

Datenbank-Programmierung

Kapitel 0: Organisatorisches

Prof. Dr. Stefan Brass

Martin-Luther-Universität Halle-Wittenberg

Sommersemester 2022

<http://www.informatik.uni-halle.de/~brass/dbp22/>

Inhalt

- 1 Inhalte
- 2 Motivation
- 3 Organisatorisches
- 4 Prüfung, Hausaufgaben
- 5 Literatur
- 6 Software
- 7 Schlussbemerkungen

Organisatorischer Hinweis

- Die Vorlesung „Datenbanken I“ (10 LP) wurde aufgespalten:
 - Einführung in Datenbanken (5 LP) (Wintersemester)
 - Datenbank-Programmierung (5 LP) (Sommersemester)
- Wenn Ihre Prüfungsordnung „Datenbanken I“ fordert, werden beide Module zusammen dafür angerechnet.
- Für Informatiker und Bioinformatiker ist nach der neuen Prüfungsordnung nur „Einführung in Datenbanken“ Pflicht.

Dieser zweite Teil, „Datenbank-Programmierung“, ist Wahlpflicht. Wenn Sie sich im Master in Datenbanken spezialisieren wollen, sollten Sie ihn aber hören. Wenn Sie sonst ernsthaft mit Datenbanken arbeiten wollen, auch.
- Für Wirtschaftsinformatiker (neue Ordnung 2020) ist DB-Programmierung auch Wahlpflicht, aber Wahl 4/5.

Themen der Vorlesung

- PostgreSQL, psql Kommandoschnittstelle, SQL Skripte
- Tabellen-Deklarationen (vertieft)
- Updates, Transaktionen, Mehrbenutzerbetrieb
- JDBC-Schnittstelle (Datenbank-Zugriff aus Java)
- Rekursive Anfragen in SQL, Reguläre Ausdrücke in SQL
- Data Warehouses/OLAP, Fortgeschrittenes SQL
- Trigger, Serverseitige Programmierung
- Index-Strukturen (kurz)
- Sicherheits-Aspekte (Zugriffsrechte, Datenschutz, Sichten)
- Ggf. Web-Datenbank-Programmierung (HTML, HTTP, PHP)

Weiterer Nutzen dieser Vorlesung (1)

- Mehr Zeit und Gelegenheit, um SQL zu üben. In der Klausur zu „Einführung in Datenbanken“ gab es recht viele Fehler:
 - Es gab 483 Anfragen von 110 Studenten (5 SQL Aufgaben).
 - Davon lieferten 151 (31%) die richtige Antwort im Beispiel-Zustand.

Man konnte die Anfrage im Beispiel-Zustand mit dem Adminer testen. Es war also klar, dass die abgegebene Anfrage nicht korrekt war. Ein Teil der Anfragen, die im Beispielszustand funktionierten, waren trotzdem falsch (z.B. so gebastelt, dass sie genau in diesem Zustand funktionieren). Das wird bei der mühsamen und frustrierenden manuellen Kontrolle gefunden. Plagiate gab es bei der Online-Klausur auch.
 - 293 (61%) liefern schon im bekannten Testzustand die falsche Antwort.
 - 39 (8%) waren nicht ausführbar (Syntaxfehler).

Weiterer Nutzen dieser Vorlesung (2)

- Einige DB-Themen, die für praktische Projekte wichtig sind, aber nicht mehr in die Einführungs-Vorlesung passten.
 - „Einführung in Datenbanken“ bringt Sie soweit, dass Sie das relationale Modell verstanden haben, und Anfragen an existierende Datenbanken stellen könnten. Es reicht aber nicht, um ein Datenbank-Projekt alleine stemmen zu können.
- „Modernes SQL“: Jede Version führt neue Konstrukte ein.
 - Der größte Teil der Einführungs-Vorlesung behandelt Konstrukte aus SQL-86 und SQL-92 (Ausnahme: WITH).
- Mehr Erfahrung in der Java-Programmierung.
 - Auch Werkzeuge wie Git, maven und Docker werden in der Übung eingeführt.
- Zum DB-Entwurf und zur DBMS-Implementierung (inkl. Administration, Tuning) gibt es Master-Vorlesungen.
 - Administration und Performance-Tuning werden hier kurz behandelt.

Inhalt

- 1 Inhalte
- 2 Motivation**
- 3 Organisatorisches
- 4 Prüfung, Hausaufgaben
- 5 Literatur
- 6 Software
- 7 Schlussbemerkungen

Motivation (1)

- Viele Anwendungen brauchen große Mengen von Daten.
- Zum Teil fallen die großen Datenmengen auch einfach an (z.B. bei Verkaufstransaktionen, Interaktion mit Webseiten). Sie sollen dann auch zum Vorteil der Firma genutzt werden.
- Datenbanken sind das Speichermedium für strukturierte Informationen (z.B. Tabellen).
- Selbst wenn die für die Anwendung benötigten Datenmengen nicht groß sind, hat man mit den Tabellen eine einfache und übersichtliche Struktur.
- Android enthält die SQLite Bibliothek. Viele Anwendungen auf Mobiltelefonen nutzen diese Datenbank.

Z.B. für Konfigurationsdaten. Auch Apple nutzt diese Datenbank in iOS.

Webbrowser wie Firefox und Chrome nutzen SQLite (z.B. Bookmarks). Skype.

Motivation (2)

- Im Prinzip kann man eine Datenbank auch wie eine Tabellenkalkulation (Excel) nutzen.

Tabellenkalkulationen haben enge Grenzen hinsichtlich der Komplexität der Strukturen und der Größe der Daten. Datenbanken dagegen praktisch nicht.

- Bei vielen DBMS kann man SQL-Skripte schreiben, die
 - eine Datenbank anlegen,
 - Daten z.B. aus CSV-Dateien laden,
 - einige Auswertungen machen und die Ergebnisse z.B. im HTML-Format ausgeben, und
 - die Datenbank wieder löschen.
- Mit SQL kann man auch relativ komplexe Auswertungen der Daten kompakt aufschreiben.

Motivation (3)

- Wenn man sich mit Datenbanken auskennt, kann man Technologie nutzen, die seit Jahrzehnten von Forschern, Firmen und Open Source Communities entwickelt wurde.

SQL ist aber mehr als `SELECT FROM WHERE`. Investieren Sie jetzt Zeit!

- SQL bleibt modern.

Andere Frameworks und Technologien kommen und gehen, SQL gibt es schon lange, und es ist kein Ende abzusehen. SQL entwickelt sich weiter. Da SQL eine deklarative Sprache ist, können DBMS (SQL-Implementierungen) leicht an neue Technologien angepasst werden.

- Transaktionen und die Synchronisation paralleler Zugriffe werden schon bei einfachen Web-Anwendungen wichtig.

Datenbanken bieten dafür wichtige Unterstützung, aber nicht immer vollautomatisch, so dass Wissen (z.B. aus dieser Vorlesung) wichtig ist.

Inhalt

- 1 Inhalte
- 2 Motivation
- 3 Organisatorisches**
- 4 Prüfung, Hausaufgaben
- 5 Literatur
- 6 Software
- 7 Schlussbemerkungen

Ansprechpartner (1)

Dozent: Prof. Dr. Stefan Brass

- Email: brass@informatik.uni-halle.de

Betreff-Zeile sollte mit [dbp22] beginnen, möglichst aussagefähig.

Nutzen Sie auch das StudIP-Forum für Fragen von allgemeinem Interesse.

- Büro: Von-Seckendorff-Platz 1, Raum 313
- Telefon: 0345/55-24740
- Sprechstunde: Montags, 12¹⁵–13⁰⁰
- Frühere Unis: Braunschweig, Dortmund, Hannover, Hildesheim, Pittsburgh, Gießen, Clausthal.
- Oracle8 Certified Database Administrator (aktuell: 19C/21C)
- IBM Certified Advanced DBA (DB2 UDB 8.1) (aktuell: 12)

Ansprechpartner (2)

Übungsleiter: MSc. Mario Wenzel

- Büro: Von-Seckendorff-Platz 1, Raum 315
- Telefon: 0345/55-24776
- Email: mario.wenzel@informatik.uni-halle.de

Sekretärin: Ramona Vahrenhold

- Büro: Von-Seckendorff-Platz 1, Raum 324
- Telefon: 0345/55-24750, Fax: 0345/55-27333
- Email: vahrenho@informatik.uni-halle.de

Zeit und Ort (1)

Vorlesung (2 SWS):

- Montags, 10¹⁵–11⁴⁵, Raum 3.04.
- Mit Video-Aufzeichnung (online asynchron anzuschauen).

Die Aufzeichnung steht vermutlich ab ca. 17 Uhr in StudIP (Reiter „OpenCast“) zur Verfügung. Es ist empfohlen, die Aufzeichnung wirklich anzuschauen, und nicht nur die (recht ausführlichen) Folien zu lesen, da es zusätzliche Erläuterungen und Betonungen gibt („Was ist besonders wichtig?“). Wenn man nicht regelmäßig dranbleibt, wird man schnell „abgehängt“.

- Bei der Präsenzveranstaltung muss eine FFP2-Maske getragen werden.

Kommen Sie bitte nur, wenn Sie keine Corona- oder Erkältungssymptome haben und in der letzten Woche kein erhöhtes Ansteckungsrisiko hatten. Wenn Sie zu spät kommen, sind Sie auf der Videoaufnahme zu sehen.

Zeit und Ort (2)

Übung (2 SWS):

- Zwei Gruppen (Anmeldung: StudIP):

Nr	Tag	Zeit	Raum	Beginn
1	Donnerstag	12 ¹⁵ –13 ⁴⁵	(online)	22.04.
2	Donnerstag	14 ¹⁵ –15 ⁴⁵	(online)	22.04.

Die Links zur Videokonferenz finden Sie im Forum in StudIP.

- Abonnieren Sie mindestens den Bereich „Ankündigungen“ im **Forum** in **StudIP**.
- Es gibt vorproduzierte Lehrvideos (aus dem letzten Jahr), die Übung ist eher als Gelegenheit zu Fragen und zum Austausch gedacht (auch über Hausaufgaben).

Die 90 min werden nicht voll ausgenutzt (mit Lernvideo schon).

Zeitliche Belastung

- Diese Vorlesung hat 5 Leistungspunkte.
- Entspricht 150 Stunden studentischer Arbeitszeit:

Lernform	SWS	Stunden
Vorlesung	2	30
Übung	2	30
Lösen von Hausaufgaben	0	30
Selbststudium	0	30
Prüfungsvorbereitung	0	30

Nutzen Sie ausreichend Zeit für praktische Übung am Rechner (Hausaufgaben, Hausarbeit, Mitmach-Videos in den Übungen). „Hands-on Experience“ ist nötig! „Selbststudium“ soll nicht verplante Zeit sein, z.B. das Lesen einiger Kapitel eines Lehrbuchs oder anderer Quellen (z.B. zur Nacharbeit der Vorlesung).

Modulanmeldung (1)

- Für fast alle Studiengänge ist die Modulanmeldung über das Löwenportal Pflicht.

Die Anmeldung muss spätestens zwei Wochen nach Vorlesungsbeginn erfolgen. Die Modulanmeldung ist Voraussetzung für die spätere Anmeldung zur Prüfung. [<http://loewenportal.uni-halle.de/>]

Falls über Löwenportal nicht möglich, dann im Prüfungsamt. Vergessen Sie das nicht, im schlimmsten Fall können Sie am Ende keine Prüfung ablegen, obwohl Sie das ganze Semester über mitgearbeitet haben!

- Für „Lehramt an Sekundarschulen“: extra Modul.

Mit gleicher Vorlesung und Übung, aber mit einer mündlichen Prüfung statt Klausur (ab Immatrikulation 2012 vorgeschrieben).

- Außerdem muss man sich bei StudIP anmelden.

Zugang zur Vorlesungs-Aufzeichnung und Daten der Übung, auch wichtiger EMail-Verteiler: [<http://studip.uni-halle.de/>] [[Diese Vorlesung](#)]

Modulanmeldung (2)

- Das Modul „Datenbank-Programmierung“ hat als „obligatorische Voraussetzungen“ die Module:
 - Einführung in Datenbanken
 - Objektorientierte Programmierung
 - „Programmierkenntnisse“ standen auch bei „Datenbanken I“ als wünschenswerte Voraussetzung in der Modulbeschreibung.
- D.h. Sie können sich nicht zu diesem Modul anmelden, wenn Sie die beiden Module nicht schon bestanden haben.
 - Die Klausurergebnisse zu „Einführung in Datenbanken“ sollen nächste Woche vorliegen. Nachholklausur zu EDB ist noch vor DBP-Prüfung.
- Wenn Sie noch zu „Datenbanken I“ angemeldet sind, können Sie Prüfungen zu beiden Teilen machen.

Inhalt

- 1 Inhalte
- 2 Motivation
- 3 Organisatorisches
- 4 Prüfung, Hausaufgaben**
- 5 Literatur
- 6 Software
- 7 Schlussbemerkungen

Prüfung (1)

- Geplante Prüfungstermine (können sich noch ändern):
 - Montag, 12.09.2022, 10⁰⁰–12⁰⁰:
Elektronische Klausur in Präsenz.
 - Montag, 30.01.2023 (letzte Vorlesungswoche), 14⁰⁰–16⁰⁰:
Ebenso oder ggf. mündliche Prüfungen.

Bei wenigen Teilnehmern wird es mündliche Prüfungen (voraussichtlich als Videokonferenz) geben. Die müssten dann eventuell über mehrere Tage verteilt werden, z.B. auch am Dienstag, 31.01.2023.

- Gibt es Diskussionsbedarf?

Da nur Präsenzteilnehmer mitdiskutieren können, wäre auch eine Diskussion in den Übungsgruppen möglich. Eventuell gibt es eine Abstimmung in StudIP. Ich habe auch über Montag, 25.07.2022 (zweite Woche nach Ende der Vorlesungszeit), nachgedacht. Das läge allerdings schon tief in den Schulferien.

Prüfung (2)

- Bei der Klausur sind bis zu 10 Seiten Notizen erlaubt.

Es ist eine gute Klausurvorbereitung, sich selbst das Wichtigste auf begrenztem Platz in eigener Strukturierung aufzuschreiben (inkl. „SQL Quick Reference“). Etwas aus dem Internet auszudrucken, und nicht vor der Klausur zu lesen, wird dagegen nur unnötig Zeit kosten. Es ist aber legal, auch Internet-Quellen in die 10 Seiten aufzunehmen.

- Mindestens eine (interessante) SQL-Anfrage kommt dran.
- Es ist möglich, dass es eine kleine Programmieraufgabe in Java mit der JDBC-Datenbank-Schnittstelle gibt.

Die Musterlösung wird nicht mehr als 20 Zeilen haben.

- Auch ein Trigger in PostgreSQL mit PL/pgSQL Programmcode könnte gefordert werden.
- Plus Ankreuzaufgaben, u.a. zum Mehrbenutzerbetrieb.

Studienleistung (1)

- Sie bekommen die Leistungspunkte für das Modul nur, wenn Sie
 - die Prüfung bestehen (Klausur) und
 - sich die Studienleistung erarbeiten (Hausaufgaben und „aktive Mitarbeit“ in den Übungen).
- Es gibt wöchentliche Hausaufgaben.
- Für die Studienleistung müssen Sie 50% der Punkte erwerben (in der Summe über das Semester).
- Hausaufgaben müssen einzeln bearbeitet werden.
 - Keine Gruppenarbeit, weil zu häufig nur „Namen mit draufgeschrieben“.
 - Bei zu ähnlichen Abgaben bekommen alle Beteiligten 0 Punkte.
 - Im Wiederholungsfall wird die Studienleistung für dieses Semester verweigert.

Studienleistung (2)

- Es gibt eine relativ sanfte Teilnahmepflicht an den Übungen:
 - Sie müssen mindestens an der Hälfte der Online-Übungen (also ca. 7) teilgenommen haben.

Wenn Sie angesprochen werden, und nicht antworten, oder lange nicht in den zugewiesenen Breakout-Raum wechseln, wird die Teilnahme nicht verbucht.
 - Zweck der Teilnahmepflicht ist ein leichter Druck zu regelmäßiger Beschäftigung mit dem Stoff.

Und auch wenigstens minimale soziale Kontakte zwischen den Studierenden.
 - Sie müssen bereit sein, in der Übung Ihre abgegebenen Hausaufgaben zu erklären.

Ohne Schutz vor Plagiaten würden die Hausaufgaben nicht unbedingt zu einer regelmäßigen Beschäftigung mit dem Stoff führen.

Studienleistung (3)

- Bei den Übungen sollen die Teilnehmenden aktiv mitarbeiten:

- Ziel ist, dass jede/r Teilnehmer/in mindestens 3–5 Wortbeiträge im Semester macht (gerne mehr).

Darüber wird nicht Buch geführt. Sie müssen aber mit Mikrophon angemeldet sein, und jederzeit bereit sein, zu sprechen. Die Hoffnung ist, dass eine Atmosphäre entsteht, in der die Teilnehmenden es natürlich finden, ungezwungen das Wort zu ergreifen.

- Regelmäßige implizite Hausaufgabe:
Überlegen Sie sich jede Woche mindestens eine Frage zum Stoff der Vorlesung oder Übung.

Sie können aufgefordert werden, ihre Frage zu stellen, damit ein Gespräch in Gang kommt (und auch sonst eher stille Studierende die Gelegenheit bekommen, die Forderung nach „aktiver Mitarbeit“ zu erfüllen).

- Es gibt Gruppenarbeit in „Breakout“-Räumen.

Studienleistung (4)

Stellen Sie Fragen!

- Eventuelle Fragen zur Korrektur der Hausaufgaben müssen vollständig aufgeklärt werden.

Man kann aus Fehlern nur lernen, wenn man den genauen Grund gut versteht. Ausserdem kommen bei der Korrektur gelegentlich auch Fehler vor.

Wir garantieren, dass nicht nachträglich Punkte abgezogen werden, wenn Sie wegen einer eventuell doch falschen Lösung fragen. Wenn Sie bei einer Aufgabe Punkte dazu bekommen, dürfen aber auch die anderen Aufgaben des Blattes neu bewertet werden. Die Summe der Punkte sinkt jedoch nicht.

- Das gilt natürlich für alle Probleme, auf die Sie bei Ihrer Beschäftigung mit dem Vorlesungsstoff stoßen.

Es ist sicher falsch, solange herumzuprobieren, bis es funktioniert, und sich dann zufrieden zu geben. Sie müssen auch verstehen, warum es vorher nicht funktioniert hat. Auch das Skript enthält möglicherweise Fehler.

Lehrvideos zu den Hausaufgaben

- In der Übung wird auch zusätzlicher praktischer Stoff vermittelt, der die Vorlesung ergänzt, z.B.
 - Docker (Container-Technologie),
 - Maven (Paketmanager und „Build“-Werkzeug u.a. für Java),
 - Git (Versionsverwaltung),
 - SQLite3 (DBMS).
- Der Übungsleiter hat dazu 2019–2021 Lehrvideos erstellt.

Es sind „Videos zum Mitmachen“, d.h. Sie sollen an einigen Stellen die Pause-Taste drücken und selbst etwas ausprobieren.
- Wenn Sie die Lehrvideos (Teil der Übung) durcharbeiten, werden Ihnen die Hausaufgaben deutlich leichter fallen!

Die Zeit für die Lehrvideos zählt zum Teil zur Hausaufgaben-Zeit, zum Teil wird die Online-Übung etwas verkürzt (danach Sprechstunde des Übungsleiters).

Inhalt

- 1 Inhalte
- 2 Motivation
- 3 Organisatorisches
- 4 Prüfung, Hausaufgaben
- 5 Literatur**
- 6 Software
- 7 Schlussbemerkungen

Vorlesungs-Materialien

- Die Folien der Vorlesung werden noch überarbeitet.
- Sie werden kurz vor der Vorlesung auf die Webseite gestellt:

<http://www.informatik.uni-halle.de/~brass/dbp22/>

Die Webseite enthält auch weitere nützliche Informationen, u.a. eine Link-Sammlung. Empfehlungen für weitere Seiten sind willkommen.

- Ich finde es wichtig, auch zusätzliche Literatur zu lesen, aber nehmen Sie doch bitte auch meine Folien zur Kenntnis!

Warum z.B. in der Klausur exotische SQL-Konstrukte verwenden, die in der Vorlesung nicht dran waren, und die deutlich weniger portabel sind und im konkreten Fall keine Vorteile gegenüber den behandelten Konstrukten haben? Auch die Definition von BCNF geht meistens schief, wenn Sie es anders als in der Vorlesung vorgestellt versuchen.

- Folien zum ersten Teil „Einführung in Datenbanken“:

<http://www.informatik.uni-halle.de/~brass/db21/>

Lehrbücher (1)

- Alfons Kemper / André Eickler:
Datenbanksysteme. Eine Einführung. 10. Auflage.
De Gruyter, 2015, ISBN 3110443759, 880 Seiten, 49.95 €.
- Gunter Saake, Kai-Uwe Sattler, Andreas Heuer:
Datenbanken. Konzepte und Sprachen. 5. Aufl.
MITP Bonn, 2013, ISBN 3-8266-9453-8, 784 Seiten, 39.95 €.
- Gottfried Vossen:
Datenbankmodelle, Datenbanksprachen und
Datenbankmanagementsysteme. 5. Aufl.
Oldenbourg, 2008, 3-486-27574-7, 842 Seiten, 64.95 €.

Lehrbücher (2)

- Georg Lausen:
Datenbanken. Grundlagen und XML-Technologien.
Spektrum/Elsevier, 2005, ISBN 3-8274-1488-1, 286 Seiten, 22.99 €.
- Ramez Elmasri / Shamkant B. Navathe:
Fundamentals of Database Systems, 7th Ed.
Pearson India, 2017, ISBN 933258270X, ca. 29.00 €
Global Ed.: Prentice Hall, 2016, ISBN 1-292-09761-2, 1272 S., 39.16 €
Deutsch: Bachelorausgabe (Teile fehlen/nur online):
Pearson, 3. Aufl., 2009, ISBN 386894012X, 560 Seiten, 34.95 €.
- Raghu Ramakrishnan, Johannes Gehrke:
Database Management Systems, 3rd Ed.
McGraw Hill, 2002, ISBN 0-07-123151-X, 1065 Seiten, 59.95 €.

Lehrbücher (3)

- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom:
Database Systems: The Complete Book, 2nd Ed.
Prentice Hall, 2009/13, ISBN 1-292-02447-X, 1152 pages, 67.64 €.
- Philip M. Lewis, Arthur Bernstein, Michael Kifer:
Database and Transaction Processing, 2nd Ed.
Addison-Wesley, 2005, ISBN 0-321-26845-8, 1235 pp., 148.99 €.
TB: ISBN 0-321-31256-2, 76.99 €. Es gibt „Introductory Version“.
- Jim Melton, Alan R. Simon: SQL: 1999 —
Understanding Relational Language Components.
Morgan Kaufmann, 2. Aufl., 2001, ISBN 1558604561, 928 pp, 63.23 €.

Bücher sind wichtig!

- Wir sind an einer Uni.

Dies ist nicht nur ein SQL-Kurs (plus JDBC, serverseitige Programmierung). Ziel ist auch die kritische Auseinandersetzung mit dem Stoff (inklusive Stoffauswahl). Eigenes Denken und Verstehen statt Auswendiglernen! Natürlich muss man am Ende syntaktische Details von SQL können. Aber auch: Was ist die beste Lösung für mein datenintensives Anwendungsproblem? Wie könnten Alternativen aussehen? Man braucht mehrere Quellen. Nur aus unterschiedlichen Blickwinkeln sieht man dreidimensional.

- Ich empfehle dringend, ein Lehrbuch zu beschaffen, und in großen Teilen zu lesen.

Folien sind kein Lehrbuch, und unterschiedliche Lernformen ergänzen sich. Je nach Vorwissen/Interessen sind unterschiedliche Bücher ansprechend.

Inhalt

- 1 Inhalte
- 2 Motivation
- 3 Organisatorisches
- 4 Prüfung, Hausaufgaben
- 5 Literatur
- 6 Software**
- 7 Schlussbemerkungen

Software: PostgreSQL

- Verbreitetes Open Source Datenbank-Managementsystem mit umfangreicher SQL-Unterstützung.

Postgres war Vorreiter der objektrelationalen Technologie.

- Empfehlung: Installieren Sie sich PostgreSQL auf Ihrem eigenen Rechner: [<https://www.postgresql.org/>]

Läuft unter Linux, Windows, MacOS, BSD Unix, Solaris. Das würde Ihnen auch Administrator-Zugriff geben. Im Notfall bieten wir ein Remote Login (ssh) auf einem Rechner mit installiertem PostgreSQL (natürlich ohne DBA-Rechte).

- Wir bieten auch eine Adminer Web-Schnittstelle zum Üben von SQL ohne eigene Installation.

[<https://www.adminer.org/>] In diesem Semester bekommt jeder einen eigenen Account, und kann dort auch eigene Tabellen anlegen. Das reicht aber nur für die ersten Wochen, nicht für die Programmier-Aufgaben.

Software: Bedeutung der Portabilität

- PostgreSQL hat durchaus seine Eigenheiten: Dinge, die nur in diesem System funktionieren.
- Halten Sie sich möglichst an die Konstrukte, die in der Vorlesung vorgestellt werden.

Schalten Sie nicht ab, wenn Portabilität diskutiert wird. Es gab bei der Lehrevaluation Kommentare, dass ich mich doch nur auf PostgreSQL konzentrieren sollte. Je mehr ich aber lerne, was in PostgreSQL alles geht, desto entschiedener lehne ich das ab.

- Nicht alles lässt sich in SQL portabel lösen, aber man muss zumindest irgendeinen Vorteil bekommen, wenn man auf Portabilität verzichtet.

Eine kürzere Formulierung, oder ein Spezialkonstrukt, von dem man weiss, dass es besonders effizient implementiert ist.

Software: Weitere DB-Managementsysteme

- MySQL: [<https://www.mysql.com/de/>]
MariaDB: [<https://mariadb.org/>]
- SQLite: [<https://www.sqlite.org/index.html>]
- Oracle: Die „Express Edition“ kostet nichts.
[<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).
Einschränkungen Express-Ed.: Max. 4 GB/11 GB DB, 1 Kern, 1 GB RAM, u.a.
- Microsoft SQL Server 2019: Auch kostenlose Express Edition.
[<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>]
Einschränkungen Express-Edition: Max. 10 GB DB, 1.4 GB Puffer (RAM),
4 Kerne, keine Hochverfügbarkeit, u.s.w.

Software für die Übungen

- In den Übungen wird PostgreSQL verwendet.
- JDBC ist eine Standard-Schnittstelle, und normale SQL-Anfragen sollten in jedem System funktionieren.
 - Es soll dort gerade leicht möglich sein, das DBMS zu wechseln.
- In dem Kapitel über fortgeschrittenes SQL gehen wir aber an die Grenzen der Systeme (neue Sprachkonstrukte).
- Serverseitige Programmierung ist systemabhängig.
 - Das betrifft den Abschnitt über Trigger und Stored Procedures.
- Falls Sie sich für Datenbanken besonders interessieren, wäre wohl vernünftig, sich auch eine Express Edition eines kommerziellen Systems zu installieren.

Docker (1)

- Docker ist eine Software zur „Container-Virtualisierung“.
- Man kann so mehr oder weniger virtuelle Linux-Rechner auf seinem Rechner laufen lassen, die jeder bestimmte Netzwerkdienste zur Verfügung stellen.

Damit ist die Anwendung mit allen ihren Komponenten und Abhängigkeiten in einem Container gekapselt, und kann leicht wieder spurlos von einem System entfernt werden, oder auf einen anderen Rechner verschoben werden. Im Gegensatz zu einer richtigen virtuellen Maschine gibt es fast keinen Performance-Verlust, da das Betriebssystem des Host-Rechners mit benutzt wird (und nicht simuliert). Das setzt entsprechende Abschottungs-Funktionen im Betriebssystem voraus. Seit WSL2 („Windows Subsystem for Linux“, muss zusätzlich installiert werden) läuft Docker auch auf Windows 10 Home.

- Es gibt zu allen großen Datenbanken Docker-Images.

Also Spezifikationen, mit denen man eine passende „virtuelle Maschine“ starten kann.

Docker (2)

- Sie brauchen eine PostgreSQL-Datenbank auf Ihrem Rechner für `psql`-Skripte und für JDBC-Programmierung.

Andere Aufgaben können Sie mit der Adminer-Webschnittstelle bearbeiten. Im Notfall fragen Sie uns nach einer Möglichkeit zum Remote Login auf einem Rechner des Instituts mit installiertem PostgreSQL.

- Sie können den Server in Docker installieren oder normal.

Die normale Installation wird in Kapitel 1 besprochen. Sie hat den Vorteil, dass Sie Dateien und Prozesse leichter beobachten können. Ein Docker-Container ist dagegen ähnlich zu einem getrennten Rechner (mit „Remote Login“).

- In der Industrie wird wahrscheinlich vorausgesetzt, dass unsere Absolventen mit Docker umgehen können.

Dies ist aber ein Datenbank-Kurs, und die 150 h Arbeitszeit für 5 LP sind mit DB-Themen recht gut ausgelastet. Nutzen Sie die Gelegenheit, und hören Sie sich an, was Herr Wenzel in den Übungen zu Docker zu sagen hat.

Inhalt

- 1 Inhalte
- 2 Motivation
- 3 Organisatorisches
- 4 Prüfung, Hausaufgaben
- 5 Literatur
- 6 Software
- 7 Schlussbemerkungen**

Vorlesungs-Etikette

- Vermeiden Sie Verhalten, das Ihre Mitstudenten oder den Professor ablenkt:
 - Vermeiden Sie Gespräche während der Vorlesung.

Wenn Sie Ihren Nachbarn etwas zur Vorlesung fragen müssen: leise und kurz. Wenn die Frage möglicherweise auch für andere interessant ist, stellen Sie sie offiziell (melden, ggf. rufen).
 - Notebooks sollten während der Vorlesung nur die Folien anzeigen (eventuell Notizen, DB).
 - Mobiltelefone bitte auch (wenn überhaupt) nur für vorlesungsrelevante Tätigkeiten verwenden.
- Denken Sie mit und stellen Sie Fragen!
Die Vorlesung soll kein Monolog werden.

E-Learning für SQL:

- Automatische Korrektur von SQL-Anfragen
- SQL-Lernspiele (insbesondere Textadventurespiel)

Deduktive Datenbanken (Sprache „Datalog“):

- Entwicklung einer eigenen Implementierung in C++
 - Interpreter für eine eigene abstrakte Maschine und direkte Compilierung nach C++ (so dann „native Code“). Wir planen parallele Auswertung.
- Benchmarks, auch im Vergleich zu SQL-Datenbanken und NoSQL-Datenbanken
 - Wir arbeiten auch an einer Vorhersage der Laufzeit basierend auf Parametern des Problems (Anfrage und Eingabedaten).