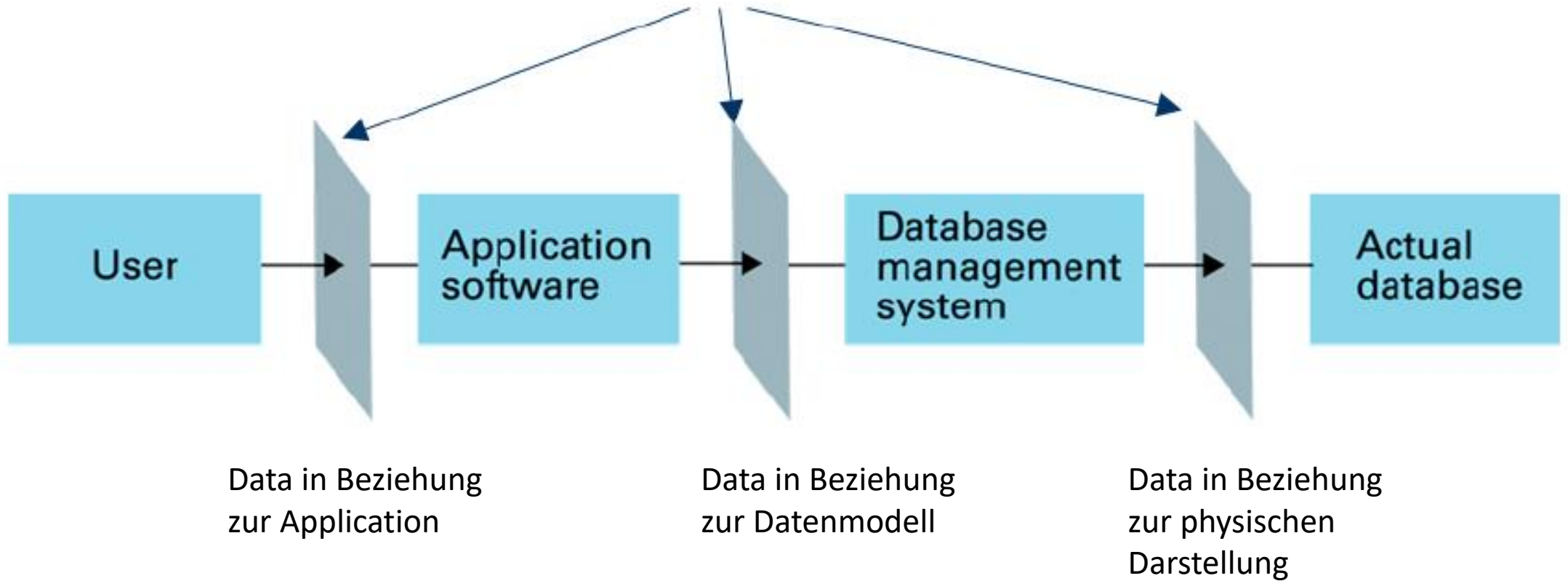


Abstraktionsschichten. Das Relationale Datenmodell

Verschiedene Abstraktionsebene

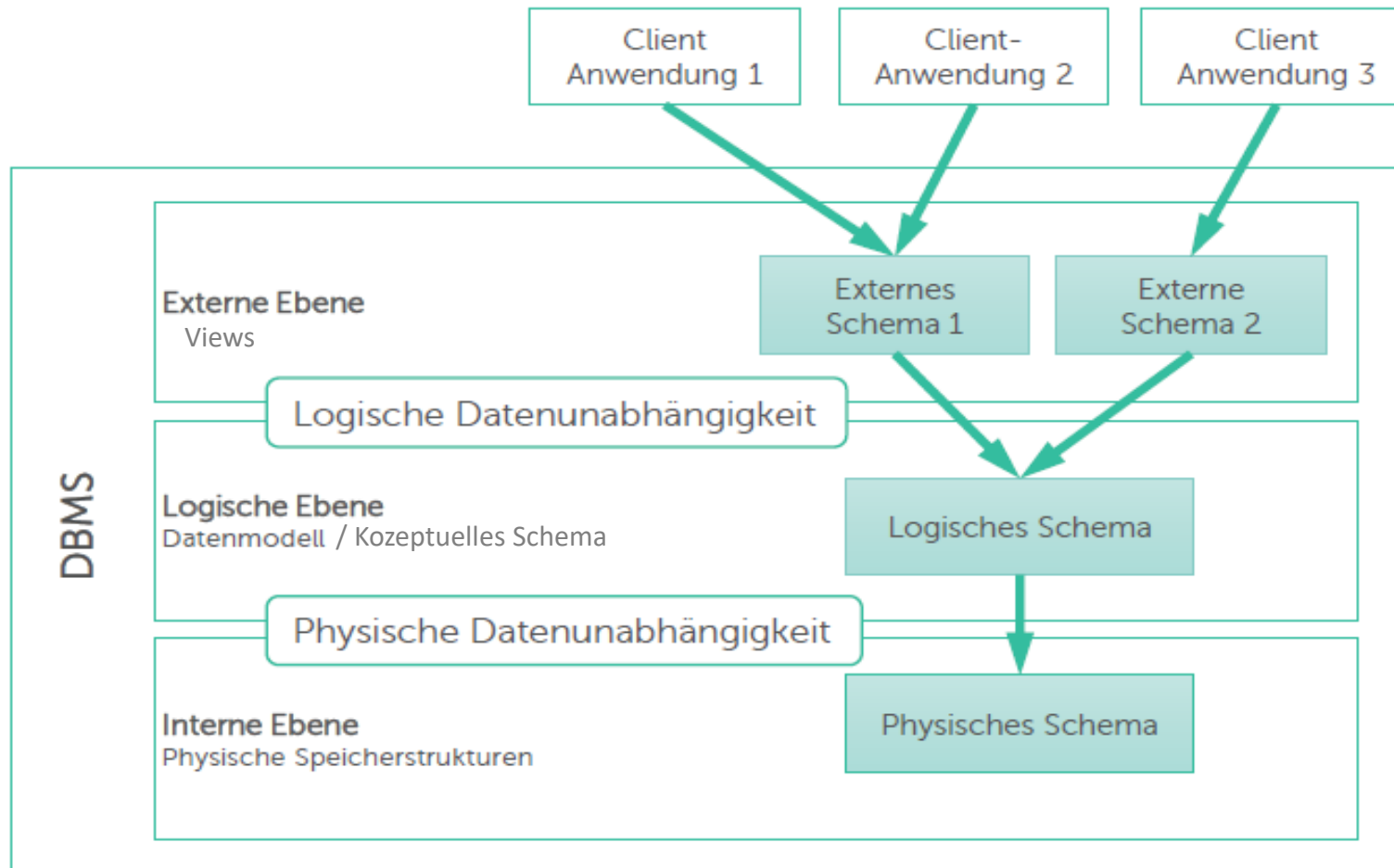


Datenunabhängigkeit

- Ziel: die Datendefinition in einer Schicht zu ändern, ohne dabei die Definition der Daten in der darüber liegenden Schicht zu beeinflussen
- Logische Datenstrukturen unabhängig von physischen Datenstrukturen
- Robustheit der Anwendungen gegenüber Änderungen

Abstraktionsebenen eines Datenbanksystems

- ANSI-SPARC-Architektur



3-Schichten nach ANSI-SPARC

- Externe Ebene:
 - Anwendungs-spezifische Sichten/Views (Ausschnitte aus dem Datenmodell)
 - Ebene der Anwendungen und Verwendung der Daten
 - Beschreibt wie der Benutzer die Daten sieht
- Logische Ebene:
 - Definition der logischen Datenstrukturen: die Beziehungen zwischen den Daten unabhängig von der physischen Repräsentation
- Interne/Physische Ebene:
 - Definition des physischen Schemas (wie die Daten gespeichert werden)
 - Beschreibt die Dateien und Indexstrukturen die benutzt werden
 - Es geht um Leistungsfähigkeit der Datenbank Anwendungen

Beispiel: Universität Datenbank

- Logische Ebene/Konzeptuelles Schema:
 - **Studenten**(sid:string, name:string, email:string, age:integer, gruppe:integer)
 - **Vorlesung**(vid:string, vname:string, ects:integer)
 - **Klausur**(sid:string, vid:string, note:integer)
- Physische Ebene:
 - Relationen werden als ungeordnete Dateien gespeichert
 - Indexstruktur auf die erste Spalte (sid) der Tabelle **Studenten**
- Externe Ebene/View:
 - Vorlesungsinfo(vid:string, enrollment:integer)

Physische Datenunabhängigkeit

- Änderungen an der Art der Datenspeicherung und den Zugriffstechniken haben keinen Einfluss auf Anwendungsprogramme
- Programme sind von interner Datenorganisation unabhängig
- Physische Datenunabhängigkeit wird durch relationale DBMS weitestgehend hergestellt
- Beispiele:
 - Verschiebung von Datenbank auf anderes Storage-System
 - Partitionierung
 - Indexstrukturen für performante Zugriffe

Logische Datenunabhängigkeit

- Ermöglicht durch externe Ebene
- Änderungen am logischen Schema haben nur geringe Auswirkungen auf die Anwendung
- Ist nur begrenzt möglich, denn Anwendungen basieren sich auf dem logischen Schema
- Sichtenkonzept (Views) ermöglicht Verbergen von Details des logischen Datenmodells
- Beispiel: nach Umbenennung einer Tabelle oder eines Attributes kann die alte Datenstruktur virtuell über eine Sicht bereitgestellt werden

Queries (Abfragen) im DBMS

- Mögliche Fragen für die Universität Datenbank:
 - Welcher ist der Name des Studenten mit sid=2310?
 - Wie viele Studenten haben sich für die Vorlesung DB angemeldet?
 - Wie viele Studenten haben in der DB Klausur mehr als 9 gekriegt?
- Fragen, die sich auf Daten gespeichert im DBMS beziehen, sind Datenbank-Abfragen
- DBMS hat bestimmte Datenbanksprachen zum Abfragen und Manipulieren der Daten (Query Language)

Datenbanksprache

- Formale Sprache die folgende Komponenten hat:
 - Datenbeschreibungssprache / Data Definition Language (DDL)
 - Befehle zur Definition des Datenbankschemas (Anlegen, Ändern und Löschen von Datenstrukturen)
 - Constraint Definition Language (CDL) – beschreibt Integritätsregeln, die von den Instanzen der Datenbank erfüllt werden sollen
 - Storage Definition Language (SDL) – um den Layout der physischen Schema zu beeinflussen
 - Datenmanipulationssprache / Data Manipulation Language (DML)
 - Befehle zur Datenmanipulation (Abfragen, Einfügen, Ändern oder Löschen von Nutzdaten)
 - Procedural DML (wie) vs. Declarative DML (was)

Abfragesprachen für relationale Datenbanken

- SQL (Structured Query Language)
 - SELECT name FROM studenten WHERE age>20
- Relationale Algebra
 - $\pi_{\text{name}}(\sigma_{\text{age}>20}(\text{Studenten}))$
- Domänenkalkül (Domain Calculus)
 - $\{ \langle X \rangle \mid \exists V \exists Y \exists Z \exists T: \text{Studenten}(V, X, Y, Z, T) \wedge Z > 20 \}$
- Tupelkalkül (Tuple Calculus)
 - $\{ X \mid \exists Y : Y \in \text{Studenten} \wedge Y.\text{age} > 20 \wedge X.\text{name} = Y.\text{name} \}$

Relationale Datenbankabfragesprache/ Relational Query Language

- Vorteil des relationalen Modells:
 - unterstützt einfache Datenbankabfragesprachen
- Abfragen können intuitiv formuliert werden und der DBMS ist zuständig für die optimale Abfrage Bearbeitung
- Aber: manchmal müssen wir die Abfrage optimal formulieren, um die gewünschte Zeitkomplexität zu erreichen

Structured Query Language (SQL)

- Ist die Sprache die von den meisten relationalen Datenbanken unterstützt wird
- Wurde 1970 von IBM entwickelt
- SQL Standard:
 - SQL 86
 - SQL 89
 - SQL 92 (SQL 2)
 - SQL 2003
 - SQL 2008
 - SQL 2011
 - SQL 2016
 - SQL 2019

SQL Befehle

- Kategorien von SQL Befehle:
 - Datenbeschreibungssprache / Data Definition Language (DDL)
 - Anlegen, Ändern und Löschen von Relationen oder Views
 - Beschreiben von Integritätsregeln
 - Datenmanipulationssprache / Data Manipulation Language (DML)
 - Einfügen, Ändern oder Löschen von Daten in den Relationen (Tupeln)
 - Abfragen (Queries)
 - Datenüberwachungssprache / Data Control Language (DCL)
 - Befehle für Rechteverwaltung (auf Tabellen und Views) und Transaktionskontrolle