

Einführung in Datenbanken

Übung 1: Einführung

Prof. Dr. Stefan Brass

PD Dr. Alexander Hinneburg

Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2023/24

<http://www.informatik.uni-halle.de/~brass/db23/>

Inhalt

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

Haben Sie organisatorische Fragen?

- Webseite (Folien, Links, aktuelle Ankündigungen):
<https://users.informatik.uni-halle.de/~brass/db23/index.html>
- Aufteilung von „Datenbanken I“ (10 LP) in
 - „Einführung in Datenbanken“ (5 LP, Winter)
 - „Datenbank-Programmierung“ (5 LP, Sommer)
- **Modul-Anmeldung nicht vergessen!**

Man kann sich auch zu „Datenbanken I“ im Löwenportal noch anmelden.

- Vorlesungs-Termin: Montag, 10¹⁵–11⁴⁵, Raum 307

Vorlesungs-Aufzeichnung

- Aufzeichnung in **StudIP**, Reiter „Opencast“.
- Sie sollten die Aufzeichnung vor der Übung ansehen (falls Sie nicht in Präsenz teilgenommen haben).
- Die Aufzeichnung steht nicht direkt nach der Vorlesung zur Verfügung, sondern erst ab ca. 17 Uhr (?).

Es geschieht zwar alles automatisch, aber die Verarbeitung der Daten dauert etwas.

- Normalerweise schneide ich am Abend noch den Anfang und das Ende weg.

Vor dem Schneiden ist am Ende noch ein Stück, wo die Aufnahme zu hängen scheint, aber die Zeit noch etwas weiter läuft.

- Gab es Probleme?

Erreichbarkeit bei Fragen

- Hier in der Übung!
- Forum in **StudIP** (Eintrag der Vorlesung)
- Prof. Dr. Stefan Brass:
 - Sprechstunde: Montags, 12⁰⁰–13⁰⁰
Raum 313, Tel. 0345/55-24740
 - EMail: brass@informatik.uni-halle.de
- PD Dr. Alexander Hinneburg:
 - (nur während Übung)

Klausur (2)

- Beim ersten Termin können Sie 10 Seiten (DIN A4) mit beliebigen Notizen verwenden, beim zweiten 6 Seiten.

Es muss auf Papier vorliegen, Notebooks/Tablets/Handys sind nicht erlaubt.

Es ist eine gute Klausurvorbereitung, sich selbst das Wichtigste auf begrenztem Platz in eigener Strukturierung aufzuschreiben (inkl. „SQL Quick Reference“). Sie können sich natürlich auch etwas aus dem Netz drucken (wenn Sie sich damit nicht vorher beschäftigen, wird es Ihnen in der Klausur nichts nutzen, nur Zeit kosten).

Sie können sich für den zweiten Termin länger vorbereiten, deswegen gibt es da weniger Seiten.

- Praktische Anwendung, kein Auswendiglernen.

Ca. 40–70%: SQL-Anfragen.

Ca. 25–40%: DB-Entwurf (ER, Logischer Entwurf, BCNF).

Außerdem u.a. Logik, Relationale Algebra.

Alte Klausuren zu „Datenbanken I“ im Netz. Eventuell spezielle Probeklausur.

Klausur (3)

- Geplant ist eine elektronische Klausur in Präsenz.
- Wer hat schon eine elektronische Klausur im Prüfungscenter der MLU geschrieben?

Lührmann-Gebäude, Mansfelder Str. 15, Haltestelle „Saline“.

[\[https://media.llz.uni-halle.de/bilder/UPC-Standort.png\]](https://media.llz.uni-halle.de/bilder/UPC-Standort.png)

- Wer hat schon die Klausursoftware „ILIAS“ verwendet?
- Voraussichtlich gibt es bei der Klausur die Möglichkeit, SQL-Anfragen zu testen (mit PostgreSQL über Adminer).
Dann würden Syntaxfehler und offensichtlich falsche Ergebnisse zu härteren Punktabzügen führen, als wenn es keine Testmöglichkeit gibt.
Es nützt aber auch nichts, Anfragen zu basteln, die nur im Testzustand mit irgendwelchen Tricks die verlangte Antwort liefern. Ich würde erwarten, dass Sie mit typischen Fehlermeldungen von PostgreSQL umgehen können.
Für langes Trail&Error gibt es keine Zeit. Nutzen Sie Hausaufgaben+Übung!

Klausur (4)

- Es ist nicht garantiert, dass man mit 50% der Punkte in der Klausur bestanden hat.

Garantiert ist das Bestehen mit 60% (die Grenze wird häufig bis in die Nähe der üblichen 50% gesenkt, aber durch die elektronische Klausur brauchen wir auch etwas Flexibilität).

Ab 95% der Punkte ist die 1.0 garantiert, aber auch diese Grenze wird häufig noch etwas gesenkt bis max. ca. 90%.

- Man kann die Klausur nur mitschreiben, wenn man angemeldet ist. Dazu ist die Modulanmeldung nötig.
- Man bekommt die Leistungspunkte für dieses Modul nur, wenn man auch die Studienleistung hat (Hausaufgaben, s.u.).

Man kann die Klausur ohne Studienleistung mitschreiben, aber das macht wenig Sinn: Die Hausaufgaben sollen ja auf die Klausur vorbereiten.

Klausur (5)

- Notenverteilung (1. Termin) über die Jahre 2020 bis 2022:

Note	Anzahl	Prozent	kumulativ
1.0	53	15.3	15.3
1.3	19	5.5	20.8
1.7	11	3.2	24.0
2.0	31	9.0	32.9
2.3	44	12.7	45.7
2.7	23	6.6	52.3
3.0	31	9.0	61.3
3.3	22	6.4	67.6
3.7	17	4.9	72.5
4.0	7	2.0	74.6
5.0	88	25.4	100.0

Klausur (6)

Problem:

- Es ist bekannt, dass die Klausur fünf SQL-Anfragen enthält, die nicht ganz einfach sind, aber vergleichbar mit Hausaufgaben in der Vorlesung (eher gegen Ende).

Es gibt auch alte Klausuren im Internet. Die Aufgabentypen sind größtenteils ähnlich, es ist aber jedes Mal ein neues Datenbank-Schema, also eine neue Anwendung.

- Z.B. bei Aufgabe a) in der Klausur im Februar 2023 hatten nur 12 von 84 (also 14%) die volle Punktzahl.

Mindestens 90% der Punkte hatten 19 (also 23%).

Mindestens 80% der Punkte: 35 (42%).

- Das war noch eine der besseren Aufgaben.

Klausur (7)

Warum ist das Klausurergebnis nicht besser?

- Es ist ein neues Datenbank-Schema, in das man sich erst hineindenken muss.

In den Übungen gibt es auch mehr als ein Datenbank-Schema, aber nicht jede Woche ein anderes.

- Die Zeit ist relativ knapp.

Es sind 10 Minuten pro Anfrage geplant, es gibt aber etwas Luft am Ende (also max. 15 Minuten). Die Musterlösung hatte mit einer schönen Formatierung 10 Zeilen (47 Worte, 348 Zeichen).

- Vielleicht werden die Hausaufgaben nicht ausreichend ernst genommen?

Es gibt auch viele Dinge, die ich in der Vorlesung klar gesagt habe (und die auf den Folien stehen), die aber bei nicht wenigen Studierenden nicht angekommen sind.

Erwerb der Leistungspunkte

Studienleistung:

- Mindestens 50% der Hausaufgabenpunkte.

Bei Syntaxfehlern in SQL-Anfragen gibt es 0 Punkte. Testen Sie Ihre Abgaben!

Es gibt voraussichtlich auch Präsenzaufgaben in den Übungen, diese geben auch Hausaufgabenpunkte und werden mitgerechnet. Die Präsenzaufgaben können in Gruppen bearbeitet werden.

- Aktive Mitarbeit in den Übungen (u.a. Vorrechnen von HAen)

Sie müssen für 20% der Aufgaben erklären, dass Sie bereit wären, vorzurechnen (und das ggf. dann auch kompetent tun). Genauer siehe unten.

- Die Studienleistung ist Voraussetzung dafür, dass Sie die Leistungspunkte für dieses Modul bekommen.

Sie hat keinen Einfluss auf die Note. Sie ist auch nicht Voraussetzung für die Teilnahme an der Klausur, aber es wäre uncles, zur Klausur anzutreten, wenn Ihnen die Übung durch die Hausaufgaben fehlt.

Plagiage, Datenschutz-Hinweis

- Die Hausaufgaben sind einzeln zu bearbeiten.

Selbstverständlich sollen Sie einander helfen, aber nicht eine Lösung zum abschreiben geben.

- Bei Plagiaten gibt es 0 Punkte für alle Beteiligten (d.h. auch Autor/Autorin des Originals).

- Im Wiederholungsfall droht Schlimmeres.

Z.B. Ausschluss von der Übung/Verweigerung der Studienleistung.

- Datenschutz-Hinweis:** Wir loggen die Zugriffe auf den Adminer (Web-Schnittstelle zum Ausprobieren der Anfragen) und verwenden die Daten zur Plagiatskontrolle und für ein Forschungsprojekt.

ChatGPT

- ChatGPT (KI-Chatbot) hat die Situation verändert.
- Sie könnten sich von diesem Programm SQL-Anfragen schreiben lassen (**Wir raten ab!**).
- Wir können das vermutlich nur selten zweifelsfrei nachweisen.

Manchmal schon, wenn ChatGPT ganz merkwürdige Lösungen produziert. Außerdem haben Sie das Risiko, untereinander des Plagiats beschuldigt zu werden, wenn mehrere Studierende ChatGPT nutzen. ChatGPT produziert aber unterschiedliche Lösungen, auch für die gleiche Eingabe.
- Manche der generierten Anfragen sind falsch.

Aber für 50% der Übungspunkte würde es sicher reichen.
- **Es bleibt also eine ganz wesentliche Kompetenz, SQL-Anfragen auch selbst schreiben zu können.**

Sie müssen ja wenigstens die Korrektheit kontrollieren.

Vorschlag/Idee/Forschungsprojekt (1)

- Sie können bei den Hausaufgaben (nicht bei der Klausur!) beliebig abschreiben und ChatGPT verwenden.
- Bedingung ist aber, dass Sie die Quelle angeben (also auch den Namen des/der Mitstudierenden, der/die die originale Lösung verfasst hat).

Oder eben ChatGPT oder eine andere KI, oder eine Webseite, oder eine Sammlung von Lösungen aus einem früheren Semester. Wir werden die Daten natürlich nur anonymisiert für unser Forschungsprojekt verwenden. Sie können auch „(modifiziert)“ (bei stilistischen Verbesserungen) oder „(überarbeitet)“ (bei inhaltlichen Veränderungen) dazu schreiben. Mehr Details morgen in der Übung und auf der Webseite zur Vorlesung.

- Beispiel:

-- Quelle: Müller, Monika (modifiziert)

Vorschlag/Idee/Forschungsprojekt (2)

- Beispiel:

-- Quelle: ChatGPT

- Wenn Sie ChatGPT als Quelle angeben, bewahren Sie bitte den „Prompt“ (also Ihre Eingaben, die zur Anfrage geführt haben) auf.

Wir fragen stichprobenmäßig nach (besonders bei Anfragen, die nicht nach ChatGPT aussehen). Bei Überarbeitung bewahren Sie bitte auch die Original-Anfrage von ChatGPT auf.

- Die Sache ist nicht so ungewöhnlich, wie es zuerst aussieht:
 - Z.B. bei Hausarbeiten müssten Sie ja auch die verwendeten Quellen korrekt zitieren.

Vorschlag/Idee/Forschungsprojekt (3)

- Wenn Sie keine Quelle angeben, verstehen wir das als Selbständigkeitserklärung.
- Dann gibt es eine normale Plagiatskontrolle, auch ein Vergleich mit Lösungen von ChatGPT.
- Mit der Angabe des Autors/der Autorin schützen Sie auch Ihren Mitstudenten/Ihre Mitstudentin:
 - Er/Sie würde, falls die Kopie erkannt wird, und die Quellenangabe fehlt, auch 0 Punkte bekommen.
 - Dann wäre es ein Täuschungsversuch.
- **Wenn Sie zu viele Anfragen nicht selbst entwickeln, werden Sie bei der Klausur durchfallen!**

Aber die Studienleistung bekommen Sie. Wir ziehen keine Punkte für die Quellenangabe ab.

Vorschlag/Idee/Forschungsprojekt (4)

- In jedem Fall (eigene Lösung oder mit Quelle) müssen Sie angeben, ob Sie bereit wären vorzurechnen:
 - --Vorrechnen:5: Sehr gerne (auch bei Fehlern möglich).
 - --Vorrechnen:4: Kein Problem.
 - --Vorrechnen:3: Bei Bedarf (nicht gerne).
 - --Vorrechnen:2: Ungerne, aber wenn es sein muss.
Dies reicht zum Erreichen der 20% Quote. Aber es muss voraussichtlich mal sein.
 - --Vorrechnen:1: Nein.
Das wird respektiert. Je nach Vorbereitung des Übungsleiters werden Sie möglicherweise aufgerufen, können dann aber sagen, dass Sie diese Aufgabe nicht vorrechnen wollen.
 - --Vorrechnen:0: Ich werde nicht da sein.

Hausaufgaben

- Diese Woche gibt es noch keine Hausaufgaben.
- Sonst Ausgabe immer Montag bis spätestens 21:00 im Netz, Abgabe bis Montag 18:00 der nächsten Woche.

Ausgabe und Abgabe erfolgen über die Übungsplattform in StudIP.

Genauere Angaben zum Abgabeformat finden sich jeweils auf dem Übungsblatt: SQL-Anfragen bitte als „Plain Text“ (ASCII), so dass sie ggf. ausprobiert werden können. Ansonsten bitte PDF, kein Word.

- Zur Klarstellung: Präsenzaufgaben sollen nur von Übungsteilnehmern bearbeitet werden (auch online).

Es ist nicht erwünscht, dass man sie ohne Übungsteilnahme abgibt. Sie können auch nur während der Übungszeiten abgegeben werden.

Vorkenntnisse (1)

- Kennen Sie schon SQL?

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

A. Sehr gut (z.B. auch **GROUP BY** und **NOT EXISTS**).

B. Etwas (**SELECT ... FROM ... WHERE ...**).

C. Nein.

- Woher?

A. Schule

B. Andere Vorlesung

C. Berufstätigkeit

D. Interesse (Buch, Webseiten)

Vorkenntnisse (2)

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

Es gibt viele verschiedene Varianten.

- A. Gut
- B. Etwas
- C. Nein

- Wer kennt UML Klassendiagramme?

Das ist ähnlich zu Entity-Relationship-Diagrammen.

- A. Gut
- B. Etwas
- C. Nein

Vorkenntnisse (3)

- Wer kennt Syntaxdiagramme (kontextfreie Grammatiken)?
 - A. Gut
 - B. Etwas
 - C. Nein
- Wer kann programmieren?

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

 - A. Ja, in Java (und ggf. anderen Sprachen)
 - B. Ja, aber nicht in Java
 - C. Ein wenig (in irgendeiner Sprache)
 - D. Nein

Vorkenntnisse (4)

- Wer kennt Aussagenlogik?
 - A. Ich weiß, wozu $\neg(p \wedge q)$ äquivalent ist.
 - B. Das nicht, aber ich kenne $\wedge$.
 - C. Weder noch.

Inhalt

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

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>]

- Nutzen Sie gitlab? Es verwendet PostgreSQL.

[<https://gitlab.com/gitlab-org/gitlab/-/blob/master/db/structure.sql>]

[<https://docs.gitlab.com/ee/administration/troubleshooting/postgresql.html>]

- Facebook hat GraphQL entwickelt (nicht relationale DB).

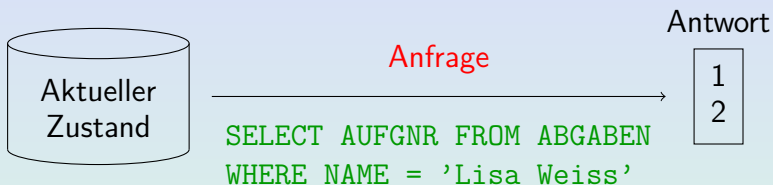
[<https://de.wikipedia.org/wiki/GraphQL>] [<https://graphql.org/>]

Inhalt

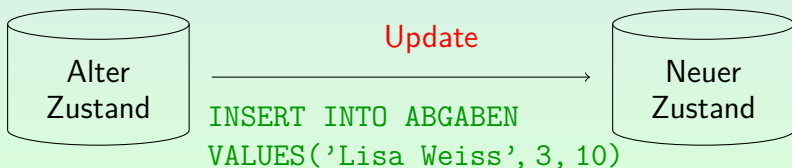
- 1 Organisatorisches
- 2 Motivation
- 3 Wiederholung**
- 4 DB-Entwurf
- 5 Einführung zum Adminer
- 6 Präsenzaufgabe

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			} DB-Schema
NAME	AUFGNR	PUNKTE	
Lisa Weiss	1	10	} DB-Zustand
Lisa Weiss	2	8	
Daniel Sommer	1	9	
Daniel Sommer	2	9	

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. Eine Datenbank besteht immer aus genau einer Tabelle.
- E. Durch das Ausführen eines Insert-Befehls kann sich der Datenbankzustand ändern.

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

- _____
- _____
- _____
- _____
- _____

Inhalt

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

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 Präsenzaufgabe

Software für die Übungen

- In den Übungen wird in erster Linie PostgreSQL verwendet.
- In der Übung sollten nur selten spezielle Features eines Systems wichtig sein.

Exotische SQL Features können als falsch gewertet werden!

- Ich versuche, den Kern des SQL Standards zu unterrichten, der von fast allen Systemen unterstützt wird.
- In dieser Vorlesung reicht die Web-Schnittstelle „Adminer“, für „DB-Programmierung“ müssen Sie sich PostgreSQL installieren (ggf. als Docker Container).
- Falls Sie sich für Datenbanken besonders interessieren, wäre wohl vernünftig, sich auch eine Express Edition eines kommerziellen Systems zu installieren.

Beispiel-Datenbank (1)

STUDENTEN

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

AUFGABEN

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

BEWERTUNGEN

<u>SID</u>	<u>ATYP</u>	<u>ANR</u>	PUNKTE
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;`

Weitere Webseiten zum Ausprobieren von SQL

- SQL Fiddle (MySQL, Oracle, PostgreSQL, SQLite, MS SQL)

[<http://sqlfiddle.com/>]

- Oracle Live SQL (Oracle)

[<https://livesql.oracle.com/>]

Benötigt kostenlose Registrierung.

- JDoodle: Online SQL Editor (SQLite)

[<https://www.jdoodle.com/execute-sql-online/>]

- JDoodle: Online MySQL Terminal (MySQL)

[<https://www.jdoodle.com/online-mysql-terminal/>]

- codingground: Exceute SQL Online (SQLite)

[https://www.tutorialspoint.com/execute_sql_online.php]

Inhalt

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

Präsenzaufgabe: SQL Anfrage

- Gehen Sie auf die Webseite

[<https://dbup2date.uni-bayreuth.de/wetterdaten.html>]

Es wird eine MySQL-Datenbank offenbar auf einem Linux-Rechner verwendet. Die Groß-/Kleinschreibung der Tabellennamen ist wichtig. Dies ist also nicht der Adminer und nicht PostgreSQL, aber auch eine Web-Schnittstelle zu einer SQL-Datenbank.

- Gesucht sind die Standorte aller Wetterstationen, deren Höhe über 1000m ist.
- Geben Sie die SQL-Anfrage für diese Aufgabe ab.
- 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/>].