

# GDB Aufgabenblatt 1

---

*Abgegeben von Timon Back, Fabian Behrendt, Nicolai Stäger*

## Aufgabe 1: Informationssysteme

### 1.A Charakterisierung

*Erläutern Sie den Begriff Informationssystem und nennen Sie in diesem Zusammenhang drei relevante Aufgaben eines rechnergestützten Informationssystems.*

Ein Informationssystem besteht aus Menschen und Maschinen, die Informationen erzeugen und/oder benutzen und die durch Kommunikationsbeziehungen miteinander verbunden sind.

Ein rechnergestütztes IS hat die Aufgabe der Erfassung, Speicherung und Transformierung von Daten, was durch den Einsatz von EDV teilweise automatisiert ist.

### 1.B Datenunabhängigkeit

*Definieren Sie kurz den Begriff Datenunabhängigkeit und unterscheiden Sie dabei die logische von der physischen Datenunabhängigkeit.*

Datenunabhängigkeit beschreibt den Grad, zu welchem ein Benutzer oder ein Anwendungsprogramm auf die Daten eines Datenbanksystems zugreifen kann, ohne Details der systemtechnischen Realisierung der Datenspeicherung und des Datenzugriffs zu kennen. Datenunabhängigkeit ermöglicht somit eine starke Isolation von Applikationen, sodass Daten unabhängig von ihrer Verwendung und ihrer Struktur gespeichert werden können.

Die physische Datenunabhängigkeit beschreibt dabei ein Datenbanksystem, bei dem Änderungen an der physischen Speicher- oder Zugriffsstruktur keine Auswirkungen auf die logische Struktur der Datenbasis haben. Logische Datenunabhängigkeit bedeutet, dass Anwendungen gegen Änderungen am Datenbankschema immun sind.

### 1.C Beispiele

Nennen Sie drei Anwendungsbeispiele für Informationssysteme und beschreiben Sie die jeweils charakteristischen Vorgänge. Vermeiden Sie die Wiederholung von Beispielen aus der Vorlesung.

- Geldautomat: Geld abheben, Kontoauszug ausdrucken
- Fahrkartenautomat: Fahrkarte kaufen, Zugverbindungen anzeigen

## Aufgabe 2: Miniwelt

*Für ein Tippspiel der kommenden Fußball-Weltmeisterschaft soll eine browserbasierte Anwendung implementiert werden. Das Tippspiel läuft dabei wie folgt ab: Jeder Mitspieler soll sich an der Anwendung anmelden und eine Tippspielgemeinschaft erstellen können. Der jeweilige Gründer verwaltet dabei die Tippspielgemeinschaft indem er für diese Wettbewerbe anlegt, Begegnungen zu einem Wettbewerb hinzufügt oder Ergebnisse einträgt. Der Tippspielgründer hat außerdem die Möglichkeit, weitere Mitspieler zu dieser Gemeinschaft hinzuzufügen oder auch zu entfernen. Diese*

hinzugefügten Mitspieler können dann Tipps auf die angelegten Begegnungen abgeben. Nach dem Beenden einer Begegnung trägt der Verwalter das Ergebnis ein und die Mitspieler erhalten für ihre Tipps entsprechende Punkte. Die Mitspieler können sich in der Anwendung jeweils über ihren aktuellen Punktestand innerhalb einer Tippgemeinschaft informieren oder auch die Ergebnisse zu den Begegnungen einsehen.

## 2.A

Leiten Sie aus der beschriebenen Miniwelt die für die Realisierung des Tippspiels relevanten Elemente (Objekttypen) und Vorgänge ab.

- Spieler
  - Anmeldung an der Anwendung
  - Tippspielgemeinschaft erstellen
  - Wettbewerbe anlegen
  - Begegnungen zu Wettbewerben hinzufügen
  - Ergebnisse eintragen
  - Mitspieler hinzufügen/entfernen zu Tippspielgemeinschaften
  - Ergebnisse und Punktestand aus einer Begegnung in einer Tippspielgemeinschaft ansehen
- Tippspielgemeinschaft
  - Wird von einem Gründer verwaltet
  - hat mehrere Wettbewerbe
  - hat mehrere Mitspieler
  - jeder Mitspieler in der Gemeinschaft hat einen Punktestand
- Wettbewerb
  - Gehört zu einer Tippspielgemeinschaft
  - hat mehrere Begegnungen
- Begegnung
  - Gehört zu einem Wettbewerb
  - hat mehrere Tipps
  - hat ein Endergebnis
- Tipp
  - Mitspieler gibt Tipp zu einer Begegnung ab

## 2.B

Welche Anforderungen an die Anwendung ergeben sich aus der beschriebenen Miniwelt?

Diskutieren Sie die in der Vorlesung genannten allgemeinen Anforderungen an Datenbanksysteme anhand des hier beschriebenen Beispiels.

- Kontrolle über die operationalen Daten
  - Die Daten aus der Miniwelt müssen für viele Nutzer zugänglich sein
  - Zwischen den Daten entstehen Zusammenhänge die ausgewertet werden müssen (z.B. Ein Spieler gibt einen Tipp zu einer Begegnung innerhalb einer Tippgemeinschaft ab)
  - Daten sollten nicht redundant gespeichert werden, um Synchronisierungsprobleme zu vermeiden
- Leichte Handbarkeit der Daten

- Datensätze werden anhand logischer Aspekte beschrieben (z.B. Ein Spieler ist Mitspieler bei beliebig vielen Wettbewerben)
- Daten sollten einfach genutzt werden können, auch ohne Bezug auf systematische Realisierung
- Anwendung hat eine eigene logische Sicht auf die Datenbank, die auf ihre Bedürfnisse zugeschnitten ist
- Die Datenbank sollte geltenden Standards entsprechen, dass die Anwendung in Zukunft portierbar bleibt und ein Datenaustausch erleichtert wird
- Kontrolle der Datenintegrität
  - Zugriff auf Daten sollte automatisiert kontrolliert werden, dass keine Konflikte beim gleichzeitigen Bearbeiten von gleichen Datensätzen entstehen
  - Verschiedene Spieler haben in verschiedenen Wettbewerben unterschiedliche Rechte (nur der Gründer kann z.B. neue Spieler einladen)
  - Daten sollten durch bestimmte Regeln beschrieben werden (z.B. eine Begegnung gehört immer zu einem Wettbewerb)
  - Daten sollten bei einer Transaktion nicht verloren gehen (wenn z.B. die Internetverbindung abbricht)
  - Daten sollten regelmäßig durch Kopien gesichert werden
- Leistung und Skalierbarkeit
  - Die Anwendung sollte einfach an die Benutzerzahlen angepasst werden können (kurze Antwortzeiten, hoher Durchsatz)
- Hoher Grad an Daten-Unabhängigkeit
  - Anwendungsprogramm sollte von den Daten isoliert werden, damit beides unabhängig voneinander gewartet, modifiziert und skaliert werden kann

### Aufgabe 3: Transaktionen

Im Folgenden ist eine Überweisung in Pseudocode von einem Konto mit der ID 5 auf das Konto mit der ID 7 skizziert...

Zum Zeitpunkt A bzw. zum Zeitpunkt B kommt es zu einem Stromausfall. Welche Folgen hat der jeweils resultierende Systemabsturz? Achten sie darauf, dass geänderte Daten nicht notwendigerweise sofort auf die Platte geschrieben werden. Wie können problematische Folgen verhindert werden, wenn der Vorgang in einem Datenbanksystem abgewickelt wird?

Zeitpunkt	Daten sofort schreiben?	Folgen
A	Ja	Die Transaktion wurde nicht abgeschlossen, Geld wurde von Konto 5 abgebucht aber nicht auf Konto 7 eingebucht. Inkonsistent
A	Nein	Die Transaktion wurde nicht abgeschlossen, ist aber unterbrochen worden und dadurch verloren gegangen. Daten sind konsistent, da unverändert.
B	Ja	Die Transaktion wurde nicht abgeschlossen, aber das Geld wurde erfolgreich verbucht. Daten sind konsistent. Aber der Nutzer weiß eventuell nicht, dass die Daten bereits erfolgreich verbucht wurden.
B	Nein	Die Transaktion wurde nicht abgeschlossen, ist aber unterbrochen worden und dadurch verloren gegangen. Daten sind konsistent, da unverändert.

Durch die Verwendung eines DBS können die oben beschriebenen Folgen verhindert werden

- Eine Transaktion wird entweder ganz oder gar nicht ausgeführt (atomar)
- Eine unterbrochene Transaktion kann nach dem Systemstart erneut ausgeführt werden

## Aufgabe 4: Warm-Up MySQL

### 4.A

Es wurde eine neue Tabelle innerhalb der Datenbank erzeugt, die die Felder id, name und password hat. Dabei ist das Feld id der Primärschlüssel der Tabelle.

Durch den Insert-Befehl wurde außerdem ein erster Datensatz in der Tabelle eingefügt mit den Werten id:1, name: „gdbNutzer“ und password: „geheim“.

### 4.B

Nach dem Select-Befehl wurde mir eine Liste von den Feldern id, name und password ausgegeben, in dem der Eintrag mit dem Datensatz des „gdbNutzer“ zu sehen war.

Danach wurde mit dem Drop-Befehl die Tabelle wieder gelöscht.

### 4.C

Die hier angegebene MySQL Architekturübersicht ist kaum zu vergleichen mit der ANSI-SPARC Architektur. Im ANSI-SPARC wird die Architektur auf drei Schichten aufgeteilt. Die MySQL Architektur hingegen hat eigentlich nur zwei Schichten und ist deutlich „vermischter“. Die SPARC Schicht „Externe Ebene“ wird in der MySQL Architektur wohl durch die Connectors, Connection Pool und eventuell SQL Interface und Parser umgesetzt. Die „Konzeptionelle Ebene“ wird in dem MySQL Modell vermisst und nicht wirklich aufgeführt. Der Rest entspricht der „Internen Ebene“.