

Einführung in Datenbanken

Übung 2: DB-Managementsysteme und DBMS-Funktionen

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 Vorkenntnisse
- 2 Einführung zum Adminer
- 3 Hausaufgabe
- 4 DBMS-Funktionen
- 5 DB-Managementsysteme
- 6 Präsenzaufgabe

Zum Konzept von Präsenzaufgaben (1)

- In den letzten Jahren hatten wir Aufgaben, die in den Übungen gelöst werden mussten, und wie Hausaufgaben einige Punkte für die Studienleistung brachten.
- Die Lösungen konnten im Prinzip nur während der Übungszeit in die Übungsplattform hochgeladen werden.
 - Es war ausdrücklich nicht erwünscht, dass Nicht-Teilnehmer der Übung auch Lösungen abgegeben haben. Die (wenigen) Punkte für die Präsenzaufgaben sollten auch die Anwesenheit in der Übung belohnen.
- Im Gegensatz zu den Hausaufgaben war bei den Präsenzaufgaben Gruppenarbeit erwünscht.
 - Als die Übungen wegen Corona online gehalten werden mussten, wurden die Gruppen vom Videokonferenz-System zufällig zusammengestellt. So ergab sich die Gelegenheit, wenigstens einige sozialen Kontakte zu haben, und auch neue Studierende kennenzulernen.

Zum Konzept von Präsenzaufgaben (2)

- Wenn die Übung im Hörsaal gehalten wird, ist das etwas schwierig, da nicht jeder ein Notebook dabei hat.
- Wer hat ein Notebook oder ein ähnliches Gerät dabei?
 - Im Prinzip würde es auch mit einem Handy gehen, aber das wäre umständlich. Man sollte schon einen etwas größeren Bildschirm und eine Tastatur haben. Es würde reichen, wenn es in jeder Gruppe ein Notebook gibt.
- Wir haben uns entschlossen, in diesem Jahr vorerst die Abgaben von Präsenzaufgaben auf Papier zu machen.
 - Das bedeutet leider zusätzlichen Aufwand bei der Korrektur.
- Abgabe in Zweier-Gruppen ist ausdrücklich erwünscht.
 - Es spart uns etwas Arbeit und gibt Ihnen die Gelegenheit, andere Teilnehmende dieser Vorlesung kennenzulernen, und sich über Datenbanken und SQL auszutauschen.

Auswertung der Vorkenntnisse (1)

- In Gruppe 1 (12–14) gab es 22 Abgaben.
In StudIP hat die Gruppe inzwischen 46 Teilnehmer. Also 48% anwesend.
- In Gruppe 2 (10–12) gab es 40 Abgaben.
In StudIP hat die Gruppe inzwischen 70 Teilnehmer. Also 57% anwesend.
Es sind 9 Studierende zu beiden Übungsgruppen angemeldet,
35 nur zu Gruppe 1, 59 nur zu Gruppe 2, 13 zu keiner.
- Studiengänge der Teilnehmenden:

Studiengang	Anzahl
BSc Informatik	42
BSc Wirtschaftsinformatik	34
BSc Bioinformatik	19
Lehramt Informatik (Gym.)	8
Sonstiges (u.a. Physik, Mathematik)	17

Auswertung der Vorkenntnisse (2)

Logik:

- $\neg(\neg p \wedge \neg q)$ ist äquivalent zu $p \vee q$ (De Morgansches Gesetz)
- Ergebnis:

Antwort	Anzahl	Prozent
Richtig	42	68%
Ich hatte keine Logik	9	15%
Ich habe es vergessen	1	2%
Falsch	10	16%

3 Studierende haben die richtige Lösung angegeben und gleichzeitig „Ich hatte das, es ist mir aber entfallen“. 1 Studierender hatte die richtige Lösung und angekreuzt „Ich hatte bisher keine Logik in meinem Studium. Beide Gruppen wurden oben einfach als „Richtig“ gezählt. Entsprechend hatten von den falschen Lösungen jeweils einer die zusätzlichen Kästchen angekreuzt. „Ich hatte keine Logik“ haben nur Wirtschaftsinformatiker und „Sonstige“ angekreuzt.

Auswertung der Vorkenntnisse (3)

UML KClassendiagramme/ER-Diagramme:

- Zeichnen Sie ein UML Klassendiagramm (oder, falls Sie das nicht hatten, ein ER-Diagramm) für folgende Anwendung:
 - Über Professoren ist ein Name und eine EMail-Adresse zu speichern.
 - Entsprechend über Vorlesungen eine Nr und ein Titel.
 - Professoren können beliebig viele Vorlesungen halten (auch 0).
 - Vorlesungen werden von höchstens einem Professor gehalten, es gibt aber auch Vorlesungen, denen noch kein Professor zugeordnet ist.

Auswertung der Vorkenntnisse (4)

UML Klassendiagramme/ER-Diagramme, Forts.:

- Ergebnis:

Antwort	Anzahl	Prozent
UML Klassendiagramm OK	6	10%
UML Klassendiagramm mit Fehlern	22	35%
ER-Diagramm, versch. Varianten	8	13%
Ganz merkwürdig	3	5%
Ich hatte das nicht	14	23%
Ich kann die Aufgabe nicht lösen	9	15%

Auswertung der Vorkenntnisse (5)

UML Klassendiagramme/ER-Diagramme, Forts.:

- UML Klassendiagramm:



- Bezeichnung „hält“ der Beziehung ist nicht nötig.
- Auf der Vorlesungsseite wäre statt „*“ auch „0..*“ möglich.
- Spezifikation von Attributen z.B. auch: `+Name: String`.
- Pfeil an Beziehungslinie ist eher nicht ok.

Auswertung der Vorkenntnisse (6)

SQL:

- Ergebnis:

Antwort	Anzahl	Prozent
Keine Vorkenntnisse	23	37%
Etwas (SELECT FROM WHERE) (dazwischen)	31	50%
Sehr gut (GROUP BY, NOT EXISTS)	2	3%
	6	10%

Als Quelle haben 20 „Schule“ genannt, 7 andere Vorlesungen (insbesondere „Grundlagen der Wirtschaftsinformatik“), 3 eigene Projekte und 1 ein Praktikum in einer Firma.

- Interessant für DB-Entwurf:
Wie geht man mit Kreuzchen zwischen den Kästen um?

Auswertung der Vorkenntnisse (7)

Syntax-Formalismen:

- Ergebnis:

Antwort	Anzahl	Prozent
Kontextfreie Grammatiken (z.B. BNF)	25	40%
Syntax-Diagramme	1	2%
Beides	3	5%
Weder-noch	32	52%
Keine Angabe	1	2%

Studiengänge der Studierenden mit „weder-noch“:

Wirtschaftsinformatik 13, Bioinformatik 6, Informatik 4,

Physik 5, Mathematik 1, Lehramt Informatik (Gym) 1, Sonstige 2.

Grammatiken stehen in der Modulbeschreibung von „Mathematische

Grundlagen der Informatik und Konzepte der Modellierung“ (Logik auch).

Inhalt

- 1 Vorkenntnisse
- 2 Einführung zum Adminer**
- 3 Hausaufgabe
- 4 DBMS-Funktionen
- 5 DB-Managementsysteme
- 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>	<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.
- 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 Vorkenntnisse
- 2 Einführung zum Adminer
- 3 Hausaufgabe**
- 4 DBMS-Funktionen
- 5 DB-Managementsysteme
- 6 Präsenzaufgabe

Hinweise zur ersten Hausaufgabe

- Es gibt ein erstes Hausaufgabenblatt (8 Punkte).
Sie finden es in der Übungsplattform im [\[StudIP-Eintrag der Vorlesung\]](#) (Reiter „Übungsplattform“) und auch auf der Webseite der Vorlesung (im Menü unter [Übung](#)). Das Blatt war noch am Montag (14.10.2024) auf der Webseite (allerdings recht spät).
- **Abgabeschluss ist nächster Montag, 21.10.2024, 18:00.**
- Die Abgabe kann einzeln oder als Zweier-Gruppe erfolgen.
Falls Sie unbedingt eine nicht selbst erstellte Lösung übernehmen wollen, müssen Sie wenigstens die Quelle korrekt angeben.
- Thema des Blattes ist das CSV-Format („comma-separated values“) zum Austausch von Tabellen.
- Außerdem sollen Sie sich mit Informationsquellen zu Datenbanken im Netz beschäftigen.

Hausaufgabe 1: CSV, Link-Liste (1)

- Sie sollen die Nützlichkeit/Qualität von fünf Webseiten bewerten, die auf den Link-Listen der Vorlesung stehen:
 - [<https://users.informatik.uni-halle.de/~brass/db24/links.html>]
 - [<https://.../~brass/db24/software.html>]
- Sie dürfen auch Seiten bewerten, die nicht auf diesen Link-Listen stehen.

Falls Sie die Seiten hoch bewerten, schicken Sie dem Dozenten eine EMail mit dem Vorschlag zur Aufnahme in die Link-Listen.
- Gewünscht ist aber, dass Sie die Seiten der Vorlesung zur Kenntnis nehmen und wenigstens drei der fünf Webseiten aus diesen Link-Listen stammen.

Hausaufgabe 1: CSV, Link-Liste (2)

- „++“: Ich bin ziemlich sicher, dass dies eine nützliche Webseite ist (werde ich voraussichtlich mehrfach anschauen).
- „+“: Nach erstem Eindruck scheint mir dies eine nützliche Seite zu sein (werde ich eventuell nochmal anschauen).
- „o“: Dies könnte vielleicht eine nützliche Seite sein, aber eher nicht für mich.
- „-“: Nach meiner Einschätzung ist die Seite keine nützliche Quelle für Studierende dieser Vorlesung.
- „--“: Ich bin ziemlich sicher, dass diese Seite von der Link-Liste der Vorlesung gestrichen werden sollte.
- „!“: Der Link funktioniert nicht mehr (zählt nicht für 5).

Hausaufgabe 1: CSV, Link-Liste (1)

- Es sollten Daten für diese Tabelle im CSV-Format abgeliefert werden:

WEBSEITEN				
STUD	NR	URI	ART	BEWERTUNG
12345	0	VORRECHNEN:4		
12345	1	http://db-engines.com/	...	+
12345	2	https://www.mysql.com/	DBMS	++

- `webseiten.csv`:

```
12345,0,VORRECHNEN:4,,
12345,1,https://db-engines.com/,Liste,+
12345,2,https://www.mysql.com,DBMS,++
```

Hausaufgabe 1: CSV, Link-Liste (2)

- In der ersten Spalte verwenden Sie bitte eine fünfstellige Zufallszahl.

Natürlich sollen alle Ihre Zeilen die gleiche Zahl in der Spalte `STUD` enthalten. Das ist eine Art Studenten-ID, die aber keine Rückschlüsse auf Ihre Person erlaubt. Wir werden die gesammelten Daten eventuell als Beispiel in der Vorlesung verwenden. Wir hoffen natürlich, dass keine zwei Studierenden die gleiche Zahl verwenden (dann würde beim Laden in die Datenbank ein Fehler auftreten).

- Sie können (und sollen) die Zufallszahl mit folgender SQL-Anfrage berechnen (z.B. im `Adminer`):

```
SELECT CAST(floor(random()*100000) AS INTEGER)
```

Die Funktion `random()` liefert eine Zufallszahl x mit $0 \leq x < 1$.

Die Funktion `floor(x)` rundet ab auf die nächste ganze Zahl, liefert das Ergebnis aber als Gleitkommazahl. Deswegen anschließend die Typumwandlung mit `CAST(x AS T)`.

Hausaufgabe 1: CSV, Link-Liste (3)

CSV-Format:

- Eine Zeile pro Tabellenzeile.
- Datenwerte in einer Zeile werden durch Kommata getrennt.
- Man darf Datenwerte in "... " einschließen.
 - Man muss das tun, wenn ein Datenwert ein Komma, einen Zeilenumbruch oder ein Anführungszeichen enthält.
 - Ein Anführungszeichen im Innern von "... " muss verdoppelt werden. Beispiel: "abc""d" ist der Wert anc"d.
- Beispiel:

```
12345,1,"https://db-engines.com/",Liste,+  
"12345","2","https://www.mysql.com","DBMS","++"
```

Hausaufgabe 1: CSV, Link-Liste (4)

- Beachten Sie bitte, dass viele Varianten des CSV-Formats benutzt werden, z.B. auch mit Semikolon statt Komma.

Sie riskieren 0 Punkte, wenn Sie eine Datei mit einem Syntaxfehler abgeben. Insbesondere sollten Sie, wenn Sie die Datei aus Excel im CSV-Format exportiert haben, sie noch einmal in einem Texteditor anschauen (z.B. Notepad++ [<https://notepad-plus-plus.org/>]).

- Codieren Sie die Zeichen bitte als UTF-8.

Also Nicht-Umlaute wie ASCII (ein Byte), Umlaute dagegen mit zwei Byte. Das CSV-Format sagt leider nichts über die Zeichencodierung aus.

- Verwenden Sie keine Kopfzeile mit den Spaltennamen.

Die Kopfzeile ist im CSV-Format erlaubt, aber nicht vorgeschrieben. Leider kann man nicht an der Datei erkennen, ob sie verwendet wird. Man kann sie mit der ersten Datenzeile verwechseln. Im MIME-Medientyp gibt es einen Parameter dafür.

Hausaufgabe 1: CSV, Link-Liste (5)

- Wir werden versuchen, alle Abgaben mit folgendem `psql`-Befehl in eine entsprechende Tabelle zu laden:

```
\COPY WEBSEITEN FROM 'webseiten.csv'  
      CSV DELIMITER ',' ENCODING 'UTF-8'
```

Eigentlich muss man den Befehl in eine Zeile schreiben. Es ist kein SQL-Befehl, sondern eine `psql`-Erweiterung. SQL wäre ja formatfrei.

- Es ist damit zu rechnen, dass es Punktabzüge gibt, wenn das nicht funktioniert.
- Leider können Sie es mit dem Adminer nicht testen.

Falls Sie PostgreSQL selbst installiert haben, können Sie eine „CREATE TABLE“-Anweisung für die Tabelle unter folgendem Link abrufen:
[\[https://users.informatik.uni-halle.de/~brass/db24/homework/webseiten.sql\]](https://users.informatik.uni-halle.de/~brass/db24/homework/webseiten.sql)

Hausaufgabe 1: CSV, Link-Liste (6)

Literatur zum CSV-Format:

- Der Internet-Standard „RFC 4180“ für den MIME-Typ text/csv:

[<https://www.rfc-editor.org/info/rfc4180>]

- Englischsprachige Wikipedia:

[https://en.wikipedia.org/wiki/Comma-separated_values]

- Deutsche Wikipedia:

[[https://de.wikipedia.org/wiki/CSV_\(Dateiformat\)](https://de.wikipedia.org/wiki/CSV_(Dateiformat))]

- Meine Folien zur Vorlesung „Datenbank-Programmierung“, ab Folie 3-32:

[https://.../~brass/dbp24/slides/p3_updat.pdf]

Inhalt

- 1 Vorkenntnisse
- 2 Einführung zum Adminer
- 3 Hausaufgabe
- 4 DBMS-Funktionen**
- 5 DB-Managementsysteme
- 6 Präsenzaufgabe

Lernziele von Kapitel 2: DBMS-Funktionen

- Die Verwendung eines DBMS zur Verwaltung persistenter Daten mit der (direkten) Verwendung von Dateien vergleichen.
 - Einige Vorteile der Lösung mit DBMS nennen.
Und auch mögliche Nachteile.
 - Vor-/Nachteile für ein konkretes Projekt bewerten.
- „Datenunabhängigkeit“ erklären. Was ist ein Index?
In diesem Kapitel wird hauptsächlich physische Datenunabhängigkeit behandelt.
- Vorteile deklarativer Anfrage/Programmiersprachen benennen.
- Den Begriff der Transaktion erklären, Vorteile der Transaktionsverwaltung im DBMS richtig einschätzen.

Persistente Daten: Wie persistent sind Platten?

- Ein DBMS dient zur Verwaltung persistenter Daten, die Sie dauerhaft speichern wollen.

Speziell strukturierte Daten, nicht einfach Texte oder Bilder.

- Meist werden letztendlich Platten dafür verwendet.
- Hatten Sie schon einmal einen Plattenschaden, bei dem alle Daten auf der Platte nicht mehr lesbar waren?
 - A. Ich selbst hatte eine defekte Platte.
 - B. Jemand aus meiner unmittelbaren Bekanntschaft.
 - C. Nein. Bislam hatten ich und meine Bekannten Glück.
- Und wie ist es mit USB-Sticks?

Aufgabe: Algorithmen im DBMS

Aufgabe:

- Wenn Sie ein DBMS als Software-Bibliothek für Ihr Programm sehen, für welche Aufgaben bekommen Sie Algorithmen in der Bibliothek (die Sie sonst selbst programmieren müssten)?

Tatsächlich ist ein DBMS mehr als eine Unterprogramm-Bibliothek, weil es z.B. auch als Kontrollinstanz die Zugriffsrechte verschiedener Nutzer verwaltet. Auch die deklarative Sprache zur Formulierung von Anfragen und den Anfrageoptimierer können Sie hier außer Acht lassen.

- _____
- _____
- _____
- _____

Indexe

- Ein Index über einer Spalte einer Tabelle hilft u.a., Zeilen mit einem bekannten Wert in dieser Spalte schnell zu finden.
- Es ist also im wesentlichen eine Abbildung von Datenwerten auf physische Adressen von Tabellenzeilen.

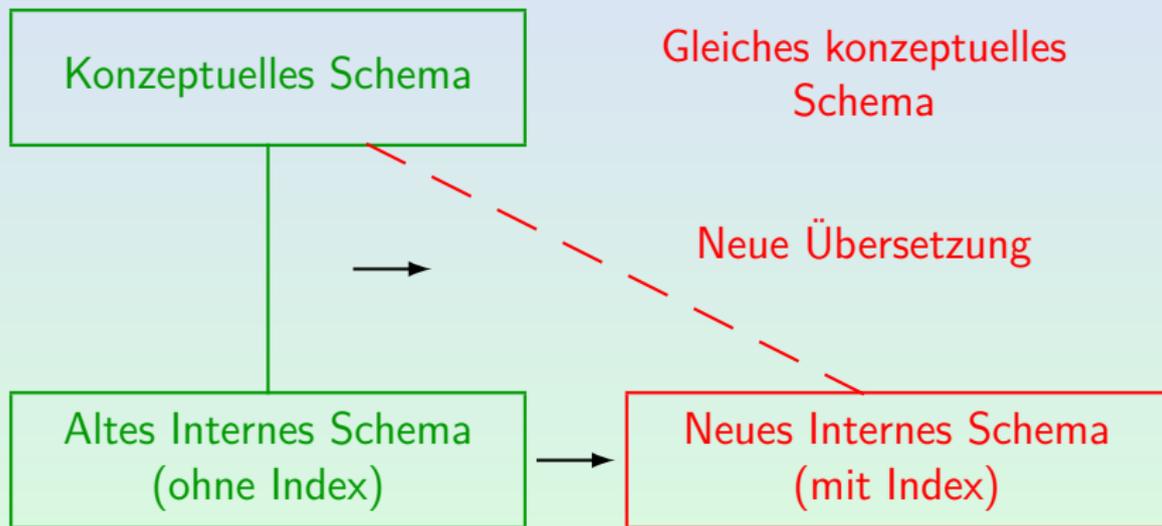
Es kann auch mehr als eine Zeile zu einem Datenwert geben. Diese Abstraktion ist etwas vereinfacht, manche Indexe können auch andere Arten von Anfragen beschleunigen, z.B. mit $<$ und $>$ -Bedingungen.

- Kennen Sie binäre Suchbäume?

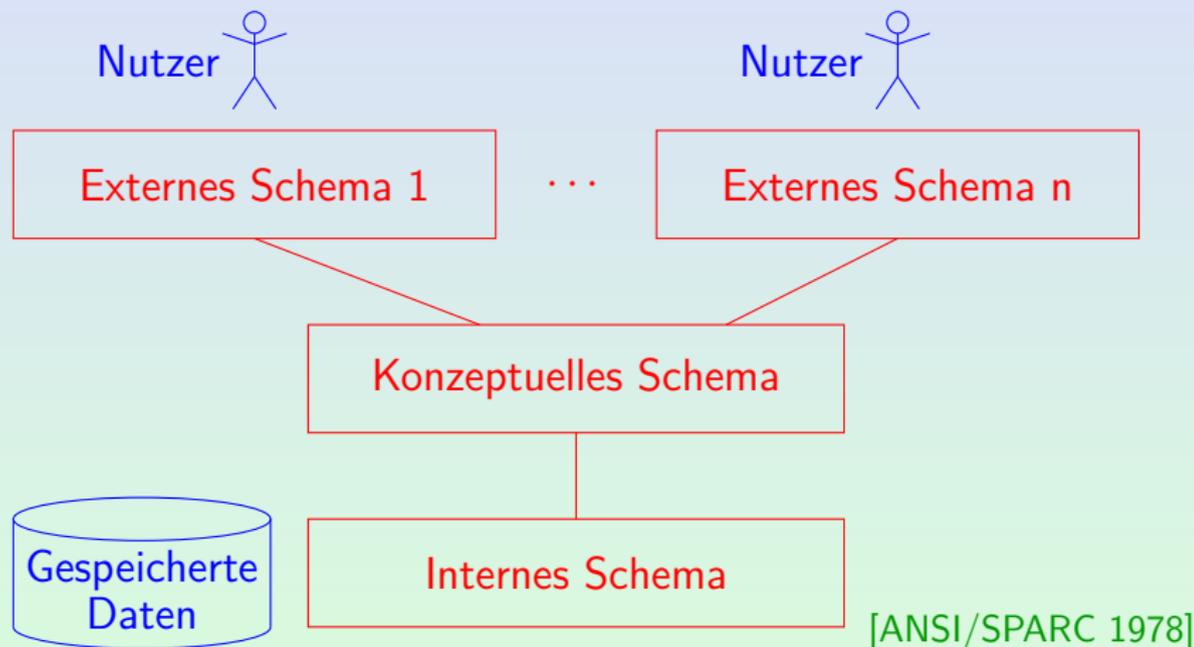
Kennt jemand auch Probleme von binären Suchbäumen?

- Kennen Sie B-Bäume oder B^+ -Bäume oder B^* -Bäume?
- Kennen Sie Hashtabellen?

Physische Datenunabhängigkeit



Drei-Schema Architektur



Beispiel-Datenbank: Konzeptuelles Schema

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

Beispiele zur Drei-Schema-Architektur

Aufgabe:

- Für welche Nutzer könnte es z.B. eigene externe Schemata geben? Wie würden diese aussehen?
- Welche nachträglichen Erweiterungen an den Tabellen könnten Sie sich vorstellen (z.B. eine zusätzliche Spalte)?

Externe Schemata für eine Anwendung würden helfen, dass die Anwendung nach so einer Erweiterung unverändert weiterläuft. Mit etwas Disziplin bei der Formulierung der SQL-Anweisungen ist für zusätzliche Spalten aber noch kein externes Schema nötig.

- Geben Sie ein Beispiel für eine Spalte, bei der das interne Schema einen Index enthalten sollte.

Hinweis zur Drei-Schema-Architektur

- Die Drei-Schema-Architektur der ANSI/SPARC Arbeitsgruppe ist ein theoretisches Modell.
- Wenn man z.B. in PostgreSQL eine relationale Datenbank anlegt, definiert man das konzeptuelle Schema und kann darauf natürlich auch zugreifen.
- Wenn man will, kann man weitere Schemata mit virtuellen Tabellen („Views“, „Sichten“) anlegen, und anderen Nutzern nur auf ein solches Schema Zugriff geben.

Man muss das aber nicht so strukturieren. Die virtuellen Tabellen können auch im gleichen Schema angelegt werden wie die ursprünglichen Tabellen.

- Das interne Schema wird nicht explizit angelegt und nicht ganz strikt getrennt. Z.B. gibt man zusätzliche Speicherparameter beim **CREATE TABLE** Befehl mit an.

DBMS-Funktionen am Beispiel

Aufgabe:

- Angenommen, Sie sollen ein System zur Evaluation der Lehre an dieser Universität entwickeln:
Studierende stimmen über die Vorlesungs-Qualität ab.

Es gibt ein Formular im Internet, in das Studierende ihre Daten eingeben können. Diese werden auf dem Web-Server abgespeichert.

Später werden die gesammelten Daten ausgewertet,
z.B. Durchschnittswerte berechnet.

- Vorschlag: Die Daten werden in einer Datei gespeichert (z.B. im CSV-Format).
- Welche Argumente gibt es, stattdessen ein DBMS zu verwenden?

Nachteile von DBMS

- Was könnten mögliche Nachteile der Verwendung eines DBMS gegenüber einfachen Dateien sein?
 - _____
 - _____
 - _____
- In welchen Situationen lohnt sich die Verwendung eines DBMS eher nicht? D.h. wann kann es seine Vorteile nicht ausspielen?
 - _____
 - _____
 - _____

Deklarative Anfragesprache am Beispiel (1)

- Stellen Sie sich vor, die Hausaufgabenpunkte sind in einer Textdatei gespeichert mit dem Format

Vorname,Nachname,Aufgabennummer,Punkte

(d.h. ein Tupel der Tabelle **ABGABEN** pro Zeile).

- Welchen Aufwand schätzen Sie für die Entwicklung eines Java-Programms, das die Gesamtpunktzahl je Student/-in ausgibt (alphabetisch geordnet)?

Anzahl Zeilen: _____ (ohne Kommentare)

Arbeitszeit: _____

- Trauen Sie sich das zu? (A: ja, B: mit viel Zeit, C: nein)
- In SQL braucht man 4 Zeilen und 2 Minuten Zeit.

Deklarative Anfragesprache am Beispiel (2)

- Wir hatten das letztes Jahr als Bonusaufgabe.
- Die Studierenden mit voller Punktzahl hatten zwischen 20 min und 180 min angegeben, Durchschnittswert war 80 min (Median: 60 min).

Ich habe es auch gelöst und habe 67 min gebraucht, kannte aber schon die Lösungen von Studierenden.

- Die Lösungen hatten zwischen 34 und 181 Zeilen (Durchschnitt: 82) (Muster-Lösung: 78).
- Reine Codezeilen waren zwischen 28 und 139 (Durchschnitt: 64) (Muster-Lösung: 59).
- Man kann die Aufgabe z.B. mit einer **TreeMap** lösen, die Studierenden-Namen auf Punkte abbildet.

SQL zum Vergleich

- Tabelle:

ABGABEN			
VORNAME	NACHNAME	AUFGABE	PUNKTE
Lisa	Weiss	1	10
Michael	Grau	1	9
Daniel	Sommer	1	5
Lisa	Weiss	2	8

- Hier zum Vergleich die Lösung in SQL:

```
SELECT VORNAME, NACHNAME, SUM(PUNKTE)
FROM   ABGABEN
GROUP  BY VORNAME, NACHNAME
ORDER  BY NACHNAME, VORNAME
```

Aggregationsfunktionen wie SUM und die GROUP BY und ORDER BY-Klauseln werden später in dieser Vorlesung ausführlich besprochen.

Inhalt

- 1 Vorkenntnisse
- 2 Einführung zum Adminer
- 3 Hausaufgabe
- 4 DBMS-Funktionen
- 5 DB-Managementsysteme**
- 6 Präsenzaufgabe

Open-Source Datenbanken

- Sie haben schon verwendet (und es wurden in der Vorlesung besprochen):
 - PostgreSQL
 - MySQL: Sehr verbreitet, z.B. in Web Hosting Paketen.
- Es gibt inzwischen viele weitere Open-Source Datenbank-Managementsysteme:
 - MariaDB (von MySQL abgespalten)
 - SQLite: Wird in vielen Programmen verwendet, z.B. Firefox.
[<https://stackoverflow.com/questions/7610896/>]
 - HSQLDB: Einfach in Java-Programme zu integrieren.

Kommerzielles DBMS: Oracle

- Verbreitetes kommerzielles System. Vermutlich Marktführer.

Ich persönlich habe mit Oracle gearbeitet seit meinem Studium.

Es ist das System, was ich am besten kenne. Recht gut standard-kompatibel.

Oracle war die erste kommerzielle SQL-Datenbank überhaupt (1979).

- Die „Express Edition“ kostet nichts:

[\[https://www.oracle.com/database/technologies/appdev/xe.html\]](https://www.oracle.com/database/technologies/appdev/xe.html)

Gibt es für Windows und Linux. Bei Linux funktioniert jedenfalls die Installation unter CentOS problemlos (ähnlich zu Red Hat Linux).

Maximal 4 GB/11 GB Datenbank, 1 Kern, 1 GB RAM, u.a. Einschränkungen.

- Oracle ist auch Anbieter von betriebswirtschaftlicher Standard-Software (Konkurrent zu SAP).

Durch viele Zukäufe, u.a. Siebel Systems (CRM), PeopleSoft, JD Edwards.

Die Datenbank war aber die Basis der Firma. Jetzt auch Cloud Anbieter.

Kommerzielles DBMS: Microsoft SQL Server

- Verbreitetes kommerzielles System, gut standard-kompatibel
- Ursprünglich war es das Sybase-Datenbanksystem, für das Betriebssystem OS/2 (Microsoft/IBM) portiert.

Die erste Version erschien 1989. Die Kooperation mit Sybase lockerte sich ab ca. 1993, inzwischen ist Microsoft SQL Server ein eigenständiges System. Es war lange Zeit nur für Microsoft Betriebssysteme erhältlich, seit 2017 auch für Linux.

- SQL Server 2019 Express kostet nichts.

[<https://www.microsoft.com/de-de/sql-server/sql-server-downloads>]

[<https://docs.microsoft.com/de-de/sql/sql-server/editions-and-components-of-sql-server-version-15>]

Bei der Express-Edition ist die Datenbankgröße begrenzt auf 10 GB, die Puffergröße im Hauptspeicher auf 1.4 GB, maximal 4 Kerne, keine Hoch-Verfügbarkeit (Failover), und andere Einschränkungen.

Kommerzielles DBMS: IBM DB2

- Weiteres wichtiges kommerzielles DBMS.

Edgar F. Codd, der Erfinder des relationalen Datenbankmodells, arbeitete im IBM Forschungslabor in San Jose, als er diese Ideen entwickelt hat. Sein Artikel „A Relational Model of Data for Large Shared Data Banks“ erschien 1970 (in den Communications der ACM). 1981 bekam er den Turing Preis. SEQUEL, eine frühere Version von SQL, wurde von Chamberlin, Boyce et al. 1974 im gleichen Forschungslabor entwickelt. System/R, einer der beiden ersten Forschungs-Prototypen einer relationalen Datenbank, wurde dort 1976/77 entwickelt. Der andere war Ingres von Michael Stonebraker (Berkeley).

- Kostenlose Express-C Version (läuft aus?)

[<https://www-01.ibm.com/marketing/iwm/iwm/web/pickUrxNew.do?source=swg-db2expressc>]

Für Entwickler gibt es diese neueren Versionen, die zum Test kostenlos sind (?):

[<https://www.ibm.com/us-en/marketplace/ibm-db2-direct-and-developer-editions/purchase>]

DB Software: Popularität im Netz

Rang	System	Art	Punkte
1.	Oracle	Relational DBMS	1309
2.	MySQL	Relational DBMS	1023
3.	Microsoft SQL Server	Relational DBMS	802
4.	PostgreSQL	Relational DBMS	652
5.	MongoDB	Document Store	405
6.	Redis	Key-Value Store	150
7.	Snowflake	Relational	141
8.	Elasticsearch	Suchmaschine	132
9.	IBM DB2	Relational DBMS	123
10.	SQLite	Relational DBMS	102

[<https://db-engines.com/de/ranking>] Alle großen nicht nur relational („Multi-Model“).
 Es fließen u.a. ein: Anzahl Treffer in Suchmaschinen (Webseiten), Google Trends (Suchanfragen), Stack Overflow, DBA Stack Exchange, Job-Angebote auf Indeed und Simply Hired, Profile auf LinkedIn und Upwork, die das System nennen, Twitter.

DB Software: Verwendung durch Entwickler

Rang	System	Entwickler
1.	MySQL	48.2%
2.	PostgreSQL	44.0%
3.	SQLite	30.9%
4.	Microsoft SQL Server	29.4%
5.	MongoDB	28.0%
6.	Redis	24.5%
7.	MariaDB	17.1%
8.	Firebase	15.9%
9.	Elasticsearch	15.7%
10.	Oracle	12.9%

Quelle: Stack Overflow Developer Survey 2021 (jeweils „Professional Developers“)

[<https://insights.stackoverflow.com/survey/2021>]

Unter „Programming, Scripting and Markup Languages“ steht SQL (50.7%) an dritter Stelle hinter JavaScript (68.6%) und HTML/CSS (56.0%).

Installations-Erfahrungen (1)

- Sie können in dieser Vorlesung mit dem Adminer-Webinterface arbeiten. Sie müssen keine Software installieren.
 - Außer später den Oracle SQL Developer Data Modeler oder ein anderes Zeichenprogramm (für 1–2 Hausaufgaben zum ER-Modell).
- Allerdings hat man auch Administrator-Rechte und kann mehr ausprobieren, wenn man einen eigenen DB-Server hat.
- Haben Sie PostgreSQL schon auf Ihrem Rechner installiert?
 - A. Ja, direkt.
 - B. Ja, über Docker,
 - C. Habe es versucht, hat aber nicht geklappt.
 - D. Nein.

Installations-Erfahrungen (2)

- Haben Sie eine andere Datenbank auf Ihrem Rechner?
 - A. MySQL oder MariaDB
 - B. Microsoft Access
 - C. Microsoft SQL Server
 - D. Anderes DBMS
 - E. Nein.
- Weiteres Web-Interface (sogar mit verschiedenen DBMS):

[\[http://sqlfiddle.com/\]](http://sqlfiddle.com/)

Links geben Sie CREATE TABLE-Befehle und INSERT-Anweisungen ein, z.B. [\[https://users.informatik.uni-halle.de/~brass/db21/sql/db0.sql\]](https://users.informatik.uni-halle.de/~brass/db21/sql/db0.sql).

Dann klicken Sie auf „Build Schema“. Anschließend geben Sie rechts Ihre Anfrage ein und klicken auf „Run Query“.

Inhalt

- 1 Vorkenntnisse
- 2 Einführung zum Adminer
- 3 Hausaufgabe
- 4 DBMS-Funktionen
- 5 DB-Managementsysteme
- 6 Präsenzaufgabe**

Präsenzaufgabe: DBMS-Funktionen

- Angenommen, Sie entwickeln ein Programm in Java, das persistente Daten braucht (z.B. eine Webseite, in der Nutzer über die Qualität von Restaurants in Halle abstimmen können).
- Nennen Sie drei (möglichst unterschiedliche) Gründe, ein DBMS zur Speicherung der Daten zu verwenden, und nicht eine Datei (z.B. im CSV-Format).
- In welcher Hinsicht wird Ihre Arbeit als Programmierer/in also vereinfacht?
- Die Abgabe erfolgt auf den ausgeteilten Papier-Zetteln.
- Zweier-Gruppen sind erwünscht.

Bei den Hausaufgaben sind Sie dagegen völlig frei, Einzelabgabe oder Zweier-Gruppen zu wählen (was immer für Sie besser ist).