

## Lektion 2: Datenorganisation, Speicherung

Lehrbuch Kapitel 1 & 2

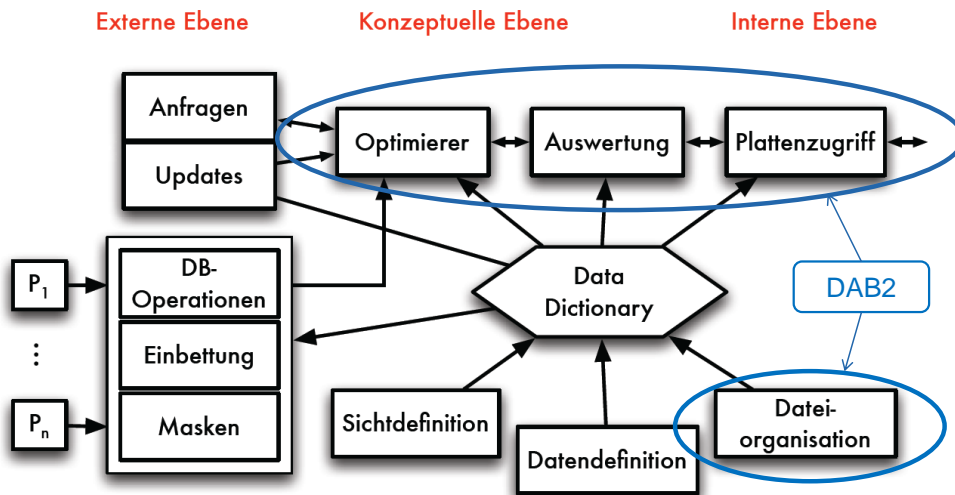
1 L	Einführung
4 L	Datenorganisation Speicherung
4 L	Optimierung
2 L	Transaktionen, Recovery
2 L	Non-Standard Datenbanken
1 L	Repetition, Abschluss

← "You are here"

- Begriffe:
  - Datenbanksystem
  - Informationssystem
- 3-Ebenen-Architektur
- Ziele des physischen Entwurfes
- Stored Procedures & Trigger

- Themen, die in DAB2 behandelt werden, einordnen können
- 5-Schichten-Modell eines RDBMS kennen
- Begriff der Speicherhierarchie kennen

## 3-Ebenen-Architektur (erweitert)



In der letzten Vorlesung wurde die Drei-Ebenen-**Schema**architektur eingeführt. Neben dieser Architektur wurde durch ANSI-SPARC auch eine Drei-Ebenen-**System**architektur entwickelt. In DAB2 werden vor allem die blau umrandeten Komponenten betrachtet.

Während in der Datenmodellierung die Drei-Ebenen-Schemaarchitektur alltäglich ist, wird für die Beschreibung der Systemarchitektur von Datenbanksystemen häufiger die Fünf-Schichten-Architektur verwendet. Auch wir werden diese verwenden. Sie wird ab der nächsten Folie eingeführt und begleitet uns durch die DAB2-Vorlesung.

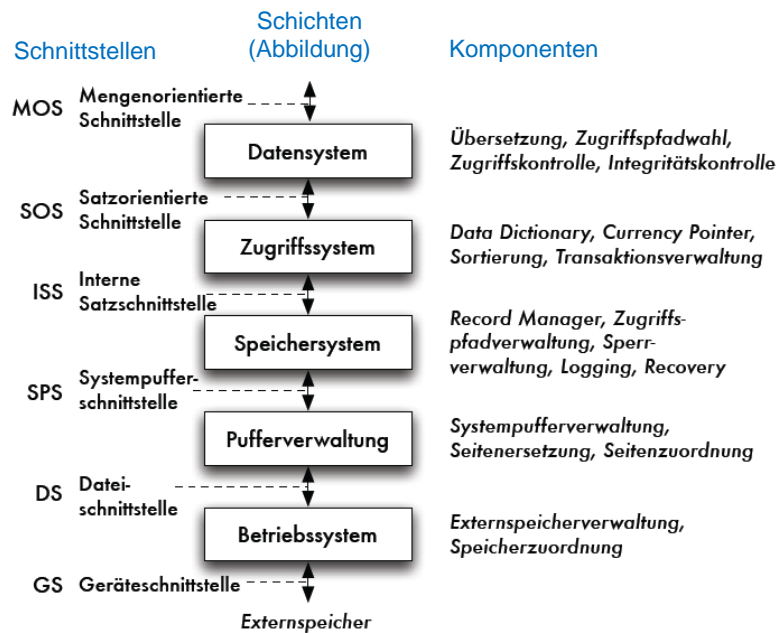
- Zweck: **Modell** für die **Systemarchitektur** eines RDBMS
- Bewährt (Nützlichkeit nachgewiesen in Forschungsprototypen), Grundlage heutiger RDBMS
- Beschreibt die in einem RDBMS enthaltenen Transformationskomponenten und Schnittstellen:
  - Schrittweise Transformation von Anfragen/Änderungen, bis hin zu Zugriffen auf Speichermedien/Hardware
  - Definition der Schnittstellen zwischen Komponenten
- Bemerkung: Andere Datenbanktechnologien erfordern andere Architekturen (XML-DB, NoSQL-DB, ...)

Diese Architektur hat sich bewährt (erste Nützlichkeit nachgewiesen im System R) und ist auch Grundlage heutiger RDBMS.

Die Architektur beschreibt die in einem RDBMS enthaltenen Transformationskomponenten und Schnittstellen:

- Komponenten: Transformieren die Anfragen/Änderungen schrittweise bis hin zu Zugriffen auf die Speichermedien/Hardware.
- Schnittstellen: Die Architektur beinhaltet 6 Schnittstellen. Eine Schnittstelle wird jeweils durch die darunter liegende Komponente implementiert.

# 5-Schichten-Architektur



Zürcher Fachhochschule

7

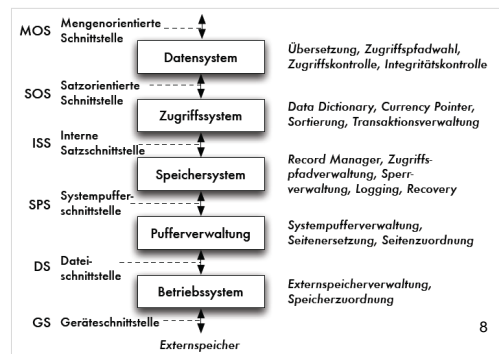
In den nachfolgenden Folien werden zunächst die Schnittstellen und anschliessend die Komponenten kurz betrachtet. Wir werden diese in späteren Vorlesungen detaillierter behandeln.

Diese Schichtung ist zunächst theoretisch, es ist ein Vorschlag zur Gliederung. Die Datenbankhersteller halten sich natürlich nicht an diese Struktur, sondern ordnen und verbinden diese Aufgaben und Komponenten so, dass eine möglichst performante Lösung möglich ist.

Bemerkung am Rande: Andere Datenbanktechnologien erfordern natürlich andere/erweiterte Architekturen (z.B. XML-DB, NoSQL-DB, ...)

# 5-Schichten-Architektur: Schnittstellen

- **MOS: Mengenorientierte Schnittstelle**  
Deklarative DML/DQL auf Tabellen, Sichten, Tupel.
- **SOS: Satzorientierte Schnittstelle**  
Externe Sätze, logische Dateien, logische Zugriffspfade. Navigierender Zugriff.
- **ISS: Interne Satzschnittstelle**  
Interne Sätze, Zugriffspfade. Manipulation von Sätzen und Zugriffspfaden.
- **SPS: Pufferschnittstelle**  
Seiten, Seitenadressen. Freigeben und Bereitstellen.
- **DS: Datei- oder Seitenschnittstelle**  
Hole Seite, Schreibe Seite.
- **GS: Geräteschnittstelle**  
Spuren, Zylinder, Armbewegungen.



Zürcher Fachhochschule

8

Hier werden die Schnittstellen und typische Funktionen dieser Schnittstellen gezeigt:

1. **Mengenorientierte Schnittstelle:** Stellt eine mengenorientierte, deklarative (Beschreibung des Problems steht im Vordergrund) Sprache zur Manipulation der Daten und Strukturen zur Verfügung. Es wird z.B. spezifiziert, was gesucht werden soll, es wird aber nicht angegeben, wie gesucht werden soll. Eine typische Sprache dafür ist SQL.
2. **Satzorientierte Schnittstelle:** Mittels dieser Schnittstelle können die Daten satzorientiert bearbeitet werden. Die Daten werden mittels navigierendem Zugriff verarbeitet. Zur Suche der Daten werden logische Zugriffspfade, Indexe verwendet.
3. **Interne Satzschnittstelle:** Diese Schnittstelle verarbeitet Datensätze einheitlich, ohne die unterschiedlichen Relationentypen zu berücksichtigen (auf dieser Ebene ist der Bezug zur inneren, logischen Struktur der Datensätze aus dem Datenmodell verloren). Die logischen Zugriffspfade (Indexe) werden mit geeigneten Speicherstrukturen (z.B. B<sup>+</sup>-Bäume, Hashtabellen) implementiert.
4. **Systempufferschnittstelle:** Die Pufferschnittstelle stellt Operationen zum Lesen und Speichern von Seiten auf dem virtuellen, linearen Adressraum zur Verfügung
5. **Datei- oder Seitenschnittstelle:** Die Dateischnittstelle stellt Operationen zur Verfügung, um Blöcke von Daten auf den Computer zu verarbeiten. Während die Seiten der Systempufferschnittstelle die 'logischen' Einheit des DBMS darstellen, sind die Blöcke die Einheiten des Betriebssystems.
6. **Geräteschnittstelle:** Die Geräteschnittstelle ergibt sich durch die eingesetzte Hardware (Harddisk, SSD, etc.).



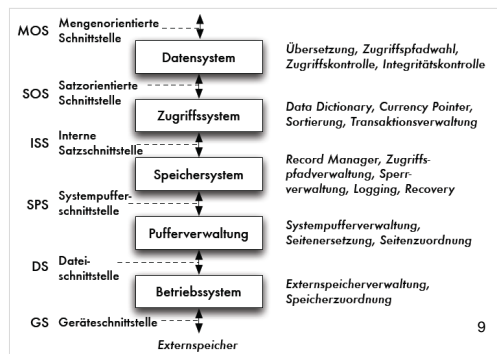
# 5-Schichten-Architektur: Begriffe

Intern/extern/logisch Tupel/Datensatz?

Je nach Schicht werden unterschiedliche Begriffe verwendet:

1. Datensystem: (externe) Tupel mit Attributen
2. Zugriffssystem: logische/externe Datensätze, interne Tupel mit Feldern
3. Speichersystem: interne Datensätze (Bezug zu Datenmodell verloren)
4. Pufferverwaltung: Seiten («Byte-Container»)

Struktur	Systemkomponente
Tupel	Datensystem
internes Tupel oder logischer Datensatz	Zugriffssystem
interner Datensatz	Speichersystem
Seite	Pufferverwaltung

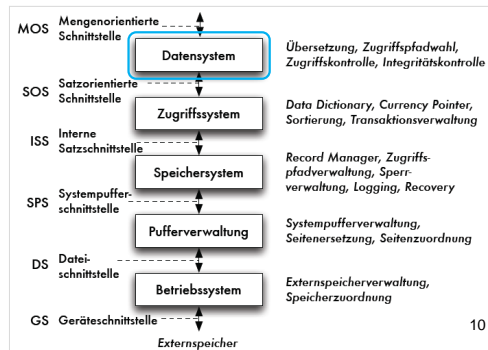


Zürcher Fachhochschule

9

Diese Begriffe werden leider häufig nicht konsistent verwendet, was oft zu Verwirrung führt.

- **Abbildung:** Wandelt Anfragen an MOS (mengenorientierte Schnittstelle) in Anfragen an SOS (satzorientierte Schnittstelle) um
- **Objekte:** (externe) Tupel und Relationen (Menge von Tupel), Sichten (Views)
- **Komponenten:** Übersetzung und Optimierung von mengenorientierten Anfragen in navigierende Zugriffe auf interne Tupel (=externe/logische Datensätze), Zugriffs- und Integritätskontrolle
- **Operationen MOS (z.B. in SQL)**
  - SELECT \* FROM <Tab.Name>
  - UPDATE <Tab.Name> ...
  - ...



Zürcher Fachhochschule

10

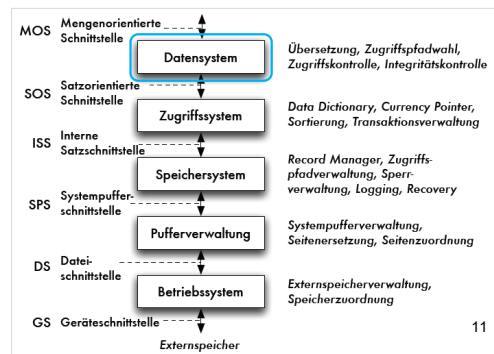
In den folgenden Folien werden jetzt die einzelnen Schichten kurz eingeführt. Jede Schicht besteht aus mehreren Komponenten, respektive Aufgaben, welche die Abbildung der nach aussen präsentierten Schnittstelle auf die darunterliegende, benutzte Schnittstelle implementieren.

Für jede Schicht sind die typischen Abbildungen, Objekte, Aufgaben/Komponenten und Operationen aufgeführt:

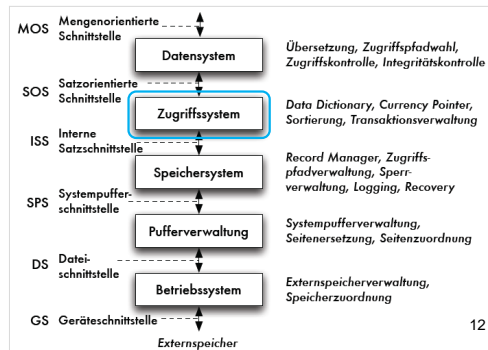
- **Abbildung:** Welche Operationen werden durch die Schicht mittels Operationen der darunterliegenden Schicht realisiert.
- **Objekt:** Zeigt die in der Schicht verwalteten Objekttypen auf.
- **Aufgaben/Komponenten:** Zeigt die Aufgaben, resp. Komponenten, welche die Schicht wahrnimmt, resp. aus denen sie besteht.
- **Operationen:** Beispiele für durch die Schicht in der Schnittstelle zur Verfügung gestellten Operationen.

Mengenorientierte Anfrage (SQL) muss bei Übersetzung durch System **optimiert** werden:

- Umformung des Anfrageausdrucks in einen effizienter zu bearbeitenden Ausdruck: Query Rewriting, konzeptuelle oder **logische Optimierung**.
- Auswahl der zur Anfragebearbeitung sinnvollen Zugriffspfade und Auswertungsalgorithmen für jeden relationalen algebraischen Operator: **physische Optimierung**.
- Auswahl des effizientesten Anfrageausdrucks nach Kostenschätzungen: **kostenbasierte Auswahl**



- **Abbildung:** Wandelt Anfragen an SOS in Anfragen in Anfragen an ISS (interne Satzschnittstelle) um.
- **Objekte:** interne Tupel/logische Datensätze (typisiert, unterschiedliche Relationstypen), logische Zugriffspfade (Indexe) und temporäre Zugriffspfade
- **Komponenten:** Verwaltung interne Tupel, Umwandlung Index-Operationen auf B\*-Bäumen und Hashtabellen, Transaktionsverwaltung, Data Dictionary, Currency Pointer (Cursor), Sortierung
- **Operationen SOS:**
  - FIND NEXT <satzname>
  - STORE <satzname>
  - ...



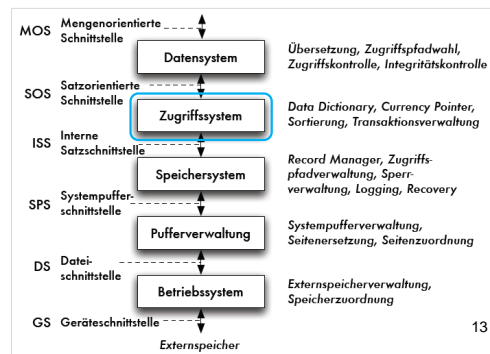
Zürcher Fachhochschule

12

**Abbildung Zugriffssystem:** Wandelt Anfragen an SOS in Anfragen in Anfragen an ISS (interne Satzschnittstelle) um.

Ein wichtige Aufgabe des Zugriffssystems ist die Abstraktion von internen Datensätzen und die Abstraktion des Zugriffs auf die Daten.

- Zugriffssystem abstrahiert von interner Darstellung der Datensätze (ISS):
  - Logische Datensätze, interne Tupel können Elemente der konzeptuellen Relation oder Elemente eines Zugriffspfads sein
  - Interne Tupel bestehen aus Feldern (entsprechen Attributen bei konzeptuellen Tupeln, der Bezug zum Datenmodell ist vorhanden)
  - Operationen im Zugriffssystem sind typischerweise Scans (interne Cursor auf Dateien oder Zugriffspfaden)
- Zugriffspfad oder -struktur:
  - Datenstruktur für zusätzlichen, schlüsselbasierten Zugriff auf Tupel (<Schlüsselattributwert, Tupeladr.>)
  - Meist als Indexdatei realisiert: Zugriffspfad auf eine Datei ist selbst wieder Datei
  - Primär- vs. Sekundärindex

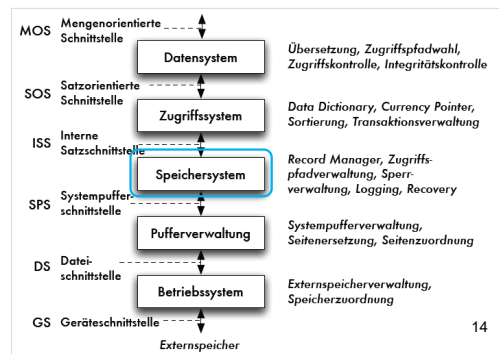


13

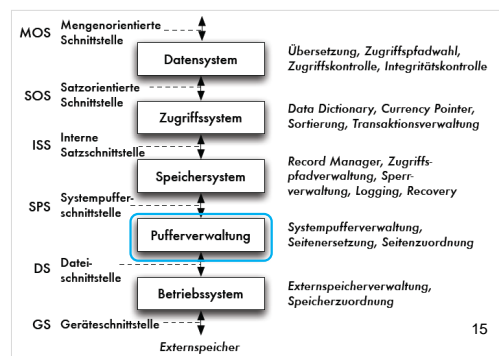
- Im SQL-Server ist der Zugriffspfad im selben File wie die Daten selbst
- Primär-Index: Index über Primärschlüssel
- Sekundärindex: Index über beliebige andere Attribute

# 5-Schichten-Architektur: Speichersystem

- **Abbildung:** Wandelt Anfragen an ISS in Anfragen an SPS (Systempufferschnittstelle) um.
- **Objekte:** einheitliche, interne Datensätze (ohne Typisierung, Bezug zu Datenmodell verloren), Bäume und Hashstrukturen
- **Komponenten:** Datenstrukturen und Operationen auf interne Seiten eines virtuellen linearen Adressraums umsetzen, Record-Manager, Verwaltung Zugriffspfade, Sperrverwaltung, Logging, Recovery
- **Operationen ISS:**
  - Speichere internen Datensatz
  - Füge Eintrag in B\*-Baum ein
  - ...

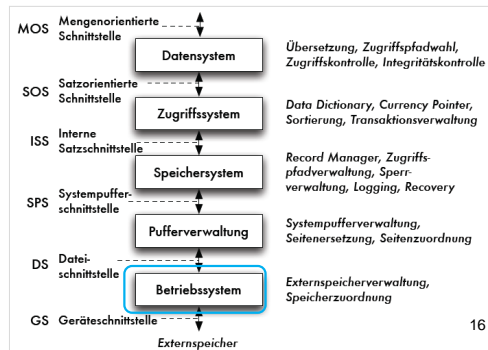


- **Abbildung:** Wandelt Anfragen an SPS in Anfragen an DS (Dateischnittstelle) um.
- **Objekte:** interne Seiten, Seitenadressen
- **Komponenten:** Systempufferverwaltung (interne Seiten auf Blöcke der DS abbilden), Seitenersetzung (Seitenwechselstrategie), Seitenzuordnung
- **Operationen SPS:**
  - Freigeben von Seiten
  - Bereitstellen/Suchen von Seiten
  - Seite sperren
  - ...



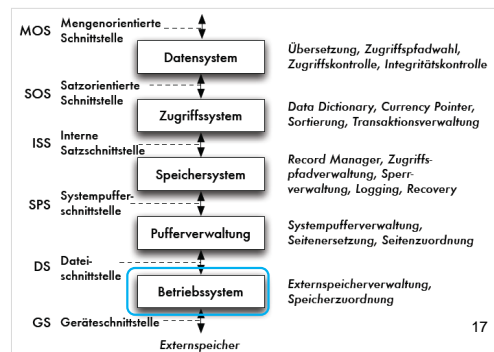
Unterschied zur Pufferverwaltung (Cache) des Betriebssystems: Der Cache des Betriebssystems und der Puffer des Datenbanksystems werden nach unterschiedlichen Strategien bewirtschaftet. Das Datenbanksystem nutzt Kenntnisse der auszuführenden Operationen um den Puffer optimaler verwenden zu können. Für die Performance von Datenbanksystemen ist daher eine eigenen Pufferverwaltung wichtig.

- **Abbildung:** Wandelt Anfragen an DS in Anfragen an GS (Geräteschnittstelle) um.
- **Objekte:** Dateien, Blöcke ...
- **Komponenten:** Externspeicherverwaltung (abstrahiert von Gerätecharakteristika/Hardware), Speicherzuordnung
- **Operationen DS:**
  - Lese Block k
  - Schreibe Block k
  - ...



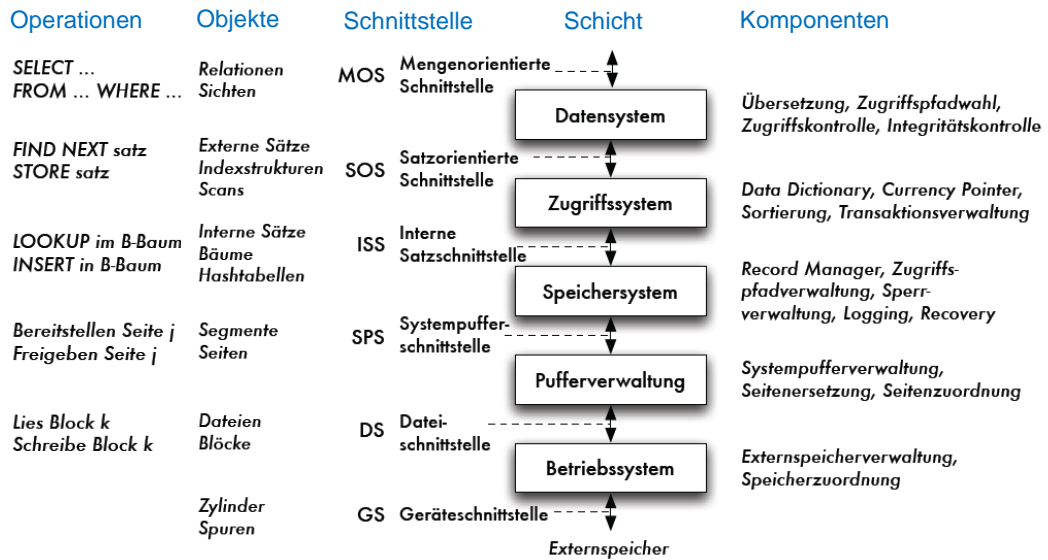


- Treiberprogramme zum Holen und Schreiben von Blöcken
- Zuordnung von Blöcken zu Seiten
- Fehlerkorrektur
- Caching-Mechanismen, die bereits gelesene Daten in einem Speicher halten und verwalten
- Operationen des Dateisystems von Betriebssystemen (Datenbanksysteme nutzen nur wenige physische Dateien. Beispiele: SQL Server, Oracle)



Datenbanksysteme nutzen häufig nur wenige physische Dateien. Der SQL-Server verwendet zur Speicherung der Daten, per Default eine einzige Datei für die Daten (plus eine Datei für die Protokollierung der Änderungen an den Daten).

# 5-Schichten-Architektur: Überblick

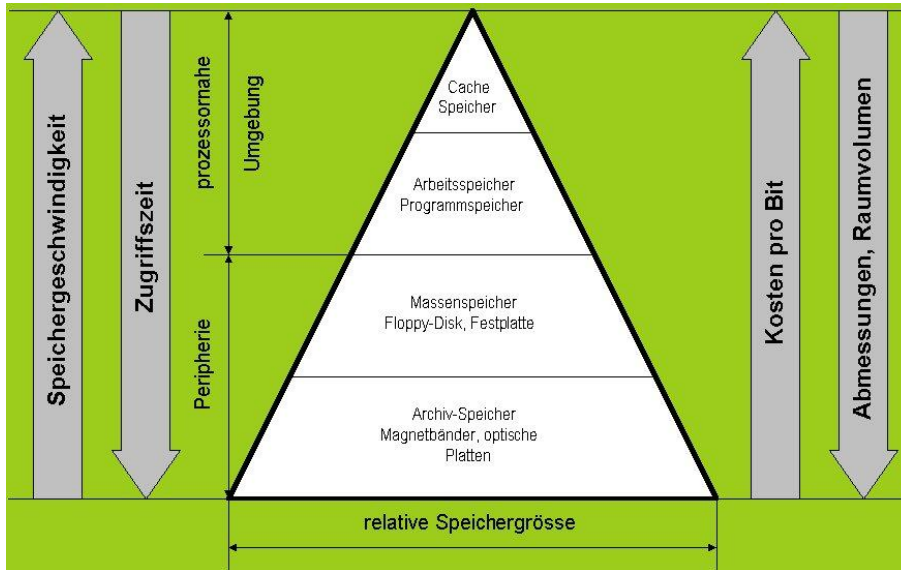


- Auf der untersten Schicht (Betriebssystem) des RDBMS geht es um das Sicherstellen der Persistenz (= dauerhafte Speicherung der Daten)
- Es gab und gibt eine unüberschaubare Vielzahl an unterschiedlichen *Speichermedien* mit ganz unterschiedlichen Eigenschaften
- Beispiele von Speichermedien: Harddisk, SSD, RAM, Tape, DVD, ...
- Eine grobe Klassifizierung hilft, Übersicht zu gewinnen:  
→ Speicherhierarchie

Dank steigenden Mengen verfügbaren Primärspeichers, kann in Spezialfällen auch die gesamte Datenbank in den Puffer geladen werden. Man spricht dann von In-Memory Datenbank, oder Main Memory Database. Diese sind wesentlich schneller, benötigen aber spezielle Mechanismen zur Sicherstellung der Persistenz.

## Speicherhierarchie:

- Primärspeicher (z.B. RAM / Cache)
  - Sehr schnell, (sehr) teuer
  - Volatil (Inhalt nur zur Laufzeit des Systems vorhanden)
  
- Sekundärspeicher (z.B. HD, SSD)
  - Faktor  $10^5$ -  $10^6$  langsamer als Primärspeicher! (SSD immer noch  $10^3$  langsamer)
  - Günstig - Billig
  - Persistent (Inhalt über die Laufzeit des Systems hinaus vorhanden)
  
- Tertiärspeicher (z.B. optische Speicher, Band)
  - Langsam, billig, hohe Kapazität
  - Wechselmedium, nicht permanent direkt zugreifbar
  - Persistent (Inhalt über die Laufzeit des Systems hinaus vorhanden)



- Verwendung von *Tertiärspeicher*medien im Zusammenhang mit Datenbanken:

- Speichern der aktuellen Datenbankinhalte für den laufenden Betrieb?  
→ Sekundärspeicher!



- Datensicherung?



- Archivierung?



Die Datensicherung wird verwendet um den aktuellen Zustand jederzeit wieder herstellen zu können.

Die Archivkopie wird verwendet, um 'beliebige' alte Zustände wieder herstellen zu können.

- Das nächste Mal: Verwaltung des Hintergrundspeichers
- Lesen: Lehrbuch Kapitel 3 (37 Seiten)