Datenbanksysteme I

Wintersemester 2026/27

—

Übungsblatt 1

Benjamin Uhrich, Lucas Lange

Abteilung Datenbanken
Institut für Informatik
Universität Leipzig

Aufgabe 1: Grundbegriffe

Geben Sie für die nachfolgenden Begriffe kurze Definitionen an und erläutern Sie die Beziehungen zwischen ihnen.

- DB, DBVS/DBMS, DBS
- Mehrbenutzerbetrieb, Transaktion, log. Einbenutzerbetrieb, Synchronisation
- OLTP, OLAP, Data Warehouse

Aufgabe 1: Grundbegriffe

DB: Eine Datenbank ist eine Sammlung gespeicherter operationaler Daten, die von Anwendungssystemen benötigt werden.

DBVS/DBMS: Ein Datenbankverwaltungssystem (DBVS, Database Management System) ist ein standardisiertes Softwaresystem zur Definition, Verwaltung, Verarbeitung und Auswertung von Datenbankdaten.

DBS: Ein Datenbanksystem besteht aus einem DBMS und einer DB.

Aufgabe 1: Grundbegriffe

Mehrbenutzerbetrieb: Mehrbenutzerbetrieb bedeutet die gleichzeitige Nutzung der operationalen Daten durch sehr viele Benutzer

Transaktion: Eine Transaktion ist eine logische Verarbeitungseinheit, die einen Vorgang der realen Welt durch eine Folge von Operationen in der Datenbank nachbildet. DBS gewährleistet die ACID-Eigenschaften für jede Transaktion. BOT = Begin of Transaction EOT = End of Transaction

log. Einbenutzerbetrieb: Beim logischen Einbenutzerbetrieb hat jede der parallel ablaufenden Transaktionen den Eindruck, als liefe sie alleine ab, d.h. logisch bilden alle Transaktionen eine serielle Ablauffolge.

Synchronisation: Als Synchronisation werden die Maßnahmen bezeichnet, die den logischen Einbenutzerbetrieb, d.h. die Vermeidung von Mehrbenutzeranomalien, sicherstellen.

Aufgabe 1: Grundbegriffe

OLTP: **O**nl**i**ne **T**ransaction **P**rocessing ist die dominierende Einsatzform von DBS, bei der vorgeplante Anwendungsprogramme (z.B. Produktbestellungen, Überweisungen) auf Datenbanken zugreifen. Die kurzen jedoch häufig ablaufenden Transaktionen umfassen wenige Daten.

OLAP: **O**nl**i**ne **A**nalytical **P**rocessing bezeichnet die umfassende Auswertung/Analyse großer Datenbestände (meist über Data Warehouses), um Geschäftsentscheidungen zu unterstützen, z B. Vertriebskontrolle, Preisoptimierung, ...

Data Warehouse: Ein Data Warehouse ist eine Datenbank, in der Datenbestände unterschiedlicher Quellen (meist innerhalb eines Unternehmens) für Analysen konsolidiert und integriert werden.

Aufgabe 2: DBVS vs. Dateisystem

Eine Versandfirma verwendet ein selbst entwickeltes Lagerverwaltungssystem, welches die relevanten Daten in mehreren Dateien (Artikelliste, Lagerbestand) verwaltet. Welche Nachteile ergeben sich hinsichtlich folgender Situationen:

- Die Ansicht der Artikelliste soll nach dem Lagerbestand sortiert werden.
- Durch eine Vergrößerung der Firma müssen jetzt mehrere Personen den Wareneingang einpflegen (vorher nur eine Person).
- Für einen neuen Online-Auftritt mit Bestellmöglichkeit kommt ein separates Programm zum Einsatz, welches aus Sicherheits- und Performanzgründen auf einem separaten Rechner mit einer Kopie der Artikelliste arbeitet.
- Für den Online-Auftritt muss die Artikelliste um einige Attribute ergänzt und die Artikelnummerierung erweitert werden.
- Durch eine Stromunterbrechung fällt der Server aus.
- Durch Firmenfusion müssen nun erheblich mehr Artikel gespeichert werden.

Aufgabe 2: DBVS vs. Dateisystem

- Die Ansicht der Artikelliste soll nach dem Lagerbestand sortiert werden.
  - Neue Sortieroperation implementieren
  - Verbund eigenständiger Daten (referenzielle Integrität, Redundanz)
- Durch eine Vergrößerung der Firma müssen jetzt mehrere Personen den Wareneingang einpflegen (vorher nur eine Person).
  - Sicherstellen der Mehrbenutzerfähigkeit bei parallelen Schreibzugriffen (Gesamtsperrung vs. Nutzbarkeit)
- Für einen neuen Online-Auftritt mit Bestellmöglichkeit kommt ein separates Programm zum Einsatz, welches aus Sicherheits- und Performanzgründen auf einem separaten Rechner mit einer Kopie der Artikelliste arbeitet.
  - Redundanz (Synchronisieren von Änderungen)
- Für den Online-Auftritt muss die Artikelliste um einige Attribute ergänzt und die Artikelnummerierung erweitert werden.
  - Änderungen in allen Programmen, die auf die Daten zugreifen, erforderlich (auch wenn neue Attribute nicht benötigt)
  - Neues Artikelnummernformat muss in allen Dateien eingeführt werden, die auf die Artikelliste verweisen
- Durch eine Stromunterbrechung fällt der Server aus.
  - Unvollständige Schreiboperationen können zu zerstörten Daten führen
  - Wiederherstellung konsistenter Zustand
  - Wiederholung aller (vor dem Ausfall) bestätigter Verwaltungsvorgänge
- Durch Firmenfusion müssen nun erheblich mehr Artikel gespeichert werden.
  - Es können sehr große Dateien entstehen, welche nicht mit herkömmlichen Programmen verwaltet werden können.

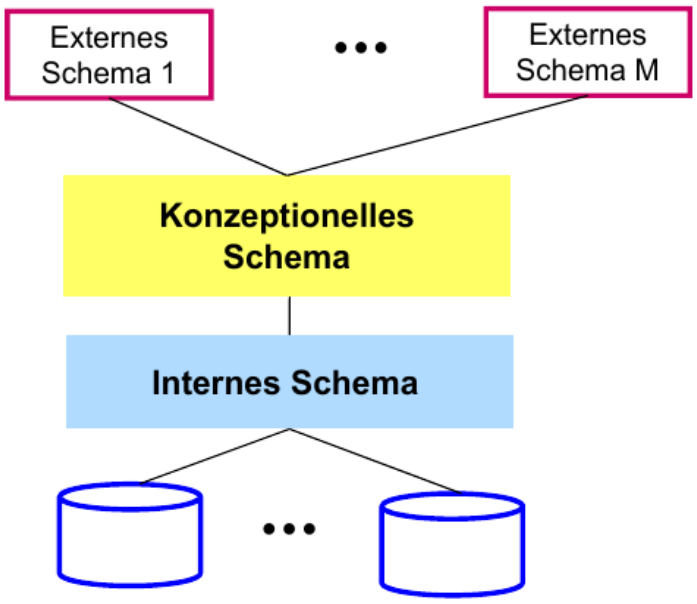
Aufgabe 3: Datenunabhängigkeit

- Was versteht man unter dem Begriff Datenunabhängigkeit?
  - Maß für die Isolation zwischen Anwendungsprog. (AWP) und Daten (umgekehrt proportional zu dem Grad der Annahmen, den AWP über die Datenstrukturen machen müssen).

- Welche Typen von Datenunabhängigkeit gibt es?
  - Physische DU: (Minimalziel): Unabhängigkeit gegenüber physischen Strukturen (wie Geräteeigenschaften, Speicherungsstrukturen, Indexstrukturen, ...)
  - Logische DU: Unabhängigkeit gegenüber logischer Strukturierung der Daten

- Warum ist Datenunabhängigkeit von Vorteil?
  - Änderungen im Informationsbedarf sowie bei Leistungsanforderungen erfordern Anpassungen bei den Speicherungsstrukturen und Zugriffsstrategien
    - Möglichst starke Isolation der Anwendungsprogramme von den Daten wichtig.
    - Anderenfalls extremer Wartungsaufwand für die AWP.
  - Verschiedene Anwendungen brauchen verschiedene Sichten auf dieselben Daten

Aufgabe 4: 3-Ebenen-Architektur



- **Internes Schema** Legt physische Struktur der DB fest
- **Konzeptionelles Schema**
  - Bietet logische Sicht auf die Struktur der Daten
  - Abstrahiert von internen Struktur der Daten
  - Physische Datenunabhängigkeit
- **Externe Schemata**
  - Definiert Nutzersichten auf die Struktur der Daten für Nutzer und Anwendungsprogramme
  - Zugriffsschutz
  - Reduzierung der Komplexität
  - Abstrahiert vom konzeptionellen Schema
  - **Teilweise** logische Datenunabhängigkeit

Aufgabe 4: 3-Ebenen-Architektur

Abteilungen	Abteilungsnummer, Bezeichnung und LeiterIn.
Projekte	Projektnummer, Titel, Abteilung
Angestellte	Personalnummer, Name, Adresse, Gehalt, Abteilung, Projekt

(a) Abteilungsleiter sollen zusätzlich zu den Daten der Angestellten ihrer Abteilung auch die Daten der Angestellten anderer Abteilungen, die an einem Projekt ihrer Abteilung mitarbeiten, lesen können.

- Externes Schema betroffen.

(b) Der Lesezugriff auf Projekttitel ist zu langsam und wird durch Anlegen eines Zugriffspfades (Index) beschleunigt.

- Internes Schema betroffen.

(c) Durch Reorganisation der Gehaltsabrechnung werden Angestelltendaten statt in alph. Reihenfolge sortiert nach Gehaltsgruppen benötigt.

- Externes Schema betroffen.

(d) Ein Anwendungsprogramm zur Anzeige der Liste aller Projekte mit ihren zugehörigen Angestellten soll hinsichtlich des Datenschutzes keine Adressdaten der Angestellten darstellen.

- Externes Schema betroffen.

(e) Eine Abteilung wird aufgelöst und alle ehemaligen Angestellten werden anderen Abteilungen zugewiesen.

- Kein Schema betroffen.

Aufgabe 5: Transaktionen

(a) Erläutern Sie die wesentlichen Transaktionseigenschaften

- **A**tomicity
  - Alles-oder-nichts-Eigenschaft = ...
  - Sicherung durch UNDO-Recovery
- **C**onsistency
  - Transaktion überführt DB von konsistenten Zustand in konsistenten Zustand (Konsistenzerhaltung)
  - Sicherung durch Integritätsbedingungen
- **I**solation
  - Logischer Einbenutzerbetrieb
  - Sicherung z.B. durch Sperren (lesen, schreiben) – Locking
- **D**urability
  - Persistenz der Änderungen abgeschlossener Transaktionen
  - Sicherung durch REDO-Recovery

Aufgabe 5: Transaktionen

(b) Welche Aufgaben übernimmt die Datenbank-Recovery?

- Gewährleistung von Atomarität (A) und Persistenz (D)
- **UNDO**-Recovery
  - Einhaltung der Atomarität bei abnormalem Ende (z.B. Systemausfall): alle Operationen in nicht per COMMIT abgeschlossenen Transaktionen müssen zurückgesetzt werden (konsistenter Zustand vor Beginn der Transaktion)
- **REDO**-Recovery
  - Einhaltung der Persistenz der Änderungen bei abnormalem Ende: Wiederholung aller ausstehenden Operationen von durch COMMIT abgeschlossenen Transaktionen
- Voraussetzung: Führen von Änderungsprotokollen - Logging

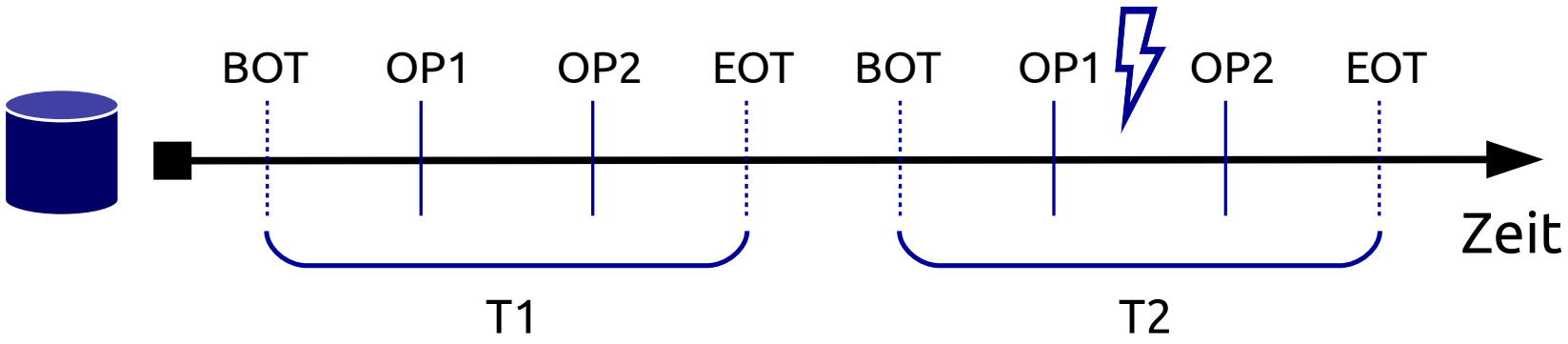
Aufgabe 5: Transaktionen

(c) Was sind Integritätsbedingungen (IB)? Sollten IB durch das DBVS oder durch Anwendungsprogramme realisiert werden?

- Annahmen (Regeln) bzgl. DB-Inhalt, welche erfüllt werden müssen
- Beschreiben 'sinnvolle' und 'zulässige' Änderungen der DB – Konsistenz-Begriff
- Untersch. Reichweite; Zustandsbedingungen vs. Übergangsbedingungen
  - Attributbedingungen: Geb.-Jahr numerisch, 4-stellig
  - Satzbedingung: Geb.Jahr $<$ Einstellungsjahr
  - Übergang von FAM-Stand 'ledig' nach 'geschieden' unzulässig
- Überwachung der IB durch DBS
  - Größere Sicherheit
  - Vereinfachte Anwendungserstellung
  - Leichtere Änderbarkeit

Aufgabe 5: Transaktionen - Big Picture

- **Szenario:** Zahlungsverkehr (Überweisung, Lastschrift)
  - Transaktion umfasst zwei Operationen:
    - OP1: Verringere Kontostand des Auftraggebers um x €
    - OP2: Erhöhe Kontostand des Empfängers um x €
  - Integritätsbedingung: Debit-Konto (Kontostand nicht negativ)



- T2 wird durch Fehler unterbrochen: IB verletzt, Stromausfall
- UNDO T2 (Atomicity), ggfs. REDO T1 (Durability)

Aufgabe 5: Transaktionen

(d) Beurteilen Sie unter Berücksichtigung der Transaktionseigenschaften die folgenden Aussagen bzgl. ihrer Richtigkeit. Begründen Sie Ihre Aussage.

- Aufgrund eines Stromausfalls wird eine Transaktion unterbrochen. Die Änderungen, welche die Transaktion vor dem Abbruch durchgeführt hat, gehen nicht verloren.
  - Falsch – wegen **A** wird dem AWP entweder der Zustand vor BOT oder nach EOT zugesichert.
- Das Datenbanksystem gewährleistet, dass in einer Transaktion nur die aktuellste Version der Daten, die von einer anderen Transaktion gerade modifiziert werden, gelesen werden kann.
  - Falsch – wegen **I** sind Daten, die von anderen Transaktionen modifiziert werden, nicht zu lesen.
- Eine Online-Auktionsbörse verwende eine Datenbank zur Verwaltung der abgegebenen Gebote. Mit Mitteln des DBVS kann das Abgeben eines Gebotes, welches das bisher höchste Gebot unterschreitet, verhindert werden.
  - Richtig – durch Definition von Integritätsbedingungen, welche durch **C** gewahrt werden.
- Nach dem Absturz des Datenbanksystems werden die Änderungen aller zuvor erfolgreich abgeschlossenen Transaktionen zurückgesetzt, sodass die DB einen konsistenten Zustand erreicht.
  - Falsch – wegen **D** dürfen erfolgreich abgeschlossene Transaktionen nicht rückgängig gemacht werden.