Einführung in Datenbanken

Übung 1: Einführung

Prof. Dr. Stefan Brass
PD Dr. Alexander Hinneburg

Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2024/25

http://www.informatik.uni-halle.de/~brass/db24/

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/db24/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, 10^{15}–11^{45}, Raum 307
- Es gibt diese Woche noch keine Hausaufgabe.

  Die Hausaufgaben werden immer Montag abend auf der Webseite und in der Übungsplattform in StudIP veröffentlicht (bis ca. 21^{00}). D.h. Sie könnten in der Übung Fragen stellen. Abgabe: Bis Montag der nächsten Woche 18^{00}.

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, 13^{00}–13^{45}

    Raum 313, Tel. 0345/55-24740.
    12^{15}–13^{00} bin ich beim freiwilligen Tutorium im Multimedia-Pool 3.02 erreichbar. Wenn es da keine Fragen gibt, schon in meinem Büro.
  - EMail: brass@informatik.uni-halle.de
- PD Dr. Alexander Hinneburg:
  - (nur während Übung)

Klausur (1)

- Klausur (E-Klausur in Präsenz), aktuell geplante Termine:
  - Montag, 03.03.2025, 10^{15}–12^{15}

    Fünf Wochen nach letzter Vorlesung (27.01.2025).

    [Kalender 2025 mit Schulferien]
  - Montag, 14.07.2025, 10^{15}–12^{15}

    Direkt nach der letzten Woche der Vorlesungszeit im Sommersemester.

    [Studienjahresablauf]
- Die Termine sind im Institut abgestimmt und schwer zu ändern.
- Natürlich sollten Sie trotzdem Überschneidungen möglichst frühzeitig melden.

  Aber auch dem anderen Professor. Dies ist eine relativ große Klausur für eine ganze Reihe von Studiengängen.

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. 40–60%: SQL-Anfragen.
  Ca. 25–40%: DB-Entwurf (ER, Logischer Entwurf, BCNF).
  Außerdem u.a. Logik, Relationale Algebra, `CREATE TABLE`.
  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]

- Wer hat schon die Klausursoftware „ILIAS“ verwendet?
- Voraussichtlich gibt es bei der Klausur die Möglichkeit, SQL-Anfragen zu testen (mit PostgreSQL über Adminer).

  Es nützt aber nichts, Anfragen zu basteln, die nur im Testzustand mit irgendwelchen Tricks die verlangte Antwort lieferen. 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 sollen in Gruppen bearbeitet werden (die Hausaufgaben können es).

- 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 unclever, zur Klausur anzutreten, wenn Ihnen die Übung durch die Hausaufgaben fehlt.

Forschungsprojekt (1)

- Sie müssen für jede Hausaufgabe angeben, ob und wie gerne Sie bereit wären, diese Aufgabe vorzurechnen.
- Bei SQL-Aufgaben ist dies als Kommentar `-- VORRECHNEN:N` zu formatieren, wobei `N` einen Wert von 0 bis 5 hat:
  - `0`: Ich werde nicht zur Übung kommen.
  - `1`: Ich möchte diese Aufgabe nicht vorrechnen.
  - `2`: Ich möchte diese Aufgabe nur ungern vorrechnen.
  - `3`: Ich kann vorrechnen, lasse aber anderen gern den Vortritt.
  - `4`: Ich kann problemlos vorrechnen.
  - `5`: Ich möchte gerne, dass meine Abgabe besprochen wird.
- Sie müssen für mindestens 20% der Aufgaben (bezogen auf die Gesamtpunktzahl) mindestens die Stufe 2 angeben.

Forschungsprojekt (2)

- Zweier-Gruppen sind erlaubt, dann müssen beide Mitglieder einzeln ihre Bereitschaft zum Vorrechnen erklären (Angaben `VORRECHNEN1:N` und `VORRECHNEN2:M`).
- Wenn Sie die Angabe vergessen, oder anders formatieren (so dass das Analyseprogramm sie nicht erkennt), zählt die Abgabe nicht für das 20% Ziel.

  Die Punkte bekommen Sie natürlich schon.

- Die Verwendung von ChatGPT ist nicht verboten, muss aber deklariert werden, wie in der Vorlesung ausführlich erklärt, z.B. „`-- ChatGPT:3`“ für vollständig von ChatGPT generierte Anfragen. Das ist aber nicht empfohlen!

  Auch weitere Quellen sind möglich, inklusive Lösungen von anderen Studenten. Sie müssen aber deklariert werden (Codierung siehe Vorlesung). Andernfalls handelt es sich um ein Plagiat (keine Quelle = Selbständigkeitserklärung).

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).

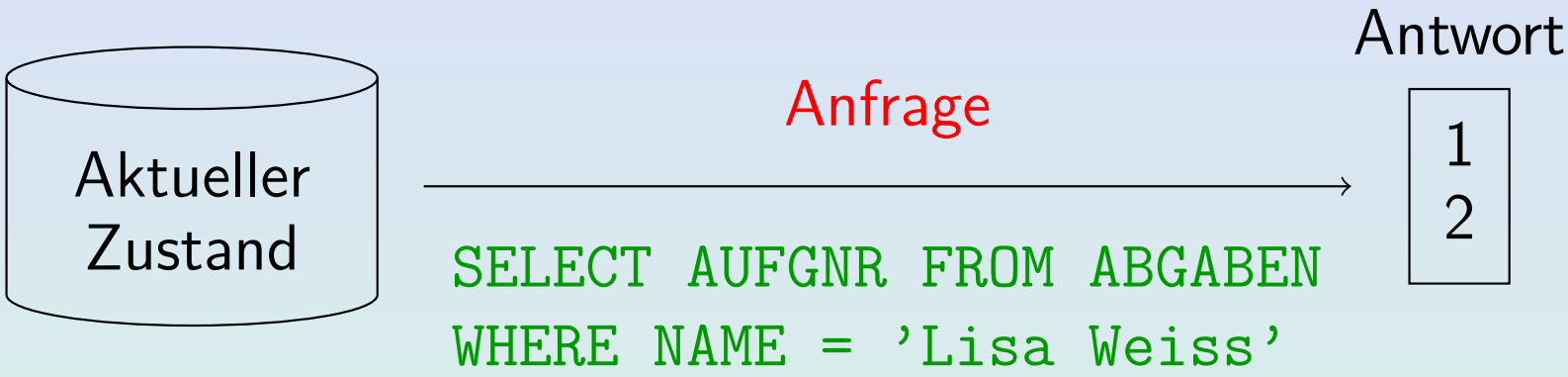
  [https://de.wikipedia.org/wiki/GraphQL] [https://graphql.org/]

Inhalt

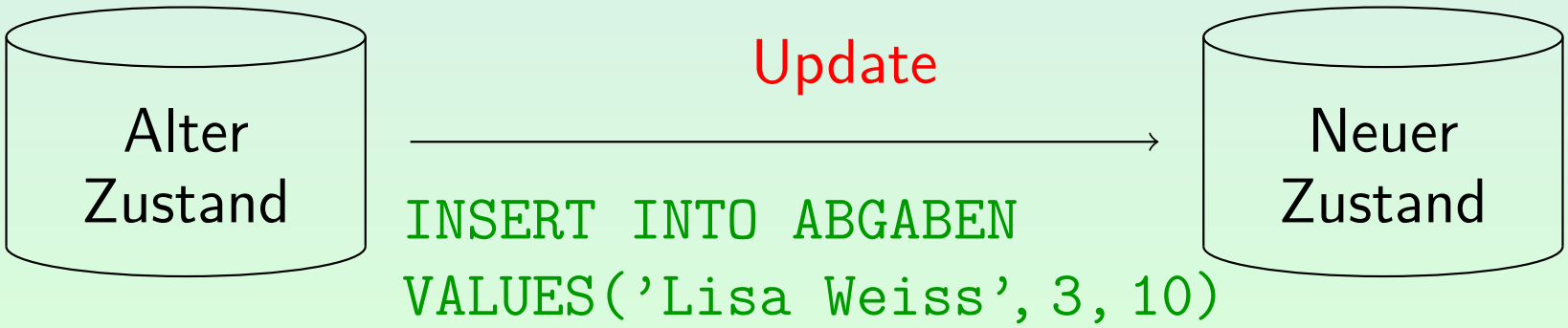
1 Organisatorisches
2 Motivation
3 Wiederholung
4 DB-Entwurf
5 Einführung zum Adminer
6 Präsenzaufgabe

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		
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. 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

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

BEWERTUNGEN

SID	ATYP	ANR	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

AUFGABEN

ATYP	ANR	THEMA	MAXPT
H	1	ER	10
H	2	SQL	10
Z	1	SQL	14

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`.
- Der Gast-Account hat keine Schreibrechte.

  Es kann also nichts kaputt gehen.

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: Vorkenntnisse

- Es werden Zettel ausgeteilt, auf denen Sie nach Ihren Vorkenntnissen befragt werden.

  Logik, UML Klassendiagramme oder ER-Diagramme, SQL, Syntaxformalismen.

  Teils mit kleinen Aufgaben zur Sicherheit/Kontrolle.

- Für vollständig ausgefüllte Zettel wird es 2 Punkte für die erste Präsenzaufgabe geben.

  Selbst dann, wenn Sie ankreuzen, dass Sie den Stoff nicht hatten oder leider vergessen haben.

- Es ist geplant, dass es in jeder Übung eine Präsenzaufgabe gibt, die auf Papier zu lösen ist.

  Leider waren in diesem Jahr nur Hörsaalübungen möglich.

- Normalerweise ist Gruppenarbeit erwünscht, diesmal bei der Abfrage von Vorkenntnissen aber natürlich nicht.