

Einführung in Datenbanken

Übung 1: Einführung

Prof. Dr. Stefan Brass
PD Dr. Alexander Hinneburg

Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2020/21

<http://www.informatik.uni-halle.de/~brass/db20/>

Inhalt

- 1 Organisatorisches
- 2 Motivation
- 3 Wiederholung
- 4 DB-Entwurf
- 5 Einführung zum Adminer
- 6 Präsenzaufgaben

Haben Sie organisatorische Fragen?

- Webseite (Folien, Links, aktuelle Ankündigungen):
<https://users.informatik.uni-halle.de/~brass/db20/index.html>
- Aufteilung von „Datenbanken I“ (10 LP) in
 - „Einführung in Datenbanken“ (5 LP, Winter)
 - „Datenbank-Programmierung“ (5 LP, Sommer)
- Modul-Anmeldung nicht vergessen!
Falls nicht möglich, bitte noch eine Woche warten, ob das Prüfungsamt eine Lösung findet. Dann notfalls mit Papier-Formular.
- Präsenz-Teilnahme (Montag, 10³⁰–12⁰⁰, Raum 328)
Weiter mit Umfrage in StudIP. Es gibt nur 33 Plätze.
- Vorlesungs-Aufzeichnung
Bei Problemen bitte melden. Erfahrungen aus anderen Vorlesungen?

Erreichbarkeit bei Fragen

- Hier in der Übung!
- Forum in **StudIP** (Eintrag der Vorlesung)
- Prof. Dr. Stefan Brass:
 - Sprechstunde: Montags, 12¹⁵–13⁰⁰
Raum 313, Tel. 0345/55-24740
 - Online-Sprechstunde: Montags, 16³⁰–16⁴⁵ (und nach Bedarf)
[\[https://mluconf.uni-halle.de/b/ste-p6j-fmd-i21\]](https://mluconf.uni-halle.de/b/ste-p6j-fmd-i21)
 - EMail: brass@informatik.uni-halle.de
- PD Dr. Alexander Hinneburg:
 - (nur während Übung)

Studienleistung und Klausur

- Studienleistung: 50% der Summe der Punkte von
 - Hausaufgaben (einzeln zu bearbeiten) [ca. 80%]
 - Präsenzaufgaben (hier, Gruppenarbeit) [ca. 20%]
- Falls Plagiate bei Hausaufgaben oder sehr still bei Übungen: Änderung (Verschärfung) der Regeln möglich.
Modulbeschreibung: „Aktive Mitarbeit in den Übungen inkl. Kurzvorträge über Hausaufgaben und Beantwortung von Fragen zum Umfeld der Aufgaben“.
- Klausur (E-Klausur in Präsenz??), aktuell geplante Termine:
 - Montag, 01.03.2021, 10:00-12:00
Zwei Wochen nach letzter Vorlesung.
 - Mittwoch, 07.04.2021, 10:00-12:00
Woche nach Ostern, letzte Woche vor Vorlesungsbeginn Sommersemester.

Vorkenntnisse

- Wer kennt schon SQL?

Woher? Schule? Auch Unteranfragen (z.B. NOT EXISTS)?

Vorkenntnisse erleichtern natürlich vieles. Es gibt aber auch das Risiko, dass man fälschlich annimmt, man wüsste schon alles.

- Wer kennt das Entity-Relationship-Modell (ER-Modell)?

Es gibt viele verschiedene Varianten.

- Wer kennt UML Klassendiagramme?

- Wer kennt Syntaxdiagramme (kontextfreie Grammatiken)?

- Wer kennt NICHT Java?

OOP wird vorausgesetzt, aber es geht mehr um Grundlagen wie formale Syntax, Datentypen, Vorstellung von Schleifen, etc. Für Fortsetzung im Sommer nötig!

- Wer kennt Aussagenlogik? \wedge ? Wozu ist $\neg(p \wedge q)$ äquivalent?

Inhalt

- 1 Organisatorisches
- 2 Motivation**
- 3 Wiederholung
- 4 DB-Entwurf
- 5 Einführung zum Adminer
- 6 Präsenzaufgaben

Motivation (1)

Aufgabe:

- Wo begegnen Ihnen im normalen Leben Datenbanken?
Welche enthalten Informationen über Sie?
 - Supermarkt (Scannerkasse)
 - _____
 - _____
 - _____
 - _____
 - _____

Motivation (2)

- Was würden Sie gerne mal in Ihrem späteren Beruf machen?
 - Gibt es da persistente, strukturierte Daten?
 - Mehr als 100 Zeilen Spreadsheet/CSV/JSON/XML?
- Haben Sie Skype auf Ihrem Rechner?

Wussten Sie, dass es Daten wie z.B. Kontakte in einer SQLite-DB verwaltet?

[\[https://stackoverflow.com/questions/44502283/skype-how-to-read-chat-message-using-main-db\]](https://stackoverflow.com/questions/44502283/skype-how-to-read-chat-message-using-main-db)
- Nutzen Sie gitlab? Es verwendet PostgreSQL.

[\[https://gitlab.com/gitlab-org/gitlab/-/blob/master/db/structure.sql\]](https://gitlab.com/gitlab-org/gitlab/-/blob/master/db/structure.sql)
[\[https://docs.gitlab.com/ee/administration/troubleshooting/postgresql.html\]](https://docs.gitlab.com/ee/administration/troubleshooting/postgresql.html)
- Facebook hat GraphQL entwickelt (nicht relationale DB).

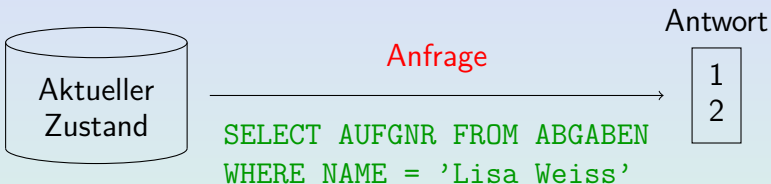
[\[https://de.wikipedia.org/wiki/GraphQL\]](https://de.wikipedia.org/wiki/GraphQL) [\[https://graphql.org/\]](https://graphql.org/)

Inhalt

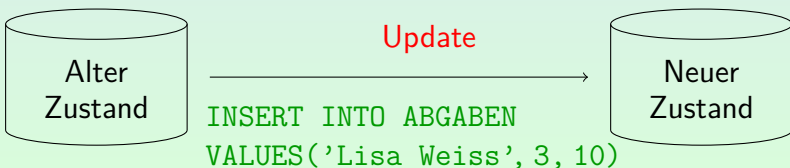
- 1 Organisatorisches
- 2 Motivation
- 3 Wiederholung**
- 4 DB-Entwurf
- 5 Einführung zum Adminer
- 6 Präsenzaufgaben

Wichtige Begriffe: DB-Zustand, Anfrage, Update

- Menge der gespeicherten Daten = DB-Zustand:



- Eingabe, Modifikation oder Löschung von Daten ändert den Zustand:



Zustand vs. Schema

- Das Schema definiert die Struktur der Datenbank.
- Im relationalen Datenmodell sind die Daten in Form von Tabellen (Relationen) strukturiert (oft mehrere/viele).
- Schema einer Tabelle: Tabellen-Name, Folge von benannten Spalten („Attribute“), jeweils mit Datentyp.
- Zustand: Menge von Zeilen (Tupel)

ABGABEN		
NAME	AUFGNR	PUNKTE
Lisa Weiss	1	10
Lisa Weiss	2	8
Daniel Sommer	1	9
Daniel Sommer	2	9

} DB-Schema

} DB-Zustand

Aufgabe

Welche Aussage ist nicht richtig?

- (A) Ein Datenbank-Schema legt die Menge aller möglichen DB-Zustände fest, die in der Datenbank gespeichert werden können.
- (B) Beim Anlegen einer Datenbank wird das Datenbank-Schema festgelegt.
- (C) Ein Datenbankzustand ist die Menge der aktuell gespeicherten Daten.
- (D) Im relationalen Datenmodell beschreibt ein DB-Zustand die Namen von Tabellen und deren Spalten.
- (E) Durch das Ausführen eines Insert-Befehls kann sich der Datenbankzustand ändern.

Datenmodell

- Ein Datenmodell (wie das relationale Modell) definiert
 - eine Menge SCH möglicher DB-Schemata,
 - für jedes DB-Schema $S \in SCH$ ein Menge $ST(S)$ möglicher DB-Zustände.
 - Eine Menge A möglicher Antworten auf Anfragen.
- Typischerweise durch Basis-Datentypen parametrisiert. Das Datenmodell definiert dann Typ-Konstrukturen, um komplexe Datenstrukturen aus Basis-Datentypen zu bilden.
- Oft kann man im Schema auch mittels Integritätsbedingungen (engl. "Constraints") die möglichen Zustände einschränken.
- Es ist auch sehr typisch, dass DB-Schemata Symbole/Namen einführen (wie z.B. Tabellennamen, Spaltennamen).

Anfragesprachen zu einem Datenmodell

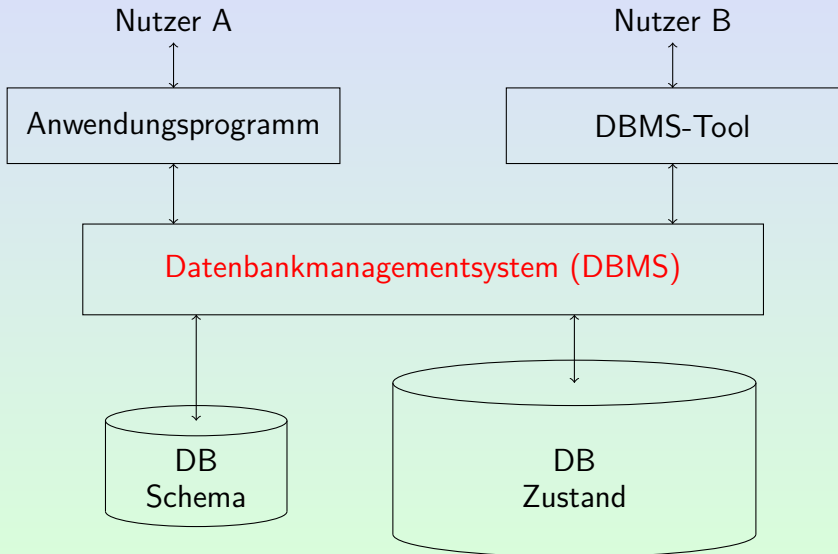
- Anfragen sind Zeichenketten einer Anfragesprache.
Genauer: Für jedes Schema S gibt es eine Menge möglicher Anfragen Q_S .
- Die Semantik einer Anfrage $Q \in Q_S$ ist eine Abbildung von Datenbank-Zuständen $ST(S)$ in Antworten \mathcal{A} .
- Bei Hausaufgaben oder der Klausur reicht es nicht, wenn die Anfrage im Beispiel-Zustand die erwartete Antwort liefert. Sie muss in allen Zuständen korrekte Antworten liefern.
- Zwei Anfragen sind äquivalent, wenn sie in allen Zuständen jeweils die gleichen Antworten liefern.
- Zwei Anfragesprachen sind gleich mächtig, wenn man Anfragen zwischen ihnen übersetzen kann, so dass man jeweils die gleiche Funktion von Zuständen in Antworten erhält.

Datenmodell: Beispiele

Aufgabe:

- Nennen Sie einige Datenmodelle (oder beschreiben Sie die Struktur, oder nennen Sie ein DBMS als typischen Vertreter des Datenmodells — denken Sie auch an NoSQL Systeme):
 - Relationales Modell
 - _____
 - _____
 - _____
 - _____
 - _____

Datenbank-Managementsystem (DBMS)



Etwas Datenbank-Vokabular

- Eine **DB** besteht aus DB-Schema und DB-Zustand.

Z.B. sagt man „Hausaufgaben-Datenbank“. Es hängt vom Kontext ab, ob man den derzeitigen Zustand meint, oder nur das Schema und den Speicherplatz oder -ort (Netzwerk-Adresse des Servers).

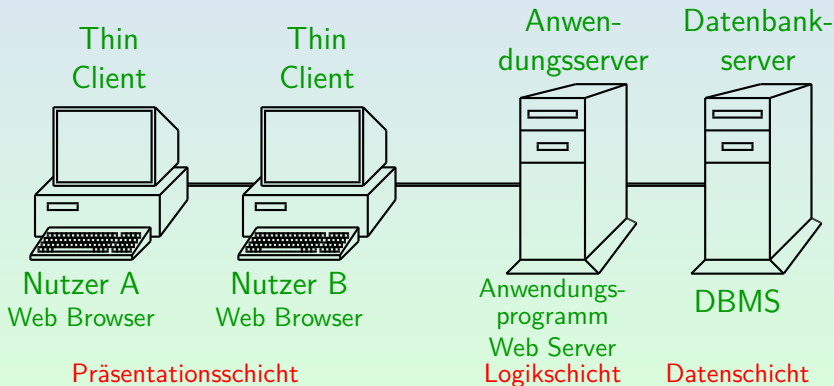
Es ist falsch, eine einzige Tabelle oder Datei Datenbank zu nennen, außer sie beinhaltet alle Daten des Schemas.
- Ein **Datenbank-System (DBS)** besteht aus einem DBMS und einer Datenbank.

Aber „Datenbank-System“ wird auch als Abkürzung für DBMS genutzt.
- Ein **Datenbank-Anwendungssystem** besteht aus einem DBS und Anwendungsprogrammen.

Ein Anwendungsprogramm unterstützt eine Aufgabe in der realen Welt und greift dazu auf die Datenbank zu (mit Anfragen und/oder Updates).

Datenbank-Anwendungssysteme

Dreischichten-Architektur (Three-Tier Architecture):



Inhalt

- 1 Organisatorisches
- 2 Motivation
- 3 Wiederholung
- 4 DB-Entwurf**
- 5 Einführung zum Adminer
- 6 Präsenzaufgaben

Übung zum DB-Entwurf (1)

- Oben wurde schon folgende einfache Tabelle zur Verwaltung von Punkten für Hausaufgaben betrachtet:

ABGABEN		
NAME	AUFGNR	PUNKTE
Lisa Weiss	1	10
Lisa Weiss	2	8
Daniel Sommer	1	9
Daniel Sommer	2	9

- Der Datenbank-Entwurf beschäftigt sich mit der Entwicklung von Datenbank-Schemata für gegebene Anwendungen.
- Dabei ist wichtig, was man mit den Daten machen möchte.

Übung zum DB-Entwurf (2)

- Der Professor möchte ein Programm entwickeln, das an jeden Studenten eine E-Mail folgender Art verschickt:

Sehr geehrte Frau Weiss,
folgende Bewertungen sind über Sie gespeichert:
- Aufgabe 1: 10 Punkte
- Aufgabe 2: 8 Punkte
Melden Sie sich bitte, falls ein Fehler vorliegt.
Mit freundlichen Grüßen, ...

- Muss er dazu obige Tabelle erweitern?
Sollte er vielleicht die Daten auf mehrere Tabellen verteilen?

Inhalt

- 1 Organisatorisches
- 2 Motivation
- 3 Wiederholung
- 4 DB-Entwurf
- 5 Einführung zum Adminer**
- 6 Präsenzaufgaben

Beispiel-Datenbank (1)

STUDENTEN

<u>SID</u>	<u>VORNAME</u>	<u>NACHNAME</u>	<u>EMAIL</u>
101	Lisa	Weiss	...
102	Michael	Grau	NULL
103	Daniel	Sommer	...
104	Iris	Winter	...

AUFGABEN

<u>ATYP</u>	<u>ANR</u>	<u>THEMA</u>	<u>MAXPT</u>
H	1	ER	10
H	2	SQL	10
Z	1	SQL	14

BEWERTUNGEN

<u>SID</u>	<u>ATYP</u>	<u>ANR</u>	<u>PUNKTE</u>
101	H	1	10
101	H	2	8
101	Z	1	12
102	H	1	9
102	H	2	9
102	Z	1	10
103	H	1	5
103	Z	1	7

Beispiel-Datenbank (2)

- **STUDENTEN**: enthält eine Zeile für jeden Studenten.
 - **SID**: "Studenten-ID" (eindeutige Nummer).
 - **VORNAME**, **NACHNAME**: Vor- und Nachname.
 - **EMAIL**: Email-Adresse (kann NULL sein).
- **AUFGABEN**: enthält eine Zeile für jede Aufgabe.
 - **ATYP**: Typ/Kategorie der Aufgabe.
Z.B. 'H': Hausaufgabe, 'Z': Zwischenklausur, 'E': Endklausur.
 - **ANR**: Aufgabennummer (innerhalb des Typs).
 - **THEMA**: Thema der Aufgabe.
 - **MAXPT**: Maximale/volle Punktzahl der Aufgabe.

Beispiel-Datenbank (3)

- **BEWERTUNGEN**: enthält eine Zeile für jede abgegebene Lösung zu einer Aufgabe.
 - **SID**: Student, der die Lösung abgegeben hat.
Dies referenziert eine Zeile in **STUDENTEN**.
 - **ATYP**, **ANR**: Identifikation der Aufgabe.
Zusammen identifiziert dies eine Zeile in **AUFGABEN**.
 - **PUNKTE**: Punkte, die der Student für die Lösung bekommen hat.
 - Falls es keinen Eintrag für einen Studenten und eine Aufgabe gibt: Aufgabe nicht abgegeben.

Adminer (1)

- Adminer ist eine Web-Schnittstelle zum Zugriff auf verschiedene Datenbanksysteme.
 - Aktuell (Version 4.7.7): MySQL, MariaDB, PostgreSQL, SQLite, MS SQL, Oracle, Firebird, SimpleDB, Elasticsearch and MongoDB.
- [<https://www.adminer.org/de/>]
[<https://en.wikipedia.org/wiki/Adminer>]
- Die wichtigste Funktion ist natürlich, dass man
 - SQL-Anfragen in ein Web-Formular eingeben kann,
 - auf den „Ausführen“-Knopf klicken, und
 - dann die Ergebnis-Tabelle angezeigt bekommt.
- Man kann aber z.B. auch nur durch Anklicken einer Tabelle das Schema und den Inhalt anzeigen lassen (GUI).

Adminer (2)

- Die Zugangsdaten unserer Installation stehen in StudIP, Reiter „Adminer“.
- [https://dbs.informatik.uni-halle.de/edb?pgsql=db&username=student_gast&db=postgres&ns=]
- Man kann Daten des Login-Bildschirms in der URL vorbelegen („ns“ steht für „Namespace“, d.h. Schema).
- Ein PostgreSQL Server kann mehrere Datenbanken verwalten.
Es gibt immer: postgres, template0, template1.
- Eine PostgreSQL-Datenbank enthält mehrere Schemata.
Es gibt immer das Schema public.

Adminer (3)

- Die Beispiel-DB steht im Schema

„**studentenaufgaben_public**“.

Sie können das in der Auswahl-Box links relativ weit oben wählen.

Sie bekommen dann eine Liste/Tabelle mit den drei Tabellen „aufgaben“, „bewertungen“, „studenten“ angezeigt (unter der Überschrift „Tabellen und Views“). Es gibt noch weitere Überschriften für andere Arten von Datenbank-Objekten, davon gibt es aber keine in diesem Schema.

- Wenn Sie auf einen Tabellennamen klicken, bekommen Sie die Schema-Information zur Tabelle.
- Wenn Sie auf „zeigen“ links neben dem Tabellennamen klicken, werden die Tabellendaten (der Zustand) angezeigt.
- Wenn Sie auf „SQL-Kommando“ im Menu links klicken, bekommen Sie eine Dialogbox für SQL-Befehle.

Zum Ausprobieren

- Einfache SQL-Anfragen haben die Form:

```
SELECT <Spalten>
FROM   <Tabelle>
WHERE  <Bedingung>
```

- Beispiel:

```
SELECT VORNAME, NACHNAME, EMAIL
FROM   STUDENTEN
WHERE  NACHNAME = 'Weiss';
```

Die Zeilenaufteilung ist egal. SQL ist eine formatfreie Sprache wie Java.
Groß-/Kleinschreibung ist nur für die Daten (wie 'Weiss') relevant,
z.B. nicht für SELECT. Das ";" am Ende gehört eigentlich nicht dazu.

- Anzeigen der ganzen Tabelle: `SELECT * FROM STUDENTEN;`

Inhalt

- 1 Organisatorisches
- 2 Motivation
- 3 Wiederholung
- 4 DB-Entwurf
- 5 Einführung zum Adminer
- 6 Präsenzaufgaben**

Präsenzaufgabe: DBMS Daten und Webseiten

- Geben Sie eine Liste mit drei bekannten DBMS ab:
 - Name
 - Webadresse der Homepage des Projekts
 - Webadresse der Download-Seite (falls möglich)
 - Webadresse der offiziellen Dokumentation (falls möglich)
 - Webadresse des Wikipedia-Artikels
 - Aktuelle Versions-Nummer
 - Jahr des Projekt-Starts (soweit möglich, ggf. ungefähr).
- Sie müssen als Gruppen aus den „Breakout-Rooms“ abgeben.
Bitte in der Übungsplattform in StudIP abgeben (als Gruppe). Es reicht eine ASCII .txt Datei. Sie z.B. über die „geteilten Notizen“ kommunizieren.
Oder [<https://sharelatex.informatik.uni-halle.de/>].