Einführung in Datenbanken

Übung 3: SQL: Lexikalische Syntax und Syntaxdiagramme

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

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]

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)]

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

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

Hausaufgabe 1: Tabelle

```
CREATE TABLE WEBSEITEN(
    STUD NUMERIC(5) NOT NULL,
    NR   NUMERIC(2) NOT NULL,
    URI  VARCHAR(80) NOT NULL,
    ART  VARCHAR(80) NULL,
    BEWERTUNG CHAR(2) NULL,
    CONSTRAINT WEBSEITEN_PK
        PRIMARY KEY(STUD,NR),
    CONSTRAINT WEBSEITEN_STUD_NOT_NEGATIVE
        CHECK(STUD >= 0),
    CONSTRAINT WEBSEITEN_NR_NOT_NEGATIVE
        CHECK(NR >= 0),
    CONSTRAINT WEBSEITEN_BEWERTUNG_GUELTIG
        CHECK(BEWERTUNG IN ('++', '+', 'o',
                            '-', '--', '!'))
);
```

Hausaufgabe 1: Laden der Daten (1)

- Es gab 70 Abgaben (davon 17 Zweier-Gruppen).
- Shellskript (in Verzeichnis `Exercise-1`, das für jede Abgabe ein Verzeichnis enthält, darin wiederum sind die Dateien `webseiten.sql`):

```
for i in */*
do
echo
echo "$i:"
echo
psql -c "\COPY WEBSEITEN FROM '$i' ..."
done
```

Alternative: `ls` in Datei und globales Substitute im Texteditor, um daraus `\COPY`-Befehle zu machen, diese Datei mit `\i` in `psql` einlesen.

Hausaufgabe 1: Laden der Daten (2)

- Von den 70 Dateien konnten ca. 30 geladen werden, alle anderen lieferten Fehler.

  `select count(distinct stud) from webseiten` lieferte 32.

  `select count(*) from webseiten` lieferte 203.

  `select count(*) from webseiten where nr <> 0` lieferte 171.

- Zum Teil war die URL zu lang:

  `value too long for type character varying(80)`

- Das war mein Fehler. Also `DROP TABLE WEBSEITEN` und verbessertes `CREATE TABLE` mit 120 Zeichen.

  `select count(distinct stud) from webseiten` lieferte 37.

  `select count(*) from webseiten` lieferte 238.

  `select count(*) from webseiten where nr <> 0` lieferte 201.

Hausaufgabe 1: Laden der Daten (3)

- Es haben Studierende eine Kopfzeile mit den Spaltennamen (`STUD,NR,URI,ART,BEWERTUNG`) verwendet:

  `invalid input syntax for type numeric`

  Der Spaltenname `STUD` wird als Datenwert für die erste Spalte interpretiert, die hat aber einen numerischen Datentyp. „Sender“ und „Empfänger“ einer CSV-Datei müssen sich darüber einig sein, ob es eine Kopfzeile geben soll oder nicht. Der Datei selbst kann man das nicht ansehen.

- Eine Leerzeile am Ende führt zu:

  `missing data for column "nr"`

  Man kann sich fragen, warum gibt es den Fehler für die zweite Spalte, aber nicht die erste Spalte? Für die erste Spalte wurde zunächst der leere String als Nullwert interpretiert, und der Fehler wird bemerkt, weil das Komma fehlt, hinter dem der zweite Wert kommen würde. Wäre später der Nullwert in die Tabelle eingefügt worden, hätte das natürlich auch zu einem Fehler geführt, da die Spalte als `NOT NULL` deklariert ist. Aber der Import bricht beim ersten Fehler ab.

Hausaufgabe 1: Laden der Daten (4)

- Auch für leere Spalten muss man das Komma angeben:

  ```
  68205,0,VORRECHNEN:2 ZEIT:45
  ```

  Das führt zu:

  ```
  missing data for column "art"
  ```

- Hier fehlt ein Komma (bei einer Tabelle mit n Spalten muss jede Zeile genau $n-1$ Kommata enthalten):

  ```
  36400,4,https://solhsa.com/g3/,-
  ```

  Ergebnis:

  ```
  missing data for column "bewertung"
  ```

- Wenn man die Spalte mit der Nummer vergisst, bekommt man:

  ```
  invalid input syntax for type numeric:
  " http://sql-island.informatik.uni-kl.de/"
  ```

Hausaufgabe 1: Laden der Daten (5)

- Ein Leerzeichen nach dem Komma zählt mit zum nächsten Datenwert:

  ```
  value too long for type character(2): " ++"
  ```

- Hier ist ein (unsichtbares) Leerzeichen am Ende der Zeile (nach dem letzten Komma):

  ```
  35510,0,VORRECHNEN2:2,,
  ```

  Das führt zu:

  ```
  new row for relation "webseiten" violates
  check constraint "webseiten_bewertung_gueltig"
  ```

  Der leere String wird beim Laden als Nullwert verstanden, der in der Spalte zulässig ist. Ein String aus einem Leerzeichen ist dagegen nicht zulässig. Ebenso wird auch `""` als Nicht-Nullwert verstanden (das ist der leere String), auch hier gibt es einen Fehler.

Hausaufgabe 1: Laden der Daten (6)

- Der Standard (RFC 4180) sagt klar: „Each field may or may not be enclosed in double quotes. [...] If fields are not enclosed with double quotes, then double quotes may not appear inside the fields.“
- Also ist ein Leerzeichen nach dem Komma und dann ein Wert `"..."` illegal.

  Es scheint aber so, dass PostgreSQL das Leerzeichen einfach dem Datenwert in `"..."` hinzufügt.

- Die folgende Zeile ist tatsächlich nur ein Wert:

  ```
  "64533,""0"",""VORRECHNEN:0"""
  ```

  Das führt zu:

  ```
  invalid input syntax for type numeric
  ```

Hausaufgabe 1: Laden der Daten (7)

- Falsche Anführungszeichen:
  `invalid input syntax for type numeric: ""“86150“""`
- Ein „Byte Order Mark“ am Anfang der Datei führt zu:
  `invalid input syntax for type numeric: "<feff>28366"`
- `'Null'` wird nicht als Nullwert interpretiert.
- Hier steht ein Punkt statt einem Komma:
  `71069.4,https://datageek.blog/2014/01/28/...`
- Die Zahl `0` geht nicht als Bewertung (`o` gemeint?).
- Ein Student hat das `CREATE TABLE` und `INSERT`-Statements in die CSV-Datei geschrieben.

Hausaufgabe 1: Auswertung der Daten (1)

- 11 Nennungen: [https://www.w3schools.com/sql/]

  ++: 8, +: 3

- 8: [https://sql-island.informatik.uni-kl.de/]

  ++: 3, +: 4, -: 1

- 7: [http://infolab.stanford.edu/~ullman/fcdb/aut07/]

  ++: 1, +: 1, o: 3, -: 1, --: 1

- 7: [https://sqlzoo.net/]

  ++: 4, +: 3

- 6: [https://www.postgresql.org/docs/9.4/sql-commands.html]

  ++: 4, +: 2

Hausaufgabe 1: Auswertung der Daten (2)

- 5: [https://www.dcs.warwick.ac.uk/~hugh/CS253/CS253-TheAskewWall.pdf]

  ++: 1, +: 2, o: 2 (SQL-Kritik, ziemlich speziell)

- 5: [https://sqlzoo.net/wiki/SQL_Tutorial]

  ++: 3, +: 2

- 5: [http://sql-island.informatik.uni-kl.de/]

  ++: 2, +: 1, o: 2

- 5: [https://www.postgresql.org/docs/9.5/datatype.html]

  ++: 4, +: 1

Hausaufgabe 1: Auswertung der Daten (3)

- Webseiten absteigend nach Anzahl Bewertungen sortiert:

```
SELECT   uri, COUNT(*) AS anzahl
FROM     webseiten
WHERE    nr <> 0
GROUP BY uri
ORDER BY anzahl DESC
```

- Bewertungen für häufig genannte Webseiten:

```
SELECT   uri, bewertung, count(*)
FROM     webseiten
WHERE    uri IN (SELECT   uri
                 FROM     webseiten
                 GROUP BY uri
                 HAVING   count(*) >= 5)
GROUP BY uri, bewertung
ORDER BY uri;
```

Inhalt

Präsenzaufgabe: DBMS-Funktionen (1)

- 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).

Präsenzaufgabe: DBMS-Funktionen (2)

Beispiele für Antwortmöglichkeiten:

- Algorithmen, die in das DBMS eingebaut sind, und die man sonst selbst programmieren müsste (z.B. Sortieren, Map-Datenstrukturen).

  Wenn man die durchschnittliche Bewertung eines Restaurants bestimmen will, muss man die abgegebenen Bewertungen nach dem Restaurant-Namen gruppieren (Bewertungen für das gleiche Restaurant zusammenfassen). Das geht z.B., indem man alle Bewertungen nach dem Restaurant sortiert. Man kann aber auch z.B. eine Hashtabelle machen, mit dem Restaurant-Namen als Suchschlüssel, in der man die bisherige Summe und Anzahl der Bewertungen abspeichert (so dass man die durchschnittliche Bewertung inkrementell bei einem Durchlauf durch alle Bewertungen berechnen kann).

Präsenzaufgabe: DBMS-Funktionen (3)

Beispiele für Antwortmöglichkeiten, Forts.:

- Sperren bei parallelen Einfügungen

  Es ist ganz typisch, dass der Webserver aus einer ganzen Reihe paralleler Prozesse besteht (damit irgendwelche Verzögerungen bei einzelnen HTTP-Requests nicht dazu führen, dass andere Nutzer, die die Webseite aufrufen, warten müssen). Es gibt damit auch parallele Aufrufe des Programms, dass für die Speicherung neuer Restaurant-Bewertungen zuständig ist. Ohne Sperren würde eine CSV-Datei zerstört werden (ein Stück einer Zeile von Prozess 1, dann eine Zeile von Prozess 2, dann der Rest der Zeile von prozess 1). Man würde also die ganze Datei sperren. Andere Sperr-Granularitäten sind für Dateien normalerweise nicht vorgesehen. Bei einer Datenbank sind aber echt parallele Einfügungen möglich (z.B. werden nur kurz einzelne Blöcke der DB-Datei gesperrt, und Einfügungen auf mehrere Blöcke verteilt).

Präsenzaufgabe: DBMS-Funktionen (4)

Beispiele für Antwortmöglichkeiten, Forts.:

- Zugriffsrechte

  Es wäre z.B. möglich, dem Datenbank-Account, den die Webschnittstelle verwendet, nur Lese- und Einfüge-Rechte für die Bewertungs-Tabelle zu geben. Dann könnten auch bei einem Bug oder Hacker-Angriff über die Webschnittstelle keine Bewertungen verändert oder gelöscht werden. Auf der Ebene der Betriessystem-Dateien muss man Schreibrechte für die Datei geben, wenn man Bewertungen einfügen will, und das sind dann beliebige Möglichkeiten zum Verändern der Datei.

- Adhoc-Anfragen

  Wenn man eine neue Idee für eine Auswertung der Daten hat, kann man das sehr schnell umsetzen (z.B. Bewertungen nach dem Weglassen von Ausreißern). Wenn man alle Anfragen als Programm schreiben muss, dauert das deutlich länger.

Inhalt

Datenbank: Komponisten

- Die Aufgaben dieses Übungsblattes beziehen sich auf die Komponisten-Tabelle in einer Datenbank mit Informationen über Musik-CDs (klassische Musik).

KOMPONIST				
KNr	Name	Vorname	geboren	gestorben
11	Händel	Georg Friedrich	1685	1759
⋮	⋮	⋮	⋮	⋮

- Sie finden Sie im Adminer, Schema „komponist_public“:

  https://dbs.informatik.uni-halle.de/edb?pgsql=db&username=student_gast&db=postgres&ns=komponist_public

Übungsblatt 2, Aufgabe 1

- Schreiben Sie eine SQL-Anfrage, die den Vornamen des Komponisten „Vivaldi“ bestimmt.
- Versuchen Sie, sich dabei an die eingeschränkte Syntax aus Kapitel 3 der Vorlesung zu halten.
- Das erwartete Ergebnis ist:

vorname
Antonio

- Schreiben Sie Ihre Bereitschaft, vorzurechnen, in einen Kommentar (ebenso eventuelle weitere Angaben zu Quellen und Arbeitszeit).

  Da die Fähigkeit, Kommentare in SQL zu schreiben, wichtig ist, wird ein Punkt abgezogen, wenn die Angabe fehlt. Dies gilt auch für die anderen Aufgaben.

Übungsblatt 2, Aufgabe 2

- Schreiben Sie eine SQL-Anfrage, die die Lebensdaten von Wolfgang Amadeus Mozart liefert.
- Dabei ist „Wolfgang Amadeus“ der Vorname und „Mozart“ der Nachname (Spalte „`Name`“).
- Das erwartete Ergebnis ist:

geboren	gestorben
1756	1791

Übungsblatt 2, Aufgabe 3 (1)

- Geben Sie alle Komponisten aus, die zwischen 1500 und 1599 geboren wurden (jeweils einschließlich der Grenzen). Drucken Sie alle Spalten der Tabelle.
- Das erwartete Ergebnis ist (die Sortierung könnte anders sein):

knr	name	vorname	geboren	gestorben
13	Monteverdi	Claudio	1567	1643
19	Byrd	William	1543	1623
42	Franck	Melchior	1580	1639
44	Mainerio	Giorgio	1545	1582
47	Da Nola	Giovan ...	1510	1592
48	Azzaiolo	Filippo	1530	1569
49	Susato	Tilman	1500	1561

Übungsblatt 2, Aufgabe 3 (2)

- Solange Sie keine bestimmte Sortierung in Ihrer SQL-Anfrage verlangen, darf das DBMS die Ergebniszeilen in irgendeiner Reihenfolge ausgeben (mathematisch: Menge/Multimenge).

  Das hängt von dem Auswertungsplan ab, für den sich der Optimierer entschieden hat. Bei der Beispiel-Ausgabe entspricht die Komponisten-Nummer `knr` wohl der Speicherreihenfolge, und der Optimierer hat einen einfachen „Full Table Scan“ gewählt. Das kann in einer anderen PostgreSQL-Version oder bei einer größeren Tabelle aber anders sein.

- Wenn Sie wollen, können Sie

  ```
  ORDER BY geboren
  ```

  an die Anfrage anhängen, um eine Sortierung nach dem Geburtsjahr zu bekommen.

  Wenn Sie für den Vergleich mit der erwarteten Ausgabe eine Sortierung nach der `knr` erzwingen wollen, schreiben Sie entsprechend „`ORDER BY knr`“.

Übungsblatt 2, Aufgabe 4

- Ein Studienkollege hat folgende Anfrage geschrieben:

```
SelecT GEBOREN
FROM   komponist
WHERE  vorname = "Peter", geboren > 1800.
```

- Leider enthält sie Syntaxfehler.
- Versuchen Sie zuerst, die Fehler zu finden, ohne die Anfrage auszuführen.
- Anschließend führen Sie die fehlerhafte Anfrage aus.

  Es ist auch nützlich, sich mit den Reaktionen von PostgreSQL auf Syntaxfehler vertraut zu machen.

- Dann korrigieren Sie die Anfrage bitte.

  Geben Sie die korrigierte Anfrage mit Kommentaren zu den Syntaxfehlern ab.

  Achten Sie darauf, nur echte Syntaxfehler zu nennen, keine Stilfehler.

Übungsblatt 2, Aufgabe 5

- Welchen Wert gibt folgende Anfrage aus?

```
SELECT char_length('\r\n''abc')
```

- Die Funktion `char_length` liefert die Länge einer Zeichenkette in Zeichen.

  Es gibt auch `octet_length` für die Länge in Bytes).

- Sie finden diese Funktion in der PostgreSQL-Dokumentation:
  [https://www.postgresql.org/docs/current/functions-string.html]
- Überlegen Sie zuerst selbst, bevor Sie die Anfrage ausführen.
- Geben Sie eine kurze Erklärung des Ergebnisses ab, und zwar als `.txt`-Datei. Bitte kein Word, und auch kein PDF.

Allgemeine Hinweise (1)

- Anfragen, die in PostgreSQL nicht ausführbar sind (wegen Syntaxfehlern) werden automatisch mit 0 Punkten bewertet!

  Eventuelle Anmerkungen müssen als SQL-Kommentar geschrieben werden.

- Abweichungen vom erwarteten Ergebnis im Beispielzustand führen dagegen nicht automatisch zu 0 Punkten (aber schon zu Punktabzug).

  Sie müssen keine besonderen Tricks machen, um die oben gezeigten Ausgaben zu erreichen (das wäre eine Art von „mogeln“). Anfragen, die zwar im Beispielzustand des gewünschte Ergebnis ausgeben, aber mit der Aufgabenstellung kaum etwas zu tun haben, können auch mit 0 Punkten bewertet werden.

Allgemeine Hinweise (2)

- Bei Unklarheiten und Problemen können Sie Fragen im Forum in StudIP stellen.

  Bitte posten Sie aber keine Anfragen, die der Lösung nahe kommen. Bei komischen Syntaxfehlern, die Sie sich nicht erklären können, können Sie probieren, dem Dozenten oder dem Übungsleiter eine persönliche EMail mit der Anfrage zu schicken.

- Sie haben natürlich auch die Möglichkeit,
  - in der Übung zu fragen,
  - oder im Anschluss an die Vorlesung,
  - oder im Tutorium jeweils am Montag 12^{15} im Multimedia-Pool (Raum 3.02). Der Dozent ist anwesend so lange wie es Fragen gibt.

Allgemeine Hinweise (3)

- Schreiben Sie Ihren Namen (sowie Matrikelnummer etc.) bitte nicht in die Abgaben oder Dateinamen, da wir die Anfragen auch für ein Forschungsprojekt verwenden wollen.

  Das soll natürlich in anonymisierter Form geschehen, aber wir können nicht garantieren, dass wir identifizierbare Informationen auch im Text der Anfrage in allen Fällen finden und eliminieren.

- Vergessen Sie nicht die Angaben zum Vorrechnen (natürlich als SQL-Kommentar).

- Eventuell verwendete Quellen wie ChatGPT (oder auch die Lösung eines anderen Studierenden) müssen angegeben werden (mit der in Kapitel 0, Folien 22 bis 27 beschriebenen Codierung).

  Sie sind erwachsene Menschen und können sich ja denken, dass Sie nur durch eigene Beschäftigung mit den Aufgaben lernen.

Inhalt

Lexikalische Syntax von SQL (1)

- Was ist `"abc"`?
  A. Eine Zeichenketten-Konstante (String).
  B. Ein Bezeichner (z.B. Tabellen- oder Spaltenname).
  C. Ein Syntaxfehler.
- Muss in SQL das schließende `'` einer String-Konstante auf der gleichen Zeile wie das öffnende stehen?
  A. Ja, wenn Sie nicht auf der gleichen Zeile geschlossen wird, ist das ein Syntaxfehler
  B. Nein, es ist möglich, dass sich Zeichenketten über mehrere Zeilen erstrecken.

Lexikalische Syntax von SQL (2)

- Was muss man tun, wenn man ein Apostroph-Zeichen `'` in einer Zeichenkette braucht?
  Beispiel `a'b`:

  A. Nichts, `'ab'c'` funktioniert.
  B. Die Zeichenkette in `"` einschließen: `"a'b"`.
  C. Verdoppeln: `'a''b'`
  D. Ein Escape-Zeichen voranstellen: `'a\'b'`
  E. Eine Entity-Referenz verwenden: `'a&apos;b'`

- Ist `_a` ein legaler Bezeichner in SQL?

  Eine andere Frage ist, ob PostgreSQL es akzeptiert: Adminer.
  Mit welcher Anfrage können Sie es testen?

Lexikalische Syntax von SQL (3)

- Was wird das Ergebnis von `'2' < '13'` sein?
  A. true
  B. false
  C. Syntaxfehler
- Wie schreibt man den Test auf Ungleichheit in SQL (nach dem SQL-Standard)?
  A. `!=`
  B. `<>`
  C. `\=`
  D. `^=`

Lexikalische Syntax von SQL (4)

- Wie schreibt man Kommentare in SQL (nach dem SQL-Standard)?

  A. `/* ... */`

  B. `// ...`

  C. `-- ...`

  D. `(: ... :)`

  E. `% ...`

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

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. Bislang 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 Unterprogrammbibliothek, 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.

- _____
- _____
- _____
- _____

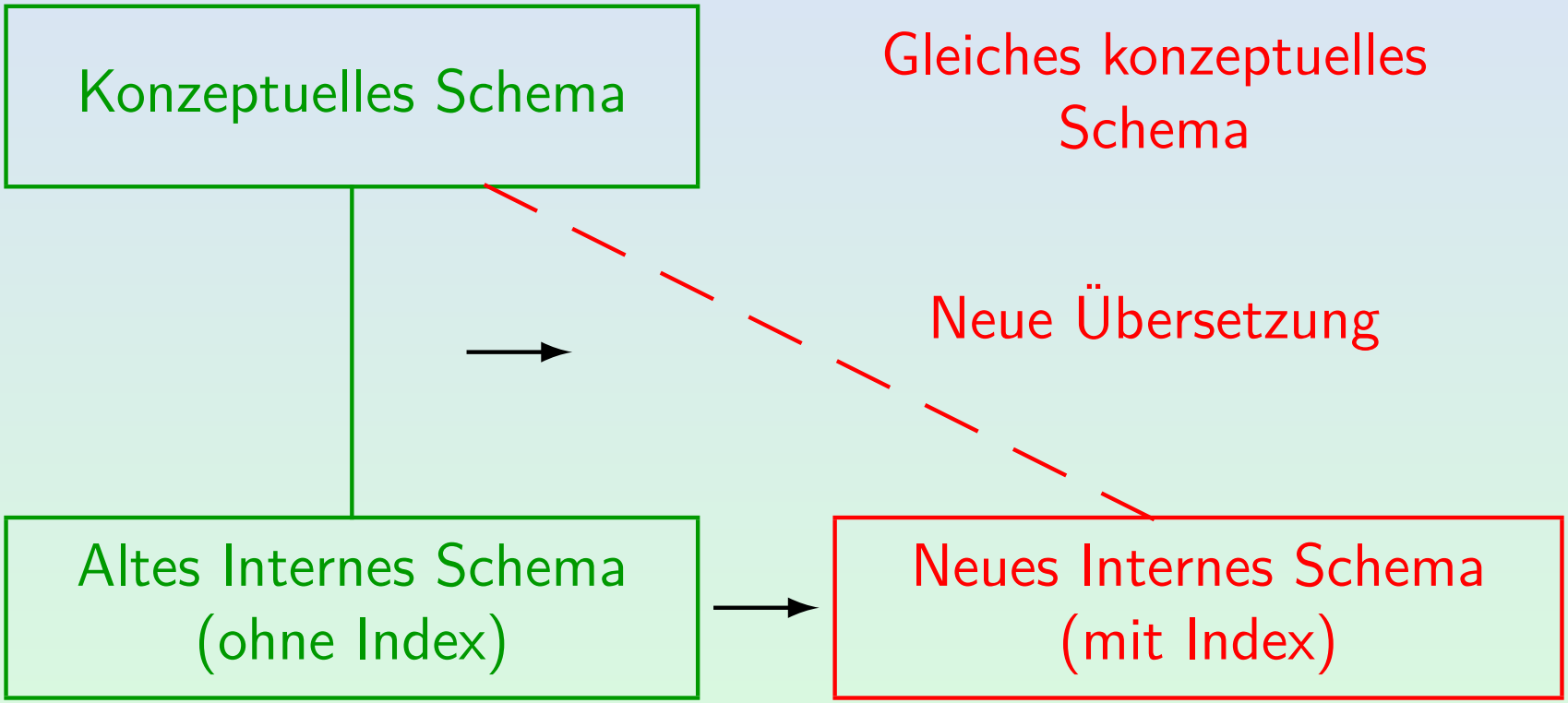
Indexe

- 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.

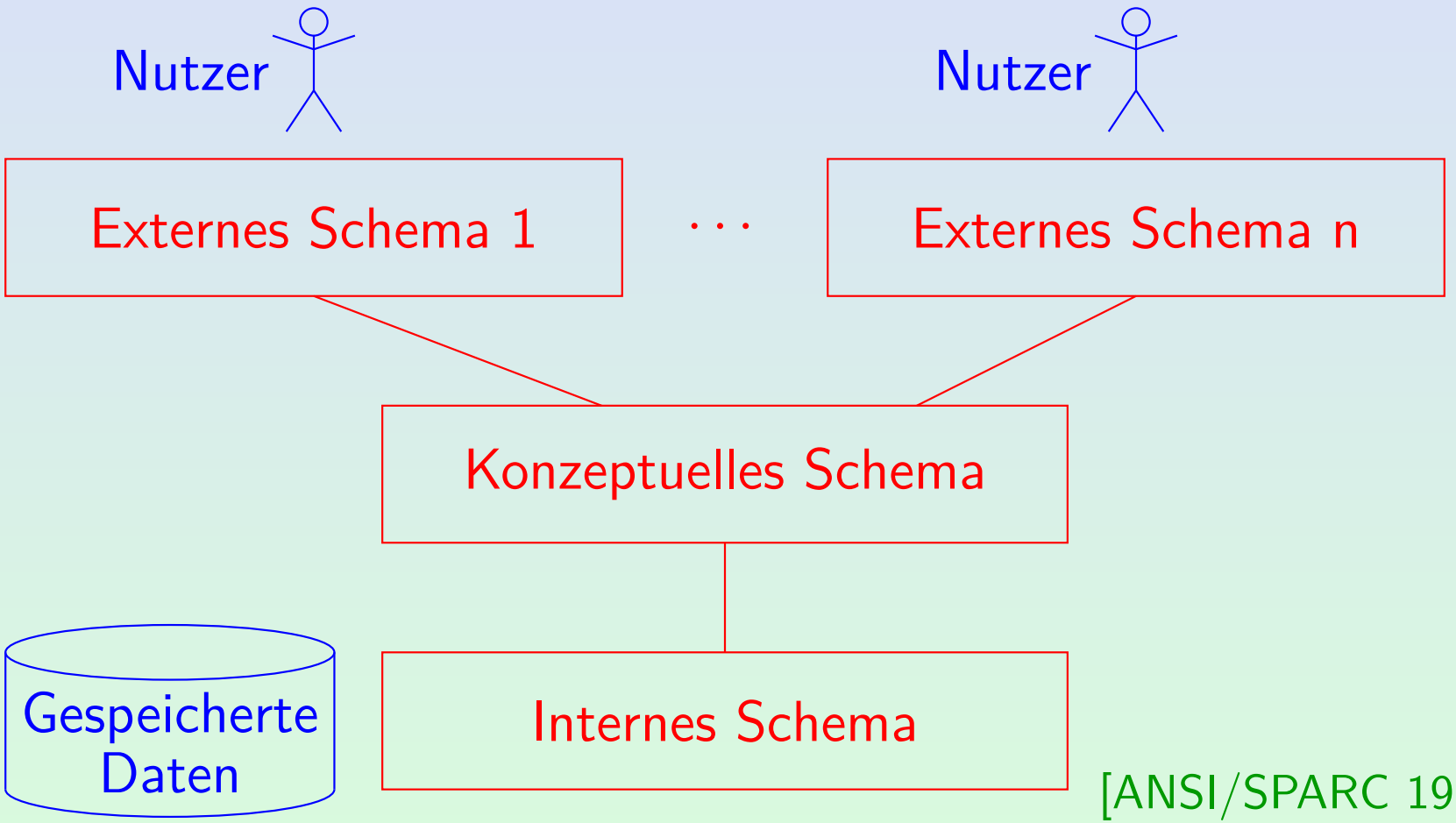
  Es kann auch mehr als eine Zeile zu einem Datenwert geben. Diese Abstraktion ist etwas vereinfacht, manche Indexe können auch andere Arten von Anfragen beschleunigen, z.B. mit < und >-Bedingungen.
- Kennen Sie binäre Suchbäume?

  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



[ANSI/SPARC 1978]

Beispiel-Datenbank: Konzeptuelles Schema

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

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.

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

Präsenzaufgabe: Syntaxgraph

- Zeichnen Sie ein Syntaxdiagramm in der Notation der Vorlesung für Zeichenketten der folgenden Form:
  - Am Anfang steht immer der Buchstabe „a“.
  - Dann kommen beliebig viele „b“ (aber mindestens eins).
  - Anschließend entweder ein „c“ oder ein „d“.
  - Am Ende optional ein „e“ (d.h. das „e“ darf auch fehlen).
- Das Syntaxdiagramm soll die syntaktische Kategorie „Test“ beschreiben. Ein mögliches Wort der definierten Sprache wäre „abbbce“.