

Datenbank-Programmierung

Kapitel 0: Organisatorisches

Prof. Dr. Stefan Brass

Martin-Luther-Universität Halle-Wittenberg

Sommersemester 2019

<http://www.informatik.uni-halle.de/~brass/dbp19/>

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.
 - Mindestens für Wirtschaftsinformatiker werden auch diese Vorlesung und “Websuche und Information Retrieval” (Prof. Hagen) als äquivalent anerkannt. Dadurch ist es möglich, die 10 LP im Wintersemester zu erbringen.
- 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.

Themen

- Tabellen-Deklarationen in der Praxis
- Index-Strukturen (kurz)
- Updates, Transaktionen, Mehrbenutzerbetrieb
- SQL Skripte
- JDBC-Schnittstelle (Datenbank-Zugriff aus Java)
- Sicherheits-Aspekte (Zugriffsrechte, Datenschutz, Sichten)
- Trigger, Serverseitige Programmierung
- Data Warehouses/OLAP, Fortgeschrittenes SQL
- Datalog, Rekursion, Ausdrucksfähigkeit von Anfragesprachen
- Ggf. Web-Datenbank-Programmierung (HTML, HTTP, PHP)

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).
- Mit SQL kann man auch relativ komplexe Auswertungen der Daten kompakt aufschreiben.
- 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!

Motivation (2)

- 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.

- Selbst wenn die für die Anwendung benötigten Datenmengen nicht groß sind, hat man mit den Tabellen eine einfache und übersichtliche Struktur.

- 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 [dbp19] beginnen, möglichst aussagefähig.

- Büro: Von-Seckendorff-Platz 1, Raum 313
- Telefon: 0345/55-24740
- Sprechstunde: Montags, 12⁰⁰–13⁰⁰
- Frühere Unis: Braunschweig, Dortmund, Hannover, Hildesheim, Pittsburgh, Gießen, Clausthal.
- Oracle8 Certified Database Administrator (aktuell: 12C→18C)
- 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 (in StudIP/ILIAS), funktioniert aber nicht immer.
Fragen sind willkommen! Ihre Stimme ist auf dem Video fast nicht zu hören.

Es gibt ausführliche Folien (Skript), aber:

- In der Vorlesung gibt es mehr Erläuterungen, Antworten auf Fragen, Betonungen (was ist besonders wichtig?).

Beispiele, Aufgaben, Diskussionen, Fragen. Wichtige Ankündigungen.
Zeitaufwand für selbst nacharbeiten eher höher. Wenn man nicht regelmäßig dranbleibt, wird man schnell “abgehängt”. Professor frustriert, wenn keine Hörer.

- Ggf. gelegentlich bepunktete Präsenzaufgaben (s.u.)

Zeit und Ort (2)

Übung (2 SWS):

- Zwei Gruppen (Anmeldung: StudIP, erste ist voll):

| Nr | Tag | Zeit | Raum | Beginn |
|----|--------|------------------------------------|--------------|--------|
| 1 | Montag | 12 ¹⁵ –13 ⁴⁵ | 0.04 (+Pool) | 08.04. |
| 2 | Montag | 14 ¹⁵ –15 ⁴⁵ | 0.04 (+Pool) | 08.04. |

Gruppe 1: 35 Teilnehmer (passt nicht in Pool), Gruppe 2: 18.

- Die Übungen sind zum Teil im Seminarraum (theoretisch), zum Teil in einem Rechner-Pool (praktisch).
- Poolbelegung:

[<https://nirvana.informatik.uni-halle.de/agendav/>]

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 |

Die Übungszeit und “Lösen von Hausaufgaben” enthalten Anteile für praktische Übung am Rechner. “Hands-on Experience” ist nötig!

“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. 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)

- 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.
- Wenn Sie aber schon zu “Datenbanken I” angemeldet sind, können Sie auch die kombinierte Prüfung zu beiden Teilen machen, ohne sich zu diesem Modul getrennt anzumelden.
 - Alles andere erfordert Anträge an den Prüfungsausschuss, die in der Übergangszeit wohlwollend betrachtet werden.

Inhalt

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

Prüfung (1)

Klausur (geplant: 22.07. und 23.09.2019, 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!

- Zwei Varianten:
 - Nur für diese Vorlesung.
 - Zusammen mit "Einf. in Datenbanken" (für "DB I").
- Bei zu wenig Teilnehmern: Mündliche Prüfung.

Prüfung (2)

Zur Klausur:

- Es ist nicht garantiert, dass man mit 50% der Punkte in der Klausur bestanden hat.
 - Garantiert ist das Bestehen mit 60%. Falls die Noten nach Ihrer Prüfungsordnung in Zahlen von 0 bis 100 ausgedrückt werden, muss die Prozentzahl der Punkte nicht genau der Note entsprechen.
 - 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:

- Sie benötigen 50% der Punkte für die Studienleistung.
- Bearbeitung einzeln.

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

- Ausgabe: Spätestens Montag 12:15 im Netz,
Abgabe: Bis Montag 10:15 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 führen automatisch zu 0 Punkten!

Wir wollen, dass Sie Anfragen und Programme wirklich ausprobieren.

Studienleistung (2)

Präsenzaufgaben (geben auch Hausaufgabenpunkte):

- In Übung und gelegentlich auch Vorlesung werden Aufgaben gestellt, die Sie unter klausur-ähnlichen Bedingungen lösen müssen.

Ziel ist einerseits eine bessere Vorbereitung auf die Klausur, andererseits eine Reduktion des Effektes von abgeschriebenen Hausaufgaben.

- Diese Aufgaben sind abzugeben, werden bepunktet und fließen in die Gesamtsumme mit ein (ca. 10%).

Wenn Sie nicht da sind, können Sie für die Präsenzaufgaben keine Punkte bekommen. Da Sie nur 50% der Punkte benötigen, und die Präsenzaufgaben nur ein Teil der Aufgaben sind (neben Hausaufgaben), sollte es kein Problem sein, wenn sie einige wenige Male fehlen.

Studienleistung (3)

Aktive Mitarbeit/Vorrechnen:

- Sie müssen jederzeit bereit sein, von Ihnen abgegebene Hausaufgaben vorzurechnen (Übung, Vorlesung, Flur, ...).
- Aktive Mitarbeit in den Übungen (inkl. Kurzvortrag über Hausaufgaben) ist Teil der verlangten Studienleistung.

Wie strikt das im Einzelfall gehandhabt wird, liegt an Plagiatsvorfällen, Aktivität der Gruppe und des Einzelnen (z.B. Vorlesung/prakt. Übung). Es ist sicher kein Problem, wenn Sie drei Mal fehlen würden. Darüber hinaus werden wir ggf. auch die verpassten Präsenzaufgaben verlangen.

- Die Übung lebt von aktiver Diskussion:
Stellen Sie Fragen! Bringen Sie alternative Lösungen ein!

Sie können natürlich auch sehr gern in der Vorlesung Fragen stellen.

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 Schlussbemerkungen

Vorlesungs-Materialien

- Die Folien für diese Vorlesung werden neu erstellt.
Einige Kapitel entstammen dem alten Skript zu "Datenbanken I":
<http://www.informatik.uni-halle.de/~brass/db17/>
- Sie werden kurz vor der Vorlesung auf die Webseite gestellt:
<http://www.informatik.uni-halle.de/~brass/dbp19/>
Formate: (1) Mit Fortschrittsindikator und Hintergrund (für Beamer).
(2) Schlichter, zum Drucken.
Bei eventuellen Problemen oder Verbesserungsvorschlägen bitte melden.
- Webseite im Aufbau.
Wenn Sie nützliche Seiten zum Thema der Vorlesung im Netz finden,
schicken Sie mir bitte eine EMail mit der Webadresse.
- Forum in StudIP.
Hier bitte auch Fragen zu Hausaufgaben.

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? Wir 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.

Die Modulbeschreibung enthält hierfür 45 Stunden Selbststudium. 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.

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

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

- Empfehlung: Installieren Sie sich PostgreSQL auf Ihrem eigenen Rechner.

Wir bieten aber auch ein Remote Login (ssh) auf einem Rechner mit installiertem PostgreSQL und MariaDB. Unter Windows benötigen Sie hierfür PuTTY: [<https://www.chiark.greenend.org.uk/~sgtatham/putty/>]

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

[<https://www.adminer.org/>] Das reicht aber nicht für alle Aufgaben.

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 2017 Express kostet nichts:

[<https://www.microsoft.com/de-de/sql-server/sql-server-2017>]

[<https://www.microsoft.com/de-de/sql-server/sql-server-downloads>]

[<https://www.microsoft.com/de-de/sql-server/sql-server-2017-editions>]

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), der hatte aber die Sprache QUEL und nicht SQL.

- Es gab früher eine kostenlose Express-C Edition.
Ich kann sie nicht mehr finden.

Für Entwickler gibt es diese Versionen, die offenbar kostenlos sind:

[<https://www.ibm.com/us-en/marketplace/ibm-db2-direct-and-developer-editions/purchase>]

DB Software: Marktanteile 2011

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/]

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

DB Software: Populartität im Netz

| Rang | System | Art | Punkte |
|------|----------------------|-----------------|--------|
| 1. | Oracle | Relational DBMS | 1279 |
| 2. | MySQL | Relational DBMS | 1198 |
| 3. | Microsoft SQL Server | Relational DBMS | 1048 |
| 4. | PostgreSQL | Relational DBMS | 470 |
| 5. | MongoDB | Document Store | 401 |
| 6. | DB2 | Relational DBMS | 177 |
| 7. | Microsoft Access | Relational DBMS | 146 |
| 8. | Redis | Key-Value Store | 146 |
| 9. | Elasticsearch | Suchmaschine | 143 |
| 10. | SQLite | Relational DBMS | 125 |

[<https://db-engines.com/de/ranking>] ... 11. Cassandra, 12. MariaDB, 14. Teradata.

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.

Software für die Übungen

- In den Übungen wird in erster Linie PostgreSQL verwendet.
- Wir werden auch MariaDB zur Verfügung stellen.
- JDBC ist eine Standard-Schnittstelle, und normale SQL-Anfragen sollten in jedem System funktionieren.

In dem Kapitel über fortgeschrittenes SQL gehen wir allerdings 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.

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.

Forschung/Bachelor-Arbeiten

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).