation*)
 - Synchronisation