

# Physische Datenorganisation

## Definition „Physische Datenunabhängigkeit“:

Änderungen an der *physischen* Speicher- oder der Zugriffsstruktur (beispielsweise durch das Anlegen einer Indexstruktur) haben keine Auswirkungen auf die *logische* Struktur der Datenbasis, also auf das Datenbankschema.<sup>12</sup>

physischen

logische

## ANSI-SPARC-Architektur (auch Drei-Schema-Architektur)<sup>345</sup>

Trennung des Datenbankschemas in 3 verschiedene Beschreibungsebenen:

**Externe Ebene / (Benutzer)sichten (Views)** Die externe Ebene stellt den Benutzern und Anwendungen individuelle Benutzersichten bereit, wie beispielsweise *Formulare, Masken-Layouts, Schnittstellen*.

Formulare

Masken-Layouts

Schnittstellen

**Konzeptionelle / logische Ebene** Die konzeptionelle Ebene beschreibt, *welche Daten* in der Datenbank *gespeichert* sind, sowie *deren Beziehungen*. Ziel des Datenbankdesigns ist eine vollständige und redundanzfreie *Darstellung aller zu speichernden Informationen*. Hier findet die Normalisierung des relationalen Datenbankschemas statt.

welche Daten

gespeichert

deren Beziehungen.

Darstellung aller zu speichernden Informationen

**Interne / physische Ebene** Die interne Ebene stellt die physische Sicht der Datenbank im Computer dar. Sie beschreibt, *wie und wo die Daten in der Datenbank gespeichert werden*. Oberstes Designziel ist ein effizienter Zugriff auf die gespeicherten Informationen, der meist durch bewusst in Kauf genommene Redundanz (z. B. Index) erreicht wird.

wie und wo die Daten in der Datenbank gespeichert werden

## Vorteile des Modells:

**physischen Datenunabhängigkeit** Da die interne von der konzeptionellen und externen Ebene getrennt ist, wirken sich physische Änderungen, z. B. des Speichermediums oder des Datenbankprodukts, nicht auf die konzeptionelle oder externe Ebene aus.

**logischen Datenunabhängigkeit** Da die konzeptionelle und die externe Ebene getrennt sind, haben Änderungen an der Datenbankstruktur (konzeptionelle Ebene) keine Auswirkungen auf die externe Ebene, also die Masken-Layouts, Listen und Schnittstellen.

Allgemein: höhere Robustheit gegenüber Änderungen

<sup>1</sup>Qualifizierungsmaßnahme Informatik - Datenbanksysteme 3, Seite 20.

<sup>2</sup>Kemper und Eickler, Datenbanksysteme, Kapitel 1.3 Datenunabhängigkeit Seite 24.

<sup>5</sup>Qualifizierungsmaßnahme Informatik - Datenbanksysteme 3, Seite 20.

## Physische Datenorganisation<sup>6</sup>

- Daten werden in Form von *Sätzen* auf der Festplatte abgelegt, um auf Sätze zugreifen zu können, verfügt jeder Satz über eine *eindeutige, unveränderliche Satzadresse*
- TID = Tupel Identifier: dient zur Adressierung von Sätzen in einem Segment und besteht aus zwei Komponenten
  - Seitennummer (Seiten bzw. Blöcke sind größere Speichereinheiten auf der Platte)
  - Relative Indexposition innerhalb der Seite
- Satzverschiebung innerhalb einer Seite bleibt ohne Auswirkungen auf TID, wird ein Satz auf eine andere Seite migriert, wird eine „Stellvertreter-TID“ zum Verweis auf den neuen Speicherort verwendet. Die eigentliche TID-Adresse bleibt stabil<sup>7</sup>

## Literatur

- [1] Alfons Kemper und André Eickler. *Datenbanksysteme. eine Einführung*. 2013.
- [2] *Qualifizierungsmaßnahme Informatik - Datenbanksysteme 3. SQL Vertiefung, Integrität, Physische Datenorganisation*. [https://www.studon.fau.de/file2480904\\_download.html](https://www.studon.fau.de/file2480904_download.html).

---

<sup>6</sup>Qualifizierungsmaßnahme Informatik - Datenbanksysteme 3, Seite 21.

<sup>7</sup>Kemper und Eickler, *Datenbanksysteme*, Seite 219.