

# Einführung in Datenbanken

---

## Kapitel 0: Organisatorisches

Prof. Dr. Stefan Brass

Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2019/20

<http://www.informatik.uni-halle.de/~brass/db19/>

# Inhalt

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

# 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.

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. Sollten Sie sich im Master in Datenbanken spezialisieren wollen, sollten Sie das hören. Wenn Sie im Beruf DB-Anwendungssysteme entwickeln wollen, auch.
- Für Wirtschaftsinformatiker werden beide Teile Pflicht.

# 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 (kurz)

# Fortsetzung: Datenbank-Programmierung

## Geplante Inhalte für das Modul im Sommer:

- Fortgeschrittenes/Modernes SQL  
Insbesondere für Data Warehouse Anwendungen
- Transaktionen (Recovery, Mehrbenutzerbetrieb)  
Synchronisation paralleler Zugriffe auf die Datenbank
- Datenbank-Schnittstellen aus Programmiersprachen, JDBC
- Integritäts-Überwachung, Serverseitige Programmierung, Trigger
- Zugriffsrechte, Datenschutz, Sichten
- Datalog, Rekursion, Ausdrucksfähigkeit von Anfragesprachen
- 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)

## Dozent: Prof. Dr. Stefan Brass

- Email: [brass@informatik.uni-halle.de](mailto:brass@informatik.uni-halle.de)

Betreff-Zeile sollte mit [db19] beginnen, möglichst aussagefähig.

- Büro: Von-Seckendorff-Platz 1, Raum 313
- Telefon: 0345/55-24740
- Sprechstunde: Donnerstags, 12<sup>00</sup>–13<sup>00</sup>
- 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)

## Übungsleiter: PD Dr. Alexander Hinneburg

- Büro: Von-Seckendorff-Platz 1, Raum 314
- Telefon: 0345/55-24732
- Email: [hinnebur@informatik.uni-halle.de](mailto: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](mailto:vahrenho@informatik.uni-halle.de)

# Zeit und Ort (1)

## Vorlesung (2 SWS):

- Montags, 10<sup>15</sup>–11<sup>45</sup>, Raum 3.28.

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.
- Einzelne bepunktete Präzenzaufgaben (s.u.) sind möglich.

# Zeit und Ort (2)

## Tafelübung (2 SWS):

- Zwei Gruppen (Anmeldung: StudIP): [Am 19.11.2019: IT<sup>2</sup>]

Nr	Tag	Zeit	Raum	Beginn
1	Dienstag	14 <sup>15</sup> –15 <sup>45</sup>	3.04	15.10.
2	Dienstag	16 <sup>15</sup> –17 <sup>45</sup>	1.30	15.10.

Gruppe 1: 93 Teilnehmer, Gruppe 2: 50 (Raum hat 30 Plätze).

## Freiwillige praktische Übungszeit am Rechner (Hausaufgaben):

- Pool reserviert (ca. 25 Plätze), leider wenig Betreuung:

Tag	Zeit	Pool
Montags	12 <sup>00</sup> –14 <sup>00</sup>	3.02 (Multimedia-Pool)

Ich bin 12–13 in meinem Büro. Ggf. am Anfang von SQL auch Tutor im Pool.

# Zeitliche Belastung

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

Lernform	SWS	Stunden
Vorlesung	2	30
Tafelübung	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. 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 setzten 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: 24.02. und 23.03.2020, 10<sup>00</sup>–12<sup>00</sup>):

- 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.

In diesem Jahr ist eine elektronische Prüfung geplant (es ist aber unklar, in wie weit man SQL-Anfragen bei der Klausur testen kann — ggf. gar nicht).

# 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% (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 Dienstag 12:00 im Netz,  
Abgabe: Dienstag 12:30 der nächsten Woche.

Abgabe 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 (und ggf. 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. 10–20%).

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. Es gibt keine Anwesenheitspflicht.

# Studienleistung (3)

## Vorrechnen:

- Sie müssen jederzeit bereit sein, von Ihnen abgegebene Hausaufgaben vorzurechnen (Übung, Vorlesung, Flur, ...).

Da Sie die Aufgaben fast direkt vor der Übung abgeben müssen, wissen weder der Übungsleiter noch Sie, ob die Lösung korrekt ist. Aber auch aus falschen Lösungen kann man viel lernen!

- Bei Plagiatsverdacht müssen Sie 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!

# 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 915 Folien (fast ein Skript).

Ich arbeite dieses Jahr daran, Duplizierungen zu entfernen, und Folien zu markieren, die nicht prüfungsrelevante Informationen zu speziellen DBMS enthalten. Es wird also nochmals eine Überarbeitung (und Kürzung) geben.

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

<http://www.informatik.uni-halle.de/~brass/db19/>

- 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 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. 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 — Übersetzung fragwürdig):  
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 €.

# Inhalt

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

# Software: PostgreSQL

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

Postgres war Vorreiter der objektrelationalen Technologie.

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

- [<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 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).

- 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/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: Populartität im Netz

Rang	System	Art	Punkte
1.	Oracle	Relational DBMS	1356
2.	MySQL	Relational DBMS	1283
3.	Microsoft SQL Server	Relational DBMS	1095
4.	PostgreSQL	Relational DBMS	484
5.	MongoDB	Document Store	412
6.	DB2	Relational DBMS	171
7.	Elasticsearch	Suchmaschine	150
8.	Redis	Key-Value Store	143
9.	Microsoft Access	Relational DBMS	131
10.	Cassandra	Wide Column Store	123

[<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	52.0%
2.	PostgreSQL	36.3%
3.	Microsoft SQL Server	34.4%
4.	SQLite	30.9%
5.	MongoDB	26.0%
6.	Redis	20.6%
7.	MariaDB	16.5%
8.	Oracle	16.4%
9.	Elasticsearch	15.9%
10.	Firebase	12.3%

Quelle: Stack Overflow Developer Survey 2019

[<https://insights.stackoverflow.com/survey/2019#technology>]

Unter "Programming, Scripting and Markup Languages" steht SQL (56.5%) an dritter Stelle hinter JavaScript (69.7%) und HTML/CSS (63.1%).



# 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.
- JDODDLE: Online SQL Editor (SQLite)  
[<https://www.jdoodle.com/execute-sql-online/>]
- JDODDLE: Online MySQL Terminal (MySQL)  
[<https://www.jdoodle.com/online-mysql-terminal/>]
- codingground: Exceute SQL Online (SQLite)  
[[https://www.tutorialspoint.com/execute\\_sql\\_online.php](https://www.tutorialspoint.com/execute_sql_online.php)]

# Inhalt

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

# Verbesserung der Lehre

- Diese Vorlesung soll kein Monolog werden:

**Fragen sind sehr willkommen.**

Vorschläge zur Verbesserung der Vorlesung sind sehr willkommen!  
Z.B. auch Hinweise zu guten Web-Ressourcen.

- Ich möchte dieses Semester in der Vorlesung mehr Aufgaben, Gruppenarbeit etc. machen.

Falls sich nicht mindestens ein Drittel der Teilnehmer melden, rufe ich auch Teilnehmer auf, die sich nicht melden. Sie dürfen die Frage aber "schieben", dann frage ich jemand anders.

- Bei manchen Teilen werde ich Sie bitten, dass Sie sich den Stoff anhand Folien selbst erarbeiten.

Wir werden dann nur noch über Fragen sprechen.

# 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).
  - Wenn Sie zu spät kommen oder früher gehen müssen, setzen Sie sich möglichst an den Rand.
  - Notebooks sollten während der Vorlesung nur die Folien anzeigen (eventuell Notizen, DB).
  - Mobiltelefone bitte auch (wenn überhaupt) nur für vorlesungsrelevante Tätigkeiten verwenden.
- Nur anwesend sein, reicht nicht. Mitdenken!