

Einführung in Datenbanken

Übung 2: DBMS-Funktionen und relationales Modell

Prof. Dr. Stefan Brass

PD Dr. Alexander Hinneburg

Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2020/21

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

Inhalt

- 1 Präsenzaufgabe von letzter Woche
- 2 DBMS-Funktionen
- 3 Relationales Modell
- 4 Präsenzaufgabe

Präsenzaufgabe von letzter Woche (1)

DBMS			
DBMS_NAME	ANZ	MIN_JAHR	MAX_JAHR
MySQL	12	1994	1995
Oracle Database	11	1977	1979
MongoDB	9	2007	2019
Microsoft Access	8	1980	1992
MariaDB	7	2009	2009
PostgreSQL	5	1986	1996
IBM DB2	4	1970	1993
SQLite	4	1995	2000
Microsoft SQL Server	4	1989	1989
Neo4j	2	2007	2010
SAP HANA	2	2010	2010

Präsenzaufgabe von letzter Woche (2)

- Weitere Nennungen (nur ein Mal):
 - Amazon DynamoDB
 - Apache Cassandra
 - Papyrus Autor
 - Ist eine Office-Software (mir nicht bekannt), die auch Datenbank-Funktionalität enthält (vermutlich als Ersatz für Access). Die Links beziehen sich aber auf das Eclipse-Plugin „Papyrus“, das ein UML Modellierungswerkzeug ist.
- Schwierigkeit u.a. unterschiedliche Namen für das gleiche DBMS:
 - „Oracle“ vs. „Oracle Database“, „Oracel“
 - „Microsoft SQL“, „Microsoft SQL Server“, „Microsoft SQL Server 2019“.

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. Bisher hatten ich und meine Bekanntschaft 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.

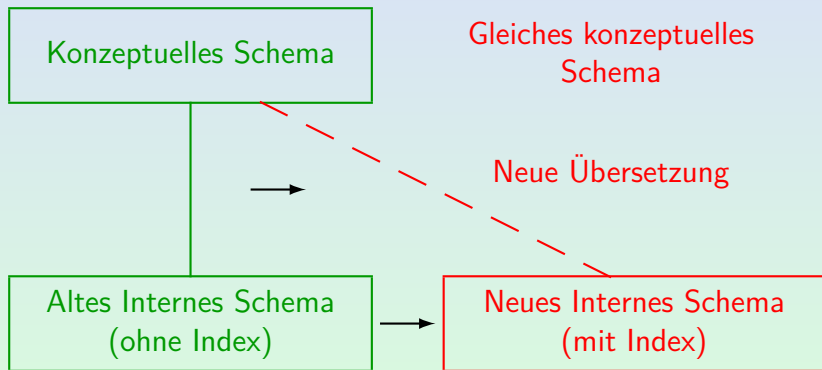
Es kann auch mehr als eine Zeile zu einem Datenwert geben. Diese Abstraktion ist etwas vereinfacht, manche Indexe können auch andere Arten von Anfragen beschleunigen, z.B. mit < und >-Bedingungen.

- Kennen Sie binäre Suchbäume?

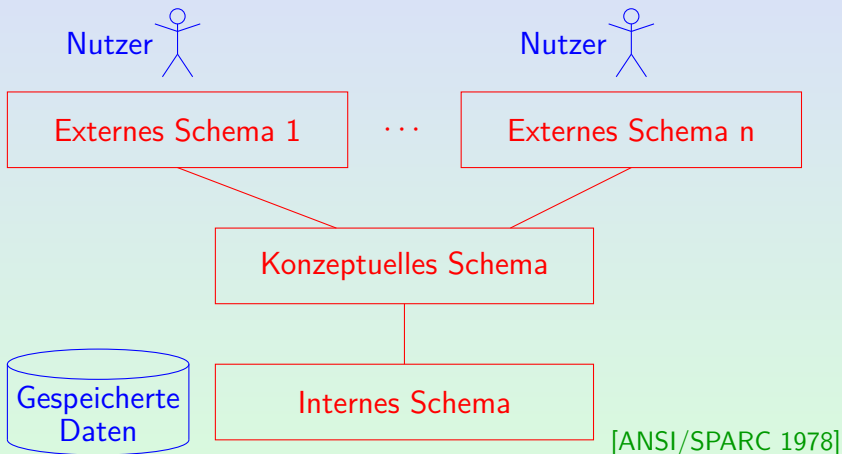
Kennt jemand auch Probleme von binären Suchbäumen?

- 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>	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

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?

Deklarative Anfragesprache am Beispiel

- 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)

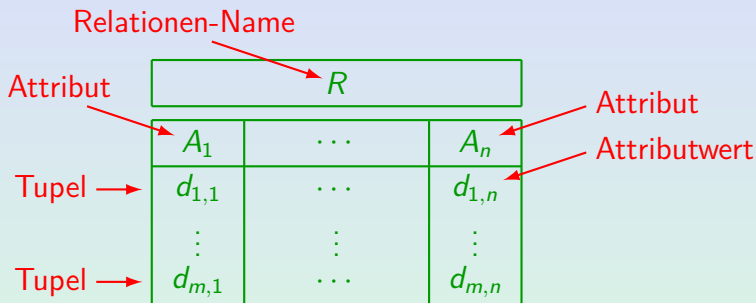
Arbeitszeit: _____

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

Lernziele von Kapitel 3: Relationales Modell

- Begriffe „Tupel“, „Attribut“, „Relation“ erklären.
- Die in dieser Vorlesung verwendete Kurz-Notation für relationale DB-Schemata $R(\underline{A}, B)$ lesen und schreiben.
- Nullwerte erklären. Vor-/Nachteile nennen.
- Das Konzept eines Schlüssels erklären.
Gegebene Tabellen auf Einhaltung von Schlüsseln prüfen.
Sinnvolle Schlüssel für eine Anwendung wählen.
- Das Konzept eines Fremdschlüssels erklären.
Die Fremdschlüssel-Bedingung für einen Zustand prüfen.
- Einfache `CREATE TABLE` Anweisungen schreiben.

Grundbegriffe relationaler Datenbanken



Synonyme: Relation und Tabelle.

Tupel, Zeile und Record (**NICHT** „Reihe“!).

Attribut, Spalte, Feld.

Attributwert, Spaltenwert, Tabelleneintrag.

Nullwerte

- Ein Nullwert wird u.a. für nicht existierende oder unbekannte Werte verwendet (Tabelleneintrag „leer“).

- Z.B. kann EMAIL in STUDENTEN Null sein:

```
CREATE TABLE STUDENTEN(  
    SID           NUMERIC(3)  NOT NULL,  
    VORNAME      VARCHAR(20) NOT NULL,  
    NACHNAME     VARCHAR(20) NOT NULL,  
    EMAIL        VARCHAR(80) )
```

- In der vereinfachten Notation der Vorlesung werden umgekehrt optionale Attribute markiert:

```
STUDENTEN(SID, VORNAME, NACHNAME, EMAILo)
```

- Alternative (wenn nur ASCII Zeichen möglich): „EMAIL?“.

Schlüssel

- Ein Schlüssel einer Relation R ist eine Spalte A oder Spalten-Kombination A_1, \dots, A_k , die die Zeilen in R eindeutig identifiziert (keine zwei Zeilen mit gleichem Schlüsselwert).
- Wenn z.B. SID als Schlüssel von $STUDENTEN$ deklariert wurde, ist dieser DB-Zustand verboten:

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

- Schlüssel sind eine Art von Integritätsbedingungen.
Bedingungen, die jeder Datenbank-Zustand erfüllen muss.

Schlüssel: Notation (1)

- Die Primärschlüssel-Attribute werden oft markiert, indem man sie unterstreicht: $R(\underline{A_1}, \dots, \underline{A_k}, A_{k+1}, \dots, A_n)$.

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

- Weitere Schlüssel muss man als Text dazu schreiben.
- Wenn man ASCII schreiben muss: # vor dem Attribut statt unterstreichen: `AUFGABEN(#ATYP, #ANR, THEMA, MAXPT)`

Schlüssel: Notation (2)

- In SQL können Schlüssel folgendermaßen definiert werden:

```
CREATE TABLE STUDENTEN(  
    SID            NUMERIC(3)  NOT NULL,  
    VORNAME       VARCHAR(20) NOT NULL,  
    NACHNAME      VARCHAR(20) NOT NULL,  
    EMAIL         VARCHAR(80),  
    PRIMARY KEY(SID),  
    UNIQUE(VORNAME, NACHNAME))
```

Die genaue Syntax der CREATE TABLE Anweisung und mehr Details zur Integritätsüberwachung sind ein Kapitel der Fortsetzungs-Vorlesung „Datenbank-Programmierung“.

Beispiel zu Schlüsseln

- Was könnten hier Schlüssel sein?

R				
A	B	C	D	E
1	10	100	300	4
1	20	100	300	5
1	30	200	400	5

- Aber Vorsicht: Ein möglicher Zustand kann nur zeigen, was nicht Schlüssel sein darf.

Man kann sich inspirieren lassen, muss aber noch prüfen, ob der Schlüssel von der Anwendung her Sinn macht (darüber ist hier nichts bekannt) und tatsächlich für alle Zustände gelten soll.

- Beachten Sie, dass interessante Schlüssel minimal bezüglich „ \subseteq “ sein müssen.

Fremdschlüssel

SID in **BEWERTUNGEN** ist ein Fremdschlüssel, der **STUDENTEN** referenziert:

STUDENTEN				BEWERTUNGEN			
<u>SID</u>	VORNAME	NACHNAME	...	<u>SID</u>	<u>ATYP</u>	<u>ANR</u>	PUNKTE
101	Lisa	Weiss	...	101	H	1	10
102	Michael	Grau	...	101	H	2	8
103	Daniel	Sommer	...	102	H	1	9
104	Iris	Winter	...	102	H	2	9
				103	H	1	5
				105	H	1	7

Fehler ?

Die hier benötigte Bedingung ist, dass jeder **SID**-Wert in **BEWERTUNGEN** auch in **STUDENTEN** auftaucht.

Fremdschlüssel: Notation (1)

- In der Attributlisten-Notation können Fremdschlüssel durch einen Pfeil und den Namen der referenzierten Tabelle markiert werden. Bei zusammengesetzten Fremdschlüsseln braucht man Klammern:

```
BEWERTUNGEN(SID → STUDENTEN,  
              (ATYP, ANR) → AUFGABEN, PUNKTE)  
STUDENTEN(SID, VORNAME, NACHNAME, EMAIL)  
AUFGABEN(ATYP, ANR, THEMA, MAXPT)
```

- Da normalerweise nur Primärschlüssel referenziert werden, ist es nicht nötig, die zugehörigen Attribute der referenzierten Tabelle anzugeben.
- Im Beispiel sind die Fremdschlüsselattribute auch Teil des Schlüssels. Das muss nicht so sein.

Fremdschlüssel: Notation (2)

- In SQL können Fremdschlüssel wie folgt deklariert werden:

```
CREATE TABLE BEWERTUNGEN(  
    SID      NUMERIC(3)    NOT NULL,  
    ATYP     CHAR(1)      NOT NULL,  
    ANR      NUMERIC(2)    NOT NULL,  
    PUNKTE   NUMERIC(4,1) NOT NULL,  
    PRIMARY KEY(SID, ATYP, ANR),  
    FOREIGN KEY(SID)  
        REFERENCES STUDENTEN,  
    FOREIGN KEY(ATYP, ANR)  
        REFERENCES AUFGABEN)
```

Beispiel zu Fremdschlüsseln

- Sie haben die Punkte-Datenbank angelegt, u.a.:

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

- Die Tabelle mit den Bewertungen enthält Fremdschlüssel:

`BEWERTUNGEN(SID → STUDENTEN, ...)`

- Was würden Sie tun, um zu prüfen, ob Ihr DBMS diesen Fremdschlüssel tatsächlich überwacht?

In MySQL wurden lange Fremdschlüssel im `CREATE TABLE` akzeptiert (aus Kompatibilitätsgründen), aber nicht überwacht.

Präsenzaufgabe: Schema für Koch-/Back-Rezepte

- Entwerfen Sie ein Schema für Rezepte (zum kochen/backen) und geben Sie es in der Kurznotation ab. Anforderungen:
 - Zu jedem Rezept muss eine Nummer, eine Bezeichnung, ein Schwierigkeitsgrad, und eine Dauer (der Zubereitung) gespeichert werden. Bezeichnungen sind nicht eindeutig.
 - Ein Rezept besteht aus mehreren Schritten. Es ist die Reihenfolge der Schritte festzuhalten, außerdem ein Text (Anweisung für den Schritt).
 - Für jeden Schritt ist zu speichern, welche Zutaten in welcher Menge benötigt werden. In einen Schritt können auch mehrere Zutaten benötigt werden (oder keine).

Die gleiche Zutat in verschiedenen Schritten eines Rezepts ist möglich.

- Sie müssen als Gruppen aus den „Breakout-Rooms“ abgeben.

Es reicht eine ASCII .txt Datei. Wenn Sie wollen, PDF.