

GDB Tutorium

Woche01

Jigao Luo

TUM

24. Oktober 2019

- Vorname: Jigao
- Master 1.Sem: Data Engineering and Analytics
- Herkunft: China VR
- Email: jigao.luo@tum.de
- Slide auf GitHub: keine Lösung

Diese Folien sind von mir privat erstellt worden.

Ich kann bei aller Sorgfalt nicht ausschließen, dass Fehler in den Folien enthalten sind.

Sollte es Unstimmigkeiten zu den Vorlesungsfolien von Prof. Neumann geben, so stimmen die Folien aus der Vorlesung.

Falls Ihr einen Fehler(+ Tippfehler + Grammatikfehler) oder eine Unstimmigkeit findet, schreibt an jigao.luo@tum.de und auf GitHub Repo Issue.

- Bonus nur wenn Übungspunkte und VL-Quizzes erfüllt
- In der Übung
 - Anwesenheit $\rightarrow +1$ P, auch bei Krankheit falls Attest vorgezeigt wurde
 - Vorrechnen $\rightarrow +1$ P (evtl. auch für Teilaufgaben)
 - Anz. Übungswochen $+ 2 = 16$ Punkte notwendig

- Bonus nur wenn Übungspunkte und VL-Quizzes erfüllt
- In der Übung
 - Anwesenheit $\rightarrow +1$ P, auch bei Krankheit falls Attest vorgezeigt wurde
 - Vorrechnen $\rightarrow +1$ P (evtl. auch für Teilaufgaben)
 - Anz. Übungswochen $+ 2 = 16$ Punkte notwendig
- In der VL:
 - regelmäßige Moodle-Quizzes
 - 25% der Punkte aus Quizzes notwendig

- Bonus nur wenn Übungspunkte und VL-Quizzes erfüllt
- In der Übung
 - Anwesenheit → +1 P, auch bei Krankheit falls Attest vorgezeigt wurde
 - Vorrechnen → +1 P (evtl. auch für Teilaufgaben)
 - Anz. Übungswochen + 2 = 16 Punkte notwendig
- In der VL:
 - regelmäßige Moodle-Quizzes
 - 25% der Punkte aus Quizzes notwendig
- Wenn beide Bedingungen erfüllt: 0.3 Notenbonus auf Haupt- und Nachklausur

- Wann beginnt?
- Wiederholung
- Aufgabenbesprechung
- Anwesenheitskontrolle

- Grundlagen: Datenbanken Website lesen + Web-tools
- Das Buch: Lehrbuch/Übungsbuch ist in großer Stückzahl in der Bibliothek (auch Teilbibliothek Chemie) vorhanden.
- Youtuber: Prof. Dr. Jens Dittrich
- Kap2-Slide selbst zu lesen.
- ALLE GDB-Website in vorherigen Semester.
- Die Übungen müssen zuhause vorbereitet werden. Die Zeit wird sonst nicht ausreichen.

Database \neq Database Management System

- Database: Tabelle + Relation Algebra + ...: in GDB
- (eher eine Darstellung mit DS)

- DB Management System:
 - C / C++
 - Linux
 - Networking
 - Hardware
 - SQL Compiler (Compilation)
 - Mathematische Query Optimierung (auch in GDB, aber nur Grundlage)
 - Software Engineering + System Engineering

Rank institutions in the world by publications from 2009 to 2019

AI [off | on]

- Systems [off | on]

- ACM SIGMOD




















SIGMOD

VLDB

ICDE

PODS

- ▶ Design automation
- ▶ Embedded & real-time systems
- ▶ High-performance computing
- ▶ Mobile computing
- ▶ Measurement & perf. analysis
- ▶ Operating systems
- ▶ Programming languages
- ▶ Software engineering

#	Institution	Count	Faculty
1	▶ National University of Singapore 	58.2	24
2	▶ Chinese University of Hong Kong 	50.1	9
3	▶ HKUST 	47.4	13
4	▶ UNSW 	42.4	8
5	▶ Tsinghua University 	42.1	22
6	▶ University of Edinburgh 	30.6	9
7	▶ Massachusetts Institute of Technology 	28.9	17
8	▶ University of Waterloo	28.5	18
9	▶ EPFL 	28.4	10
10	▶ TU Munich 	27.4	4
11	▶ Duke University 	27.1	7
12	▶ Stanford University 	26.3	11
13	▶ Aalborg University 	24.8	9
14	▶ University of Michigan 	24.3	10
15	▶ University of Wisconsin - Madison 	24.1	15
16	▶ University of California - Berkeley 	23.8	16
17	▶ University of Maryland - College Park 	23.3	11
18	▶ University of Oxford 	22.6	7
19	▶ University of Washington 	22.4	8
20	▶ ETH Zurich 	22.2	8

Formulieren Sie die folgenden Anfragen auf dem Universitätsschema 1 in Relationenalgebra. Geben Sie die Lösungen in der in der Vorlesung besprochenen Operatorbaum-Darstellung an.

a) Geben Sie Namen und Semester aller Studenten an, die mindestens im 7. Semester sind.

Formulieren Sie die folgenden Anfragen auf dem Universitätsschema 1 in Relationenalgebra. Geben Sie die Lösungen in der in der Vorlesung besprochenen Operatorbaum-Darstellung an.

b) Geben Sie die Namen aller Professoren an, die mindestens eine Vorlesung mit mindestens 4 SWS lesen.

Formulieren Sie die folgenden Anfragen auf dem Universitätsschema 1 in Relationenalgebra. Geben Sie die Lösungen in der in der Vorlesung besprochenen Operatorbaum-Darstellung an.

c) Geben Sie alle Vorlesungen an, die der Student Xenokrates gehört hat.

alle Vorlesungen \equiv VorlNr + Titel

Formulieren Sie die folgenden Anfragen auf dem Universitätsschema 1 in Relationenalgebra. Geben Sie die Lösungen in der in der Vorlesung besprochenen Operatorbaum-Darstellung an.

d) Geben Sie die Namen aller Assistenten an, deren Boss mindestens eine Vorlesung geprüft hat.

Formulieren Sie die folgenden Anfragen auf dem Universitätsschema 1 in Relationenalgebra. Geben Sie die Lösungen in der in der Vorlesung besprochenen Operatorbaum-Darstellung an.

e) Geben Sie die Titel der direkten Voraussetzungen für die Vorlesung Wissenschaftstheorie an.

a) Die Installation und Wartung eines Datenbanksystems ist aufwendig, die Erstellung eines eigenen Datenformats ist straight-forward und flexibler.

b) Mehrbenutzersynchronisation wird in diesem Fall nicht benötigt.

c) Es ist unsinnig, dass jeder Entwickler zunächst eine eigene Anfragesprache (SQL) lernen muss, nur um Daten aus der Datenbank zu extrahieren.

d) Redundanz ist hilfreich, wieso sollte man auf sie verzichten?

Gruppenaufgabe 1

NICHT KLAUSURRELEVANT

- MySQL
- SQLite
- PostgreSQL

Finden Sie ein Beispiel für ein Problem (bzw. eine Inkonsistenz), die auftreten kann, wenn **unkontrolliert parallel** auf Daten zugegriffen wird.

Ein traditionelles Beispiel hierfür ist eine gegenseitige Bank-Überweisung zwischen zwei Konten A und B. Wenn A einen Betrag x zu B überweist und B einen Betrag x_0 zu A, sollte immer gelten $\text{Kontostand}(A) + \text{Kontostand}(B)$ ist konstant, da sonst Geld verschwunden ist.

Konstruieren Sie einen Ablauf zweier gegenseitiger Überweisungen, bei dem die Eigenschaft, dass die Kontostandssumme konstant sein soll nach dem Abschluss der zwei Überweisungen verletzt ist.

Stimmt es?

- 1 A liest eigenen Kontostand in Variable a ein.
- 2 B liest eigenen Kontostand in Variable b ein.
- 3 A liest Bs Kontostand in Variable b' ein.
- 4 B liest As Kontostand in Variable a' ein.

Stimmt es?

- 1 A liest eigenen Kontostand in Variable a ein.
- 2 B liest eigenen Kontostand in Variable b ein.
- 3 A liest Bs Kontostand in Variable b' ein.
- 4 B liest As Kontostand in Variable a' ein.

Ja, da read(lesen) shared Zugriff (nicht blockierend, nicht exclusive) haben sollte.

- Wie ein multithreaded Program interpretieren.
- Critical Section: write (Anfang bis Ende).
- Atomic: inkrement/dekrement + das Zurückschreiben.
- Readers–writer lock: single-writer lock

Ab wann stimmt das Beispiel aus der Lösung nicht?
Wie wird es **kontrolliert parallel** auf Daten zugegriffen?