

Einführung in Datenbanken

Kapitel 0: Organisatorisches

Prof. Dr. Stefan Brass

Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2020/21

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

Inhalt

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

Ist dies die richtige Vorlesung? (1)

- 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.

Bei besonderen Schwierigkeiten durch diese Aufspaltung kontaktieren Sie mich.
- Für Informatiker und Bioinformatiker ist nach der neuen Prüfungsordnung nur noch diese Vorlesung Pflicht.

Der zweite Teil, „Datenbank-Programmierung“, ist Wahlpflicht. Wenn Sie sich im Master in Datenbanken spezialisieren wollen, sollten Sie das hören.
- Für Wirtschaftsinformatiker ist der erste Teil Pflicht, der zweite fast (Wahlbereichspflichtbereich 1.3: 4 aus 5).

Ist dies die richtige Vorlesung? (2)

- Dies ist nicht die Vorlesung
„Einführungen in Datenbanken und das WWW (ASQ)“,
die u.a. auch im Ergänzungsstudiengang Medienbildung
belegt werden kann.

Die ASQ-Vorlesung wird momentan leider nicht angeboten.
Falls Sie diese (anspruchsvollere) Vorlesung als Ersatz belegen möchten,
sprechen Sie bitte mit dem Dozenten.
- Falls Sie Lehramt an Sekundarschulen mit Erweiterungsfach
Informatik studieren, benötigen Sie eine mündliche Prüfung.

Eine Prüfung über beide Teile statt der zwei Klausuren. Auch in diesem
Fall bitte den Dozenten ansprechen (gegen Ende).

Themen

Diese Vorlesung ist eine grundlegende Einführung in die Anwendung relationaler Datenbank-Managementsysteme.

Inhalte:

- Grundlegende Begriffe, relationales Datenmodell
- Etwas Logik (mit Anwendungen für Datenbanken/SQL)
 - Man kann viele Datenmodelle und Datenbanksprachen mit den Begriffen der Logik verstehen. Logik hilft, SQL besser zu nutzen.
- Relationale Algebra (kurz)
- Die Datenbanksprache SQL (Schwerpunkt der Vorlesung)
- Einführung in DB-Entwurf (ER-Modell, BCNF, → DB IIA)
- Updates, Transaktionen (sehr kurz)

Fortsetzung: Datenbank-Programmierung

Geplante Inhalte für das Modul im Sommer:

- Fortgeschrittenes/Modernes SQL
 - Insbesondere für Data Warehouse Anwendungen
 - Auch rekursive Anfragen in SQL (ggf. mit Ausblick auf Datalog)
- Transaktionen (Recovery, Mehrbenutzerbetrieb)
 - Synchronization paralleler Zugriffe auf die Datenbank
- Datenbank-Schnittstellen aus Programmiersprachen, JDBC
- Integritäts-Überwachung, Serverseitige Programmierung, Trigger
- Zugriffsrechte, Datenschutz, Sichten
- Ggf. Einführung in die Web-Datenbank-Programmierung

Inhalt

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

Motivation (1)

Warum sind Datenbanken wichtig?

- „Informations-Gesellschaft“. Information ist eines der wichtigsten Aktiva vieler Wirtschaftsunternehmen.
- Datenbanken sind das Speichermedium für strukturierte Informationen (z.B. Tabellen).
- Sie werden im Berufsleben in der einen oder anderen Form mit Datenbanken zu tun haben.
- Viele Webseiten beziehen die Inhalte aus Datenbanken. Jeder Webshop hat als Grundlage eine Datenbank.
- „Big Data“ ist in aller Munde.
 - Häufig werden die Daten für Analysen in Datenbanken abgelegt.
 - Datenbanken müssen nicht immer klassische relationale SQL-Systeme sein.

Motivation (2)

Warum machen Datenbanken Spaß?

- Interessante Verbindung von Theorie und Praxis, z.B. beruht SQL auf der Prädikatenlogik 1. Stufe.

Viele DB-Begriffe sind so einfach, dass noch formal zu definieren.

- Zumeist sehr anschaulich, anwendungsorientiert.
- Relativ wenig komplexe Programmierung, wenn man nicht gerade ein neues DBMS implementiert.
- Interessante Algorithmen und Datenstrukturen, gute Anwendung für Parallelität.
- Noch nicht von Microsoft dominiert.

Motivation (3)

Aufgabe:

- Wo begegnen Ihnen im normalen Leben Datenbanken?
Welche enthalten Informationen über Sie?
 - Supermarkt (Scannerkasse)
 - _____
 - _____
 - _____
 - _____
 - _____

Inhalt

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

Ansprechpartner (1)

Vorlesung und Übungsgruppe 1–3: Prof. Dr. Stefan Brass

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

Betreff-Zeile sollte mit [db20] beginnen, möglichst aussagefähig.

- Büro: Von-Seckendorff-Platz 1, Raum 313

Ich mache viel „Home Office“, bin dort also eher selten zufällig anzutreffen.

- Sprechstunde (Büro, 0345/55-24740): Montags, 12¹⁵–13⁰⁰

- Online-Sprechstunde: Montags, 16³⁰–16⁴⁵ (und nach Bedarf),
[\[https://mluconf.uni-halle.de/b/ste-p6j-fmd-i21\]](https://mluconf.uni-halle.de/b/ste-p6j-fmd-i21)

- Frühere Unis: Braunschweig, Dortmund, Hannover, Hildesheim, Pittsburgh, Gießen, Clausthal.

- Oracle8 Certified Database Administrator (aktuell: 12C→18C,19C)
IBM Certified Advanced DBA (DB2 UDB 8.1) (aktuell: 12)

Ansprechpartner (2)

Gruppe 4 der Übungen: PD Dr. Alexander Hinneburg

- Büro: Von-Seckendorff-Platz 1, Raum 314
- Telefon: 0345/55-24732
- Email: hinnebur@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, 10³⁰–12⁰⁰, Raum 3.28.
Und online asynchron als Aufzeichnung.
- Wegen der realen Gefahr durch Corona (COVID-19) ist möglichst großer Abstand zu halten und die Plätze im Hörsaal reichen nicht für alle Teilnehmer.

Momentan werden die beschränkten Plätze über eine Umfrage in StudIP vergeben. Nur wer dort „ja“ angekreuzt hat, darf kommen. Im Gebäude und im Hörsaal gilt Maskenpflicht (inzwischen selbst dann, wenn der Mindestabstand eingehalten werden kann). Es wird häufig gelüftet (ziehen Sie sich warm an). Nutzen Sie die [Corona-Warn-App](#).

- Die Aufzeichnung wird (mit etwas Verzögerung) über StudIP zugänglich gemacht (ggf. zusätzlich vorher anderes Video).

Zeit und Ort (2)

Tafelübung (2 SWS):

- Vier Gruppen (Anmeldung: StudIP):

Nr	Tag	Zeit	Raum	Beginn
1	Dienstag	10 ³⁰ –12 ⁰⁰	(online)	3.11.
2	Dienstag	15 ³⁰ –17 ⁰⁰	(online)	3.11.
3	Dienstag	18 ⁰⁰ –19 ³⁰	(online)	3.11.
4	Dienstag	10 ³⁰ –12 ⁰⁰	1.30+online	3.11.

Gruppe 1: [<https://mluconf.uni-halle.de/b/ste-phq-nam-b6r>] (26 Teilnehmer)

Gruppe 2: [<https://mluconf.uni-halle.de/b/ste-ibu-5h9-lnw>] (27 Teilnehmer)

Gruppe 3: [<https://mluconf.uni-halle.de/b/ste-nmr-59o-s1s>] (25 Teilnehmer)

Gruppe 4: voraussichtlich Raum 1.30 + online (StudIP Reiter „Meetings“)
(29 Teilnehmer)

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	45
Spezielle Prüfungsvorbereitung	0	15

„Lösen von Hausaufgaben“ enthält Zeit für praktische Übung am Rechner (bes. SQL). „Hands-on Experience“ ist nötig!

Falls Sie deutlich mehr als zwei Stunden pro Woche für die Hausaufgaben benötigen, und Ihre Bekannten auch, sollten Sie das sagen.

„Selbststudium“ könnte das Lesen eines Lehrbuches sein. Es enthält auch die 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 (ggf. Verlängerung als Corona-Erleichterung).

Die Modulanmeldung ist Voraussetzung für die spätere Anmeldung zur Prüfung. [<http://loewenportal.uni-halle.de/>]

Falls über Löwenportal nicht möglich, dann im Prüfungsamt.

- 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.

Als Zugang zur Übungsplattform. Bitte auch für die Vorlesung registrieren (EMail-Verteiler): [<http://studip.uni-halle.de/>]

Modulanmeldung (2)

Modulvoraussetzungen (Bedingung für Teilnahme):

- Studienleistung von „Objektorientierte Programmierung“.

Wenn Sie die Voraussetzung nicht haben, und sich nicht im Löwenportal anmelden können, könnten Sie beim zuständigen Prüfungsausschuss nach einer (gut begründeten) Ausnahme fragen. SQL ist eine sehr spezielle Programmiersprache. Natürlich wird SQL in der Vorlesung von Grund auf definiert. Erfahrungen mit formaler Syntax und z.B. Wertausdrücke und Bedingungen in Programmiersprachen wären aber zum Verständnis nützlich. Einzelne Hausaufgaben setzen ggf. Java voraus. Konzepte wie Klassen wären zum Verständnis des ER-Modells nützlich.

- Auch nützlich (nicht obligatorisch): „Mathematische Grundlagen der Informatik und Konzepte der Modellierung“ oder „Einführung in die Wirtschaftsinformatik“.

Hören Sie mir aber zu, selbst wenn Sie glauben, das ER-Modell oder relationale Normalformen schon zu kennen.

Inhalt

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

Prüfung (1)

Klausur (geplant: 01.03. und 07.04.2021, 10⁰⁰–12⁰⁰):

- Es können drei DIN A4-Blätter mit beliebigen Notizen verwendet werden.

Sie dürfen Vorder- und Rückseite beschreiben oder bedrucken. Es ist eine gute Klausurvorbereitung, sich selbst das Wichtigste auf begrenztem Platz in eigener Strukturierung aufzuschreiben (inkl. „SQL Quick Reference“).

Die Termine der Klausuren sind noch nicht endgültig, achten Sie auf Änderungen! Melden Sie Überschneidungen frühzeitig!

- Praktische Anwendung, kein Auswendiglernen.

Ca. 40–70%: SQL-Anfragen.

Ca. 25–40%: DB-Entwurf (ER, Logischer Entwurf, BCNF).

Außerdem u.a. Logik, Relationale Algebra.

Alte Klausuren zu „Datenbanken I“ im Netz. Eventuell spezielle Probeklausur.

Prüfung (2)

- Geplant ist eine elektronische Klausur in Präsenz, aber wegen Corona ist alles unklar.

Falls Präsenzprüfungen nicht möglich sind, wäre für den ersten Termin eine Online-Prüfung zu Hause mit Videoüberwachung denkbar. Wenn es beim zweiten Termin deutlich weniger Teilnehmer sind, wäre vielleicht da eine elektronische Klausur in Präsenz möglich, ansonsten würden wir auch über mündliche Prüfungen als Videokonferenz nachdenken.

- Voraussichtlich gibt es bei der Klausur die Möglichkeit, SQL-Anfragen zu testen (mit PostgreSQL über Adminer).

Dann würden Syntaxfehler und offensichtlich falsche Ergebnisse zu härteren Punktabzügen führen, als wenn es keine Testmöglichkeit gibt. Es nützt aber auch nichts, Anfragen zu basteln, die nur im Testzustand mit irgendwelchen Tricks die verlangte Antwort liefern. Ich würde erwarten, dass Sie mit typischen Fehlermeldungen von PostgreSQL umgehen können. Für langes Trail&Error gibt es keine Zeit. Nutzen Sie Hausaufgaben+Übung!

Prüfung (3)

- Es ist nicht garantiert, dass man mit 50% der Punkte in der Klausur bestanden hat.

Garantiert ist das Bestehen mit 60% (die Grenze wird häufig bis in die Nähe der üblichen 50% gesenkt, aber durch die elektronische Klausur brauchen wir auch etwas Flexibilität).

Ab 95% der Punkte ist die 1.0 garantiert, aber auch diese Grenze wird häufig noch etwas gesenkt bis max. ca. 90%.

- Man kann die Klausur nur mitschreiben, wenn man angemeldet ist. Dazu ist die Modulanmeldung nötig.
- Man bekommt die Leistungspunkte für dieses Modul nur, wenn man auch die Studienleistung hat (Hausaufgaben, s.u.).

Man kann die Klausur ohne Studienleistung mitschreiben, aber das macht wenig Sinn: Die Hausaufgaben sollen ja auf die Klausur vorbereiten.

Studienleistung (1)

Hausaufgaben (mindestens 50% der Punkte):

- Bearbeitung einzeln.

Keine Gruppenarbeit, weil zu häufig nur „Namen mit draufgeschrieben“.
Bei zu ähnlichen Abgaben bekommen alle Beteiligten 0 Punkte.

- Ausgabe: Spätestens Montag 21:00 im Netz,
Abgabe: Montag 18:00 der nächsten Woche.

Ausgabe und Abgabe erfolgen über die Übungsplattform in StudIP.
Genauere Angaben zum Abgabeformat finden sich jeweils auf dem Übungsblatt: SQL-Anfragen bitte als „Plain Text“ (ASCII), so dass sie ggf. ausprobiert werden können. Ansonsten bitte PDF, kein Word.

- Syntaxfehler in SQL führen automatisch zu 0 Punkten!

Probieren Sie die Anfragen also selbst aus. Voraussichtlich werden Tests zur Verfügung gestellt (siehe Übung). Wir arbeiten an automatischer Korrektur.

Studienleistung (2)

Präsenzaufgaben (geben auch Hausaufgabenpunkte):

- In der Übung werden Aufgaben gestellt. Diese müssen Sie
 - erst einzeln bearbeiten (klausur-ähnlich),
 - dann in Kleingruppen (ca. 2–4 Teilnehmer) diskutieren,
Zufällig zusammengestellte „Breakout Rooms“ in der Videokonferenz-Software „BigBlueButton“.
 - und schließlich als Gruppe gemeinsam abgeben.
In einem Hausaufgaben-Ordner von StudIP hochladen. Alle Teilnehmer der Gruppe bekommen die gleiche Punktzahl. Ich notiere mir gelegentlich die Zusammensetzung der Gruppen. Falls auf abgegebenen Lösungen Teilnehmer stehen, die nicht anwesend waren, gibt es 0 Punkte.
- Diese Aufgaben fließen in die Gesamtsumme der Hausaufgabenpunkte mit ein (ca. 20%).

Studienleistung (3)

Vorrechnen, aktive Mitarbeit:

- Sie müssen jederzeit bereit sein, von Ihnen abgegebene Hausaufgaben vorzurechnen.

Auch aus falschen Lösungen kann man viel lernen. Wir machen alle manchmal Fehler. Bei Plagiatsverdacht müssen Sie unbedingt einen Kurzvortrag zu Ihren Aufgaben halten, sonst gibt es keine Studienleistung! Aktive Mitarbeit und Kurzvorträge stehen in der Modulbeschreibung.

- Die Übung lebt von aktiver Diskussion:
Stellen Sie Fragen! Bringen Sie alternative Lösungen ein!
- Falls sich Plagiatsverdachts-Fälle häufen, oder es in den Übungsgruppen sehr still ist, wird die verlangte „aktive Mitarbeit“ als Teil der Studienleistung erzwungen.

Ansonsten ist es kein Problem, einige wenige Male zu fehlen. Man verliert zwar die Punkte für die Präsenzübungen, aber man benötigt nur 50% aller Punkte.

Studienleistung (4)

Bei Unklarheiten mit der Korrektur:

- Eventuelle Fragen zur Korrektur von Hausaufgaben/Klausur 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 nicht.

- Bitte bei den Hausaufgabenpunkten nicht feilschen (boolesche Entscheidung, kein Einfluss auf Note).

Wenn Sie am Ende knapp unter 50% haben, können Sie noch fragen. Krasse Fehlbewertungen sollen natürlich sofort korrigiert werden.

Inhalt

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

Vorlesungs-Materialien

- Das alte Skript zu „Datenbanken I“ hatte ca. 1100 Folien.

Es ist verfügbar unter folgender Adresse:

<http://www.informatik.uni-halle.de/~brass/db17/>

- Im letzten Jahr gab es 967 Folien (fast ein Skript).

Ein Teil sind nicht prüfungsrelevante Anmerkungen zu speziellen DBMS.

Ein Teil sind aber auch Hinweise zu Fehlern und Missverständnissen, die mir aufgefallen sind. Es gibt jedes Jahr eine Überarbeitung.

- Die Folien werden kurz vor der Vorlesung auf folgende Webseite gestellt:

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

- Webseite enthält auch: Alte Klausuren, Links.
- Forum in StudIP (z.B. für Fragen zu Hausaufgaben).

Bücher sind wichtig!

- Wir sind an einer Uni.

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 Datenbanken sind viel mehr als nur SQL, und auch da kann man sich ja fragen, ob es nicht bessere Anfragesprachen geben könnte.

- Man braucht mehrere Quellen.

Nur aus unterschiedlichen Blickwinkeln sieht man dreidimensional.

- Ich empfehle dringend, ein Lehrbuch zu beschaffen, und in Teilen zu lesen (reicht oft auch für Master-Vorlesungen).

Die Modulbeschreibung enthält hierfür 45–60 Stunden Selbststudium.

Folien sind kein Lehrbuch, und unterschiedliche Lernformen ergänzen sich.

Je nach Vorwissen/Interessen sind unterschiedliche Bücher ansprechend.

- Es sind auch Online-Quellen auf der Webseite verlinkt.

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. 6. Aufl.
MITP Bonn, 2018, ISBN 978-3958457768, 800 Seiten, 39.99 €.
- Gottfried Vossen:
Datenbankmodelle, Datenbanksprachen und
Datenbankmanagementsysteme. 5. Aufl.
Oldenbourg, 2008, 3-486-27574-7, 842 Seiten, 64.95 €.

Lehrbücher (2)

- 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 — Übersetzung fragwürdig):

Pearson, 3. Aufl., 2009, ISBN 386894012X, 560 Seiten, 34.95 €.

- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom:
Database Systems: The Complete Book, 2nd Ed.

Prentice Hall, 2008/13, ISBN 978-1292024479, 1203/1140 pages, 61.20 €.

- Raghu Ramakrishnan, Johannes Gehrke:
Database Management Systems, 3rd Ed.

McGraw Hill, 2002, ISBN 0-07-123151-X, 1065 Seiten, 65.99 €.

Lehrbücher (3)

- 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 €.

- Georg Lausen:
Datenbanken. Grundlagen und XML-Technologien.

Spektrum/Elsevier, 2005, ISBN 3-8274-1488-1, 286 Seiten, nur noch
gebraucht.

Inhalt

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

Software: PostgreSQL

- Verbreitetes Open Source Datenbank-Managementsystem (DBMS) mit umfangreicher SQL-Unterstützung.

Postgres war Vorreiter der objektrelationalen Technologie.

Die Hausaufgaben sollten allerdings möglichst in Standard-SQL gelöst werden.

Bei der Klausur kann man voraussichtlich Anfragen mit PostgreSQL testen.

- [<https://www.postgresql.org/>]

Läuft unter Linux, Windows, MacOS, BSD Unix, Solaris.

- In der Übung wird ein Docker-Container zur Verfügung gestellt (mit Adminer Web-Schnittstelle).
- Wir bieten auch eine Adminer Web-Schnittstelle zum Üben von SQL ohne eigene Installation.

[<https://www.adminer.org/>] [<http://dbs2.informatik.uni-halle.de/edb>]

Software: MySQL/MariaDB

- Verbreitetes Open Source Datenbank-Managementsystem
 - Vermutlich verbreiteter als PostgreSQL, ist auch bei vielen Web-Hosting Angeboten mit dabei. LAMP (Linux, Apache, MySQL, PHP) ist typisch für dynamische Webseiten. Die Variante XAMPP gibt es auch für Windows.
- Das SQL war früher wenig standard-konform, das hat sich aber deutlich gebessert.
- MySQL wurde zuerst von Sun gekauft, dann wurde Sun vom kommerziellen Datenbank-Anbieter Oracle gekauft.
 - MySQL und auch MariaDB sind etwas kommerzieller als PostgreSQL.
- Daraufhin hat sich MariaDB abgespalten.
- [<https://mariadb.org/>] [<https://www.mysql.com/de/>]

Software: 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>]

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.

Software: Microsoft SQL Server

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

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.

Software: 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: Marktanteile 2011 (Gartner)

DBMS-Markt (Relationale Systeme) 2011 [Gartner]

Anbieter	Marktanteil	Änderung zu 2010
Oracle	48.8%	+0.6%
IBM	20.2%	-0.5%
Microsoft	17.0%	-0.6%
SAP/Sybase	4.6%	+1.0%
Teradata	3.7%	+0.1%
Andere	5.8%	-0.5%

Marktgröße (2011): 24 Mrd. US-Dollar (+16.3%).

[<http://itknowledgeexchange.techtarget.com/>]

[[eye-on-oracle/oracle-the-clear-leader-in-24-billion-rdbms-market/](http://eye-on-oracle.com/oracle-the-clear-leader-in-24-billion-rdbms-market/)]

Zahlen: Einkünfte der Firmen aus DB-Geschäft, nicht die Anzahl Lizenzen.

DB Software: Marktanteile 2017 (IDC)

Relational Database Vendor Share 2017 [IDC]

Anbieter	Einnahmen	Marktanteil
Oracle	14,0 Mrd. \$	42%
Microsoft	8,0 Mrd. \$	24%
IBM	4,2 Mrd. \$	13%
SAP/Sybase	2,3 Mrd. \$	7%
Teradata	1,0 Mrd. \$	3%
Andere	3,7 Mrd. \$	11%

Marktgröße (2017): 33 Mrd. US-Dollar.

Zitiert nach: Database Software Market: The Long Awaited Shake-up
(Jason Adler, Billy Fitzsimmons, Sebastian Naji - William Blair
Equity Research)

[<https://blocksandfiles.com/wp-content/uploads/2019/03//>]

[Database-Software-Market-White-Paper.pdf]

DB Software: Popularität im Netz

Rang	System	Art	Punkte
1.	Oracle	Relational DBMS	1369
2.	MySQL	Relational DBMS	1256
3.	Microsoft SQL Server	Relational DBMS	1043
4.	PostgreSQL	Relational DBMS	542
5.	MongoDB	Document Store	448
6.	DB2	Relational DBMS	162
7.	Elasticsearch	Suchmaschine	154
8.	Redis	Key-Value Store	153
9.	SQLite	Relational DBMS	125
10.	Cassandra	Wide Column Store	119

[<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	53.5%
2.	PostgreSQL	38.5%
3.	Microsoft SQL Server	34.8%
4.	SQLite	30.6%
5.	MongoDB	26.7%
6.	Redis	20.5%
7.	MariaDB	16.9%
8.	Oracle	16.3%
9.	Elasticsearch	15.5%
10.	Firebase	13.9%

Quelle: Stack Overflow Developer Survey 2020

[<https://insights.stackoverflow.com/survey/2020#technology>]

Unter „Programming, Scripting and Markup Languages“ steht SQL (56.9%) an dritter Stelle hinter JavaScript (69.7%) und HTML/CSS (62.4%).

Software für die Übungen

- In den Übungen wird in erster Linie PostgreSQL verwendet.
- Über die Adminer-Schnittstelle werden zusätzlich MariaDB und voraussichtlich Oracle zur Verfügung gestellt.
- 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.
- Falls Sie sich für Datenbanken besonders interessieren, wäre wohl vernünftig, sich auch eine Express Edition eines kommerziellen Systems zu installieren.

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: Execute SQL Online (SQLite)
[https://www.tutorialspoint.com/execute_sql_online.php]

Inhalt

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

Aktive Mitarbeit, Verbesserung der Lehre

- Fragen sind sehr willkommen!

In der Präsenz-Version der Vorlesung und vor allem den Übungen.
Eventuell im Forum in StudIP, nur selten per EMail (ich bekomme sehr viele EMails, und schreibe teils lange Antworten, was mir viel Zeit kostet).

- Vorschläge zur Verbesserung der Lehrveranstaltung sind willkommen.

Das gern per EMail. Z.B. Hinweise auf Tippfehler, veraltete Angaben, unklare Stellen, u.s.w. Auch Hinweise auf nützliche Webseiten.

- Beteiligen Sie sich in den Übungen mit Wortbeiträgen.

Im Notfall muss ich irgendwen aufrufen, der sich nicht gemeldet hat, und die ganze Sache wird zäh und für alle (auch mich) wenig erfreulich.

- Geben Sie bei der Videokonferenz Ihren echten Namen an.

Bedeutung praktischer Übung

- Ich möchte, dass Sie am Ende dieser Vorlesung SQL problemlos beherrschen, und zwar nicht nur simple Anfragen, sondern auch komplexere.

Z.B. mit NOT EXISTS, GROUP BY, UNION, CASE, LEFT JOIN.

- Neben theoretischem Verständnis braucht man dazu auch praktische Erfahrung mit einem konkreten DBMS.

Wie bei jeder Programmiersprache. Zum Teil ist es Handwerk. Man muss üben.

- Klären Sie Fehler vollständig auf. Fehler sind nützlich!

Kopieren Sie sich Anfrage und Fehlermeldung bzw. Ergebnis und fragen Sie in der nächsten Übung, wenn Sie es nicht selbst aufklären können.

Wenn jemand in der Klausur mit einer simplen Fehlermeldung von PostgreSQL nicht umgehen kann, die er eigentlich schon viele Male gesehen haben muss, frage ich mich doch, ob er die Hausaufgaben selbst gemacht hat.

Ein Erfolgsrezept

- Ich habe den Studenten mit dem besten Klausurergebnis im WS 19/20 (110%) nach seinem Erfolgsrezept gefragt:
 - „Das praktische Lösen der Aufgaben hat gefühlt am meisten zu meinem Verständnis beigetragen.“
 - „Dabei habe ich mir die resultierenden Tabellen stets visuell vorgestellt — ein bisschen wie Dominosteine, die an den passenden Stellen zusammengesetzt, ausgesiebt und sortiert werden.“
 - „Generell glaube ich aber, dass mein Erfolgsrezept hauptsächlich in der Begeisterung fürs Knobeln liegt. Zur Not mache ich die gleiche Aufgabe ein drittes Mal, wenn mir ein kürzerer oder schönerer Lösungsweg einfällt.“
- Jeder Mensch lernt natürlich etwas anders (z.B. auch Logik).