



合肥大學
HEFEI UNIVERSITY



Datenbanken

33. Auswahl des Datenmodells

Thomas Weise (汤卫思)
tweise@hfuu.edu.cn

Institute of Applied Optimization (IAO)
School of Artificial Intelligence and Big Data
Hefei University
Hefei, Anhui, China

应用优化研究所
人工智能与大数据学院
合肥大学
中国安徽省合肥市

Databases



Dies ist ein Kurs über Datenbanken an der Universität Hefei (合肥大学).

Die Webseite mit dem Lehrmaterial dieses Kurses ist <https://thomasweise.github.io/databases> (siehe auch den QR-Code unten rechts). Dort können Sie das Kursbuch (in Englisch) und diese Slides finden. Das Repository mit den Beispielen finden Sie unter <https://github.com/thomasWeise/databasesCode>.



Outline



1. Einleitung
2. Datenmodelle
3. Zusammenfassung





Einleitung



Einleitung



- Zwischen dem Entwurf des konzeptuellen Schemas und dem Entwurf des logischen Schemas müssen wir ein *Datenmodell* für unsere Datenbank auswählen⁷⁴.

- Zwischen dem Entwurf des konzeptuellen Schemas und dem Entwurf des logischen Schemas müssen wir ein *Datenmodell* für unsere Datenbank auswählen⁷⁴.

Definition: Datenmodell

Das *Datenmodell* (EN: *data model*) spezifiziert die Notation und Sprache zum Definieren von Datentypen und um auf die Datenbank zuzugreifen und diese zu verändern.

- Zwischen dem Entwurf des konzeptuellen Schemas und dem Entwurf des logischen Schemas müssen wir ein *Datenmodell* für unsere Datenbank auswählen⁷⁴.

Definition: Datenmodell

Das *Datenmodell* (EN: *data model*) spezifiziert die Notation und Sprache zum Definieren von Datentypen und um auf die Datenbank zuzugreifen und diese zu verändern.

- Es gibt viele wichtige Datenmodelle.

- Zwischen dem Entwurf des konzeptuellen Schemas und dem Entwurf des logischen Schemas müssen wir ein *Datenmodell* für unsere Datenbank auswählen⁷⁴.

Definition: Datenmodell

Das *Datenmodell* (EN: *data model*) spezifiziert die Notation und Sprache zum Definieren von Datentypen und um auf die Datenbank zuzugreifen und diese zu verändern.

- Es gibt viele wichtige Datenmodelle.
- Hier schauen wir uns einige davon an.



Datenmodelle



Flache Unverbundene Tabellen



- Wir könnten unsere Daten als einzelne oder nur schwach verbundene Tabellen darstellen.

Flache Unverbundene Tabellen



- Wir könnten unsere Daten als einzelne oder nur schwach verbundene Tabellen darstellen.
- Diese könnten in Formaten wie Comma-Separated Values (CSV)⁷⁶ oder in den Dateiformaten von Microsoft Excel^{12,37,53} oder LibreOffice Calc^{55,75} gespeichert werden.

Flache Unverbundene Tabellen



- Wir könnten unsere Daten als einzelne oder nur schwach verbundene Tabellen darstellen.
- Diese könnten in Formaten wie Comma-Separated Values (CSV)⁷⁶ oder in den Dateiformaten von Microsoft Excel^{12,37,53} oder LibreOffice Calc^{55,75} gespeichert werden.
- Diese Formate können tausende gleich-strukturierte Datensätze speichern und Berechnungen auf diesen ausführen.

Flache Unverbundene Tabellen



- Wir könnten unsere Daten als einzelne oder nur schwach verbundene Tabellen darstellen.
- Diese könnten in Formaten wie Comma-Separated Values (CSV)⁷⁶ oder in den Dateiformaten von Microsoft Excel^{12,37,53} oder LibreOffice Calc^{55,75} gespeichert werden.
- Diese Formate können tausende gleich-strukturierte Datensätze speichern und Berechnungen auf diesen ausführen.
- Sie sind ungeeignet, wenn Datensätze verschiedener Typen in Beziehung stehen und Einschränkungen auf diese Beziehungen oder die Daten festgelegt werden.

Flache Unverbundene Tabellen



- Wir könnten unsere Daten als einzelne oder nur schwach verbundene Tabellen darstellen.
- Diese könnten in Formaten wie Comma-Separated Values (CSV)⁷⁶ oder in den Dateiformaten von Microsoft Excel^{12,37,53} oder LibreOffice Calc^{55,75} gespeichert werden.
- Diese Formate können tausende gleich-strukturierte Datensätze speichern und Berechnungen auf diesen ausführen.
- Sie sind ungeeignet, wenn Datensätze verschiedener Typen in Beziehung stehen und Einschränkungen auf diese Beziehungen oder die Daten festgelegt werden.
- Sie sind auch nicht wirklich geeignet in Szenarios wo mehrere Benutzer am selben Dokument arbeiten.

Flache Unverbundene Tabellen



- Wir könnten unsere Daten als einzelne oder nur schwach verbundene Tabellen darstellen.
- Diese könnten in Formaten wie Comma-Separated Values (CSV)⁷⁶ oder in den Dateiformaten von Microsoft Excel^{12,37,53} oder LibreOffice Calc^{55,75} gespeichert werden.
- Diese Formate können tausende gleich-strukturierte Datensätze speichern und Berechnungen auf diesen ausführen.
- Sie sind ungeeignet, wenn Datensätze verschiedener Typen in Beziehung stehen und Einschränkungen auf diese Beziehungen oder die Daten festgelegt werden.
- Sie sind auch nicht wirklich geeignet in Szenarios wo mehrere Benutzer am selben Dokument arbeiten.
- Für unser Lehre-Management-System sind sie also nicht geeignet.

Hierarchische Dokumente für Unverbundene Daten



- Eine andere Möglichkeit sind hierarchisch strukturierte einzelne Dokumente, in Form von wie Extensible Markup Language (XML)^{17,24,50}, JavaScript Object Notation (JSON)^{16,86} oder YAML Ain't Markup Language™ (YAML)^{23,31,50}.

Hierarchische Dokumente für Unverbundene Daten



- Eine andere Möglichkeit sind hierarchisch strukturierte einzelne Dokumente, in Formate wie Extensible Markup Language (XML)^{17,24,50}, JavaScript Object Notation (JSON)^{16,86} oder YAML Ain't Markup Language™ (YAML)^{23,31,50}.
- Diese Formate können tausende Datensätze speichern, die entweder gleich oder verschiedenartig strukturiert sind.

Hierarchische Dokumente für Unverbundene Daten



- Eine andere Möglichkeit sind hierarchisch strukturierte einzelne Dokumente, in Formate wie Extensible Markup Language (XML)^{17,24,50}, JavaScript Object Notation (JSON)^{16,86} oder YAML Ain't Markup Language™ (YAML)^{23,31,50}.
- Diese Formate können tausende Datensätze speichern, die entweder gleich oder verschiedenartig strukturiert sind.
- Sie sind ungeeignet, wenn Datensätze von verschiedenen Typen in Beziehungen stehen und Einschränkungen für die Beziehungen gelten.

Hierarchische Dokumente für Unverbundene Daten



- Eine andere Möglichkeit sind hierarchisch strukturierte einzelne Dokumente, in Formate wie Extensible Markup Language (XML)^{17,24,50}, JavaScript Object Notation (JSON)^{16,86} oder YAML Ain't Markup Language™ (YAML)^{23,31,50}.
- Diese Formate können tausende Datensätze speichern, die entweder gleich oder verschiedenartig strukturiert sind.
- Sie sind ungeeignet, wenn Datensätze von verschiedenen Typen in Beziehungen stehen und Einschränkungen für die Beziehungen gelten.
- Sie sind auch nicht gut geeignet, wenn mehrere Benutzer gleichzeitig an den selben Daten arbeiten.

Hierarchische Dokumente für Unverbundene Daten



- Eine andere Möglichkeit sind hierarchisch strukturierte einzelne Dokumente, in Formate wie Extensible Markup Language (XML)^{17,24,50}, JavaScript Object Notation (JSON)^{16,86} oder YAML Ain't Markup Language™ (YAML)^{23,31,50}.
- Diese Formate können tausende Datensätze speichern, die entweder gleich oder verschiedenartig strukturiert sind.
- Sie sind ungeeignet, wenn Datensätze von verschiedenen Typen in Beziehungen stehen und Einschränkungen für die Beziehungen gelten.
- Sie sind auch nicht gut geeignet, wenn mehrere Benutzer gleichzeitig an den selben Daten arbeiten.
- Es gibt aber OSS software wie BaseX^{38,39} die Datenbankfunktionalität in dieses Gebiet bring.

Hierarchische Dokumente für Unverbundene Daten



- Eine andere Möglichkeit sind hierarchisch strukturierte einzelne Dokumente, in Formate wie Extensible Markup Language (XML)^{17,24,50}, JavaScript Object Notation (JSON)^{16,86} oder YAML Ain't Markup Language™ (YAML)^{23,31,50}.
- Diese Formate können tausende Datensätze speichern, die entweder gleich oder verschiedenartig strukturiert sind.
- Sie sind ungeeignet, wenn Datensätze von verschiedenen Typen in Beziehungen stehen und Einschränkungen für die Beziehungen gelten.
- Sie sind auch nicht gut geeignet, wenn mehrere Benutzer gleichzeitig an den selben Daten arbeiten.
- Es gibt aber OSS software wie BaseX^{38,39} die Datenbankfunktionalität in dieses Gebiet bring.
- Trotzdem sind solche Formate für unser System ungeeignet.

Hierarchische Datenbanken



- In der frühen Entwicklung des Gebiets der Datenbanken entstanden hierarchische Datenbanken wie IMS^{8,49,51}.



Hierarchische Datenbanken



- In der frühen Entwicklung des Gebiets der Datenbanken entstanden hierarchische Datenbanken wie IMS^{8,49,51}.
- Diese bieten die Vorteile der hierarchischen Dokumente mit der Fähigkeit von gleichzeitigem Zugriff.



Hierarchische Datenbanken



- In der frühen Entwicklung des Gebiets der Datenbanken entstanden hierarchische Datenbanken wie IMS^{8,49,51}.
- Diese bieten die Vorteile der hierarchischen Dokumente mit der Fähigkeit von gleichzeitigem Zugriff.
- Die hierarchische Struktur erzeugt aber Redundanz.

Hierarchische Datenbanken



- In der frühen Entwicklung des Gebiets der Datenbanken entstanden hierarchische Datenbanken wie IMS^{8,49,51}.
- Diese bieten die Vorteile der hierarchischen Dokumente mit der Fähigkeit von gleichzeitigem Zugriff.
- Die hierarchische Struktur erzeugt aber Redundanz:
- Wenn ein Professor zwei Kurse unterrichtet, dann werden die Daten des Professors zweimal gespeichert.

Hierarchische Datenbanken



- In der frühen Entwicklung des Gebiets der Datenbanken entstanden hierarchische Datenbanken wie IMS^{8,49,51}.
- Diese bieten die Vorteile der hierarchischen Dokumente mit der Fähigkeit von gleichzeitigem Zugriff.
- Die hierarchische Struktur erzeugt aber Redundanz:
- Wenn ein Professor zwei Kurse unterrichtet, dann werden die Daten des Professors zweimal gespeichert.
- Im Grunde ist dieser Ansatz zu Datenbanken veraltet.

Hierarchische Datenbanken



- In der frühen Entwicklung des Gebiets der Datenbanken entstanden hierarchische Datenbanken wie IMS^{8,49,51}.
- Diese bieten die Vorteile der hierarchischen Dokumente mit der Fähigkeit von gleichzeitigem Zugriff.
- Die hierarchische Struktur erzeugt aber Redundanz:
- Wenn ein Professor zwei Kurse unterrichtet, dann werden die Daten des Professors zweimal gespeichert.
- Im Grunde ist dieser Ansatz zu Datenbanken veraltet.
- Es gibt aber noch open source hierarchische DBMSs wie z. B. MUMPS^{63,64}.

Hierarchische Datenbanken



- In der frühen Entwicklung des Gebiets der Datenbanken entstanden hierarchische Datenbanken wie IMS^{8,49,51}.
- Diese bieten die Vorteile der hierarchischen Dokumente mit der Fähigkeit von gleichzeitigem Zugriff.
- Die hierarchische Struktur erzeugt aber Redundanz:
- Wenn ein Professor zwei Kurse unterrichtet, dann werden die Daten des Professors zweimal gespeichert.
- Im Grunde ist dieser Ansatz zu Datenbanken veraltet.
- Es gibt aber noch open source hierarchische DBMSs wie z. B. MUMPS^{63,64}.
- Der hierarchische Schlüssel-Wert-Speicher YottaDB ist auch eine Implementierung von MUMPS¹¹.

Netzwerkdatenmodell



- Ein weiterer historischer Ansatz ist das Netzwerkdatenmodell, das aus IDS^{3,4,40} und CODASYL⁸⁴ entstand.

Netzwerkdatenmodell



- Ein weiterer historischer Ansatz ist das Netzwerkdatenmodell, das aus IDS^{3,4,40} und CODASYL⁸⁴ entstand.
- Auch heute gibt es noch CODASYL- und COBOL-basierte DBMSe⁶⁶.

Netzwerkdatenmodell



- Ein weiterer historischer Ansatz ist das Netzwerkdatenmodell, das aus IDS^{3,4,40} und CODASYL⁸⁴ entstand.
- Auch heute gibt es noch CODASYL- und COBOL-basierte DBMSe⁶⁶.
- Diese Systeme werden oft als schwer zu bedienen angesehen.

Netzwerkdatenmodell



- Ein weiterer historischer Ansatz ist das Netzwerkdatenmodell, das aus IDS^{3,4,40} und CODASYL⁸⁴ entstand.
- Auch heute gibt es noch CODASYL- und COBOL-basierte DBMSe⁶⁶.
- Diese Systeme werden oft als schwer zu bedienen angesehen.
- Wie bei hierarchischen Datenbanken kann man diese Systeme als veraltet betrachten.

Schlüssel-Wert Speicher



- Ein weiteres Datenmodell sind Schlüssel-Wert-Speicher (EN: *key-value stores*).

Schlüssel-Wert Speicher



- Ein weiteres Datenmodell sind Schlüssel-Wert-Speicher (EN: *key-value stores*).
- Diese sind im Grunde Schema-los und geeignet, wenn die Daten eine Schlüssel-Wert-Struktur haben.

Schlüssel-Wert Speicher



- Ein weiteres Datenmodell sind Schlüssel-Wert-Speicher (EN: *key-value stores*).
- Diese sind im Grunde Schema-los und geeignet, wenn die Daten eine Schlüssel-Wert-Struktur haben.
- Für unsere Lehre-Management-Plattform erscheinen sie ungeeignet.

Relationales Datenmodell

- Der Fokus unseres Kurses ist auf dem relationalen Datenmodell, das Codd²² erfunden hat, und der Sprache Structured Query Language (SQL).



Relationales Datenmodell

- Der Fokus unseres Kurses ist auf dem relationalen Datenmodell, das Codd²² erfunden hat, und der Sprache SQL.
- Dieses Modell passt zu sehr vielen Anwendungsfällen.



Relationales Datenmodell



- Der Fokus unseres Kurses ist auf dem relationalen Datenmodell, das Codd²² erfunden hat, und der Sprache SQL.
- Dieses Modell passt zu sehr vielen Anwendungsfällen.
- Wenn Sie Ihre Daten in einem der vorhin genannten Modelle ausdrücken können, dann können Sie es wahrscheinlich auch mit dem relationalen Modell.

Relationales Datenmodell



- Der Fokus unseres Kurses ist auf dem relationalen Datenmodell, das Codd²² erfunden hat, und der Sprache SQL.
- Dieses Modell passt zu sehr vielen Anwendungsfällen.
- Wenn Sie Ihre Daten in einem der vorhin genannten Modelle ausdrücken können, dann können Sie es wahrscheinlich auch mit dem relationalen Modell.
- Im Vergleich zu den vorhin genannten Modellen gibt es einfach unendlich viel mehr open source oder kommerzielle relationale DBMSs.

Relationales Datenmodell



- Der Fokus unseres Kurses ist auf dem relationalen Datenmodell, das Codd²² erfunden hat, und der Sprache SQL.
- Dieses Modell passt zu sehr vielen Anwendungsfällen.
- Wenn Sie Ihre Daten in einem der vorhin genannten Modelle ausdrücken können, dann können Sie es wahrscheinlich auch mit dem relationalen Modell.
- Im Vergleich zu den vorhin genannten Modellen gibt es einfach unendlich viel mehr open source oder kommerzielle relationale DBMSs.
- In diesen werden Entitäten als Relationen beschrieben und in Tabellen in der Datenbank gespeichert.

Relationales Datenmodell



- Der Fokus unseres Kurses ist auf dem relationalen Datenmodell, das Codd²² erfunden hat, und der Sprache SQL.
- Dieses Modell passt zu sehr vielen Anwendungsfällen.
- Wenn Sie Ihre Daten in einem der vorhin genannten Modelle ausdrücken können, dann können Sie es wahrscheinlich auch mit dem relationalen Modell.
- Im Vergleich zu den vorhin genannten Modellen gibt es einfach unendlich viel mehr open source oder kommerzielle relationale DBMSs.
- In diesen werden Entitäten als Relationen beschrieben und in Tabellen in der Datenbank gespeichert.
- Wie wir bereits ausprobiert haben, können Tabellen in Beziehungen stehen und Einschränkungen können sicherstellen, dass die Beziehungen zwischen ihnen konsistent bleiben.

Relationales Datenmodell



- Der Fokus unseres Kurses ist auf dem relationalen Datenmodell, das Codd²² erfunden hat, und der Sprache SQL.
- Dieses Modell passt zu sehr vielen Anwendungsfällen.
- Wenn Sie Ihre Daten in einem der vorhin genannten Modelle ausdrücken können, dann können Sie es wahrscheinlich auch mit dem relationalen Modell.
- Im Vergleich zu den vorhin genannten Modellen gibt es einfach unendlich viel mehr open source oder kommerzielle relationale DBMSs.
- In diesen werden Entitäten als Relationen beschrieben und in Tabellen in der Datenbank gespeichert.
- Wie wir bereits ausprobiert haben, können Tabellen in Beziehungen stehen und Einschränkungen können sicherstellen, dass die Beziehungen zwischen ihnen konsistent bleiben.
- Die meisten unserer Entitätstypen können wahrscheinlich einfach in zweidimensionale Tabellen abgebildet werden.

Relationales Datenmodell



- Der Fokus unseres Kurses ist auf dem relationalen Datenmodell, das Codd²² erfunden hat, und der Sprache SQL.
- Dieses Modell passt zu sehr vielen Anwendungsfällen.
- Wenn Sie Ihre Daten in einem der vorhin genannten Modelle ausdrücken können, dann können Sie es wahrscheinlich auch mit dem relationalen Modell.
- Im Vergleich zu den vorhin genannten Modellen gibt es einfach unendlich viel mehr open source oder kommerzielle relationale DBMSs.
- In diesen werden Entitäten als Relationen beschrieben und in Tabellen in der Datenbank gespeichert.
- Wie wir bereits ausprobiert haben, können Tabellen in Beziehungen stehen und Einschränkungen können sicherstellen, dass die Beziehungen zwischen ihnen konsistent bleiben.
- Die meisten unserer Entitätstypen können wahrscheinlich einfach in zweidimensionale Tabellen abgebildet werden.
- Die meisten unserer Attribute sind einfach.

Relationales Datenmodell



- Der Fokus unseres Kurses ist auf dem relationalen Datenmodell, das Codd²² erfunden hat, und der Sprache SQL.
- Dieses Modell passt zu sehr vielen Anwendungsfällen.
- Wenn Sie Ihre Daten in einem der vorhin genannten Modelle ausdrücken können, dann können Sie es wahrscheinlich auch mit dem relationalen Modell.
- Im Vergleich zu den vorhin genannten Modellen gibt es einfach unendlich viel mehr open source oder kommerzielle relationale DBMSs.
- In diesen werden Entitäten als Relationen beschrieben und in Tabellen in der Datenbank gespeichert.
- Wie wir bereits ausprobiert haben, können Tabellen in Beziehungen stehen und Einschränkungen können sicherstellen, dass die Beziehungen zwischen ihnen konsistent bleiben.
- Die meisten unserer Entitätstypen können wahrscheinlich einfach in zweidimensionale Tabellen abgebildet werden.
- Die meisten unserer Attribute sind einfach.
- Die Beziehungen in den Diagrammen können wahrscheinlich direkt in Fremdschlüssel oder zusätzliche Tabellen überführt werden.



Zusammenfassung



Zusammenfassung



- Wir entscheiden uns, unsere Lehre-Management-Plattform im *relationalen Datenmodell* zu implementieren.

Zusammenfassung



- Wir entscheiden uns, unsere Lehre-Management-Plattform im *relationalen Datenmodell* zu implementieren.
- Und dieses Datenmodell schauen wir uns im Folgenden genauer an.



谢谢你们！
Thank you!
Vielen Dank!



References I



- [1] Adam Aspin und Karine Aspin. *Query Answers with MariaDB – Volume I: Introduction to SQL Queries*. Tetras Publishing, Okt. 2018. ISBN: 978-1-9996172-4-0. See also² (siehe S. 49, 61).
- [2] Adam Aspin und Karine Aspin. *Query Answers with MariaDB – Volume II: In-Depth Querying*. Tetras Publishing, Okt. 2018. ISBN: 978-1-9996172-5-7. See also¹ (siehe S. 49, 61).
- [3] Charles William „Charlie“ Bachman. "Software for Random Access Processing". *Datamation* 9(4):36–41, Apr. 1965. Chicago, IL, USA: Technical Publishing Co. and Boston, MA, USA: Cahners Publishing Company. ISSN: 0011-6963 (siehe S. 29–32).
- [4] Charles William „Charlie“ Bachman. "The Origin of the Integrated Data Store (IDS): The First Direct-Access DBMS". *IEEE Annals of the History of Computing* 31(4):42–54, Okt.–Dez. 2009. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE). ISSN: 1058-6180. doi:10.1109/MAHC.2009.110. URL: <https://tschwarz.mscs.mu.edu/Classes/DB23/HW/bachmanIDS.pdf> (besucht am 2025-01-08) (siehe S. 29–32).
- [5] Daniel J. Barrett. *Efficient Linux at the Command Line*. Sebastopol, CA, USA: O'Reilly Media, Inc., Feb. 2022. ISBN: 978-1-0981-1340-7 (siehe S. 61, 63).
- [6] Daniel Bartholomew. *Learning the MariaDB Ecosystem: Enterprise-level Features for Scalability and Availability*. New York, NY, USA: Apress Media, LLC, Okt. 2019. ISBN: 978-1-4842-5514-8 (siehe S. 61).
- [7] Ben Beitler. *Hands-On Microsoft Access 2019*. Birmingham, England, UK: Packt Publishing Ltd, März 2020. ISBN: 978-1-83898-747-3 (siehe S. 61).
- [8] Uri Berman, Jackie Berman und Bob Patrick. *The Birth of IMS/360*. Techn. Ber. 102762458. Mountain View, CA, USA: Computer History Museum (CHM), Apr. 2007. URL: <https://archive.computerhistory.org/resources/access/text/2016/12/102762458-05-01-acc.pdf> (besucht am 2025-01-08) (siehe S. 22–28).
- [9] Tim Berners-Lee. *Re: Qualifiers on Hypertext links...* Geneva, Switzerland: World Wide Web project, European Organization for Nuclear Research (CERN) und Newsgroups: alt.hypertext, 6. Aug. 1991. URL: <https://www.w3.org/People/Berners-Lee/1991/08/art-6484.txt> (besucht am 2025-02-05) (siehe S. 63).

References II



- [10] Alex Berson. *Client/Server Architecture*. 2. Aufl. Computer Communications Series. New York, NY, USA: McGraw-Hill, 29. März 1996. ISBN: 978-0-07-005664-0 (siehe S. 60).
- [11] K.S. Bhaskar. *The Heritage and Legacy of M (MUMPS) – and the Future of YottaDB*. Malvern, PA, USA: YottaDB, LLC., 14. Feb. 2018. URL: <https://yottadb.com/heritage-legacy-m-mumps-future-yottadb> (besucht am 2025-04-04) (siehe S. 22–28).
- [12] Bernard Obeng Boateng. *Data Modeling with Microsoft Excel*. Birmingham, England, UK: Packt Publishing Ltd, Nov. 2023. ISBN: 978-1-80324-028-2 (siehe S. 10–15, 62).
- [13] Silvia Botros und Jeremy Tinley. *High Performance MySQL*. 4. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., Nov. 2021. ISBN: 978-1-4920-8051-0 (siehe S. 62).
- [14] Ed Bott. *Windows 11 Inside Out*. Hoboken, NJ, USA: Microsoft Press, Pearson Education, Inc., Feb. 2023. ISBN: 978-0-13-769132-6 (siehe S. 62).
- [15] Ron Brash und Ganesh Naik. *Bash Cookbook*. Birmingham, England, UK: Packt Publishing Ltd, Juli 2018. ISBN: 978-1-78862-936-2 (siehe S. 60).
- [16] Tim Bray. *The JavaScript Object Notation (JSON) Data Interchange Format*. Request for Comments (RFC) 8259. Wilmington, DE, USA: Internet Engineering Task Force (IETF), Dez. 2017. URL: <https://www.ietf.org/rfc/rfc8259.txt> (besucht am 2025-02-05) (siehe S. 16–21, 61).
- [17] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen und Eve Maler, Hrsg. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. W3C Recommendation. Wakefield, MA, USA: World Wide Web Consortium (W3C), 26. Nov. 2008–7. Feb. 2013. URL: <http://www.w3.org/TR/2008/REC-xml-20081126> (besucht am 2024-12-15) (siehe S. 16–21, 64).
- [18] Jason Cannon. *High Availability for the LAMP Stack*. Shelter Island, NY, USA: Manning Publications, Juni 2022 (siehe S. 61, 63).
- [19] Donald D. Chamberlin. "50 Years of Queries". *Communications of the ACM (CACM)* 67(8):110–121, Aug. 2024. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 0001-0782. doi:10.1145/3649887. URL: <https://cacm.acm.org/research/50-years-of-queries> (besucht am 2025-01-09) (siehe S. 63).

References III



- [20] Christmas, FL, USA: Simon Sez IT. *Microsoft Access 2021 – Beginner to Advanced*. Birmingham, England, UK: Packt Publishing Ltd, Aug. 2023. ISBN: **978-1-83546-911-8** (siehe S. **61**).
- [21] David Clinton und Christopher Negus. *Ubuntu Linux Bible*. 10. Aufl. Bible Series. Chichester, West Sussex, England, UK: John Wiley and Sons Ltd., 10. Nov. 2020. ISBN: **978-1-119-72233-5** (siehe S. **63**).
- [22] Edgar Frank „Ted“ Codd. "A Relational Model of Data for Large Shared Data Banks". *Communications of the ACM (CACM)* 13(6):377–387, Juni 1970. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: **0001-0782**. doi:**10.1145/362384.362685**. URL: <https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf> (besucht am 2025-01-05) (siehe S. **36–44, 62**).
- [23] Coding Gears und Train Your Brain. *YAML Fundamentals for DevOps, Cloud and IaC Engineers*. Birmingham, England, UK: Packt Publishing Ltd, März 2022. ISBN: **978-1-80324-243-9** (siehe S. **16–21, 64**).
- [24] Timothy W. Cole und Myung-Ja K. Han. *XML for Catalogers and Metadata Librarians (Third Millennium Cataloging)*. 1. Aufl. Dublin, OH, USA: Libraries Unlimited, 23. Mai 2013. ISBN: **978-1-59884-519-8** (siehe S. **16–21, 64**).
- [25] "csv – CSV File Reading and Writing". In: *Python 3 Documentation. The Python Standard Library*. Beaverton, OR, USA: Python Software Foundation (PSF), 2001–2025. URL: <https://docs.python.org/3/library/csv.html> (besucht am 2024-11-14) (siehe S. **60**).
- [26] *Database Language SQL*. Techn. Ber. ANSI X3.135-1986. Washington, D.C., USA: American National Standards Institute (ANSI), 1986 (siehe S. **63**).
- [27] Matt David und Blake Barnhill. *How to Teach People SQL*. San Francisco, CA, USA: The Data School, Chart.io, Inc., 10. Dez. 2019–10. Apr. 2023. URL: <https://dataschool.com/how-to-teach-people-sql> (besucht am 2025-02-27) (siehe S. **63**).
- [28] *Database Language SQL*. International Standard ISO 9075-1987. Geneva, Switzerland: International Organization for Standardization (ISO), 1987 (siehe S. **63**).
- [29] Paul Deitel, Harvey Deitel und Abbey Deitel. *Internet & World Wide WebW[?]: How to Program*. 5. Aufl. Hoboken, NJ, USA: Pearson Education, Inc., Nov. 2011. ISBN: **978-0-13-299045-5** (siehe S. **63**).

References IV



- [30] Pooyan Doozandeh und Frank E. Ritter. "Some Tips for Academic Writing and Using Microsoft Word". *XRDS: Crossroads, The ACM Magazine for Students* 26(1):10–11, Herbst 2019. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 1528-4972. doi:10.1145/3351470 (siehe S. 62).
- [31] Ingy döt Net, Tina Müller, Pantelis Antoniou, Eemeli Aro, Thomas Smith, Oren Ben-Kiki und Clark C. Evans. *YAML Ain't Markup Language (YAML™) version 1.2. Revision 1.2.2*. Seattle, WA, USA: YAML Language Development Team, 1. Okt. 2021. URL: <https://yaml.org/spec/1.2.2> (besucht am 2025-01-05) (siehe S. 16–21, 64).
- [32] Russell J.T. Dyer. *Learning MySQL and MariaDB*. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2015. ISBN: 978-1-4493-6290-4 (siehe S. 61, 62).
- [33] *ECMAScript Language Specification*. Standard ECMA-262, 3rd Edition. Geneva, Switzerland: Ecma International, Dez. 1999. URL: https://ecma-international.org/wp-content/uploads/ECMA-262_3rd_edition_december_1999.pdf (besucht am 2024-12-15) (siehe S. 61).
- [34] Steve Fanning, Vasudev Narayanan, „flywire“, Olivier Hallot, Jean Hollis Weber, Jenna Sargent, Pulkit Krishna, Dan Lewis, Peter Schofield, Jochen Schiffers, Robert Großkopf, Jost Lange, Martin Fox, Hazel Russman, Steve Schwettman, Alain Romenenne, Andrew Pitonyak, Jean-Pierre Ledure, Drew Jensen und Randolph Gam. *Base Guide 7.3. Revision 1. Based on LibreOffice 7.3 Community*. Berlin, Germany: The Document Foundation, Aug. 2022. URL: <https://books.libreoffice.org/en/BG73/BG73-BaseGuide.pdf> (besucht am 2025-01-13) (siehe S. 61).
- [35] Luca Ferrari und Enrico Pirozzi. *Learn PostgreSQL*. 2. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Okt. 2023. ISBN: 978-1-83763-564-1 (siehe S. 62).
- [36] Jonas Gamalielsson und Björn Lundell. "Long-Term Sustainability of Open Source Software Communities beyond a Fork: A Case Study of LibreOffice". In: *8th IFIP WG 2.13 International Conference on Open Source Systems: Long-Term Sustainability OSS'2012*. 10.–13. Sep. 2012, Hammamet, Tunisia. Hrsg. von Imed Hammouda, Björn Lundell, Tommi Mikkonen und Walt Scacchi. Bd. 378. IFIP Advances in Information and Communication Technology (IFIPAICT). Berlin/Heidelberg, Germany: Springer-Verlag GmbH Germany, 2012, S. 29–47. ISSN: 1868-4238. ISBN: 978-3-642-33441-2. doi:10.1007/978-3-642-33442-9_3 (siehe S. 61).
- [37] Dawn Griffiths. *Excel Cookbook – Receipts for Mastering Microsoft Excel*. Sebastopol, CA, USA: O'Reilly Media, Inc., Mai 2024. ISBN: 978-1-0981-4332-9 (siehe S. 10–15, 62).

References V



- [38] Christian Grün. "Pushing XML Main Memory Databases to their Limits". In: *Tagungsband zum 18. GI-Workshop über Grundlagen von Datenbanken. 18th GI-Workshop on the Foundations of Databases*. 6.–9. Juni 2006, Wittenberg, Sachsen-Anhalt, Germany. Hrsg. von Stefan Brass und Alexander Hinneburg. 2006, S. 60–64. URL: https://dbs.informatik.uni-halle.de/GvD2006/gvd06_gruen.pdf (siehe S. 16–21).
- [39] Christian Grün, Alexander Holupirek und Michael Seiferle. *BaseX*. Konstanz, Baden-Württemberg, Germany: BaseX GmbH, Jan. 2010–März 2025. URL: <https://basex.org> (besucht am 2025-04-09) (siehe S. 16–21).
- [40] Thomas Haigh. "How Charles Bachman Invented the DBMS, A Foundation of Our Digital World". *Communications of the ACM (CACM)* 59(7):25–30, Juli 2016. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 0001-0782. doi:10.1145/2935880. URL: <https://cacm.acm.org/opinion/how-charles-bachman-invented-the-dbms-a-foundation-of-our-digital-world> (besucht am 2025-05-08) (siehe S. 29–32).
- [41] Terry Halpin und Tony Morgan. *Information Modeling and Relational Databases*. 3. Aufl. Burlington, MA, USA/San Mateo, CA, USA: Morgan Kaufmann Publishers, Juli 2024. ISBN: 978-0-443-23791-1 (siehe S. 62).
- [42] Jan L. Harrington. *Relational Database Design and Implementation*. 4. Aufl. Burlington, MA, USA/San Mateo, CA, USA: Morgan Kaufmann Publishers, Apr. 2016. ISBN: 978-0-12-849902-3 (siehe S. 62).
- [43] Michael Hausenblas. *Learning Modern Linux*. Sebastopol, CA, USA: O'Reilly Media, Inc., Apr. 2022. ISBN: 978-1-0981-0894-6 (siehe S. 61).
- [44] Christian Heimes. "`defusedxml` 0.7.1: XML Bomb Protection for Python stdlib Modules". In: 8. März 2021. URL: <https://pypi.org/project/defusedxml> (besucht am 2024-12-15) (siehe S. 64).
- [45] Matthew Helmke. *Ubuntu Linux Unleashed 2021 Edition*. 14. Aufl. Reading, MA, USA: Addison-Wesley Professional, Aug. 2020. ISBN: 978-0-13-668539-5 (siehe S. 61, 63).
- [46] Manuel Hoffmann, Frank Nagle und Yanuo Zhou. *The Value of Open Source Software*. Working Paper 24-038. Boston, MA, USA: Harvard Business School, 1. Jan. 2024. URL: https://www.hbs.edu/ris/Publication%20Files/24-038_51f8444f-502c-4139-8bf2-56eb4b65c58a.pdf (besucht am 2025-06-04) (siehe S. 62).

References VI



- [47] John Hunt. *A Beginners Guide to Python 3 Programming*. 2. Aufl. Undergraduate Topics in Computer Science (UTICS). Cham, Switzerland: Springer, 2023. ISBN: 978-3-031-35121-1. doi:10.1007/978-3-031-35122-8 (siehe S. 62).
- [48] *Information Technology – Database Languages – SQL – Part 1: Framework (SQL/Framework)*, Part 1. International Standard ISO/IEC 9075-1:2023(E), Sixth Edition, (ANSI X3.135). Geneva, Switzerland: International Organization for Standardization (ISO) und International Electrotechnical Commission (IEC), Juni 2023. URL: [https://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_IEC_9075-1_2023_ed_6_-_id_76583_Publication_PDF_\(en\).zip](https://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_IEC_9075-1_2023_ed_6_-_id_76583_Publication_PDF_(en).zip) (besucht am 2025-01-08). Consists of several parts, see <https://modern-sql.com/standard> for information where to obtain them. (Siehe S. 63).
- [49] Barbara Klein, Richard Alan Long, Kenneth Ray Blackman, Diane Lynne Goff, Stephen P. Nathan, Moira McFadden Lanyi, Margaret M. Wilson, John Butterweck und Sandra L. Sherrill. *Introduction to IMS: Your Complete Guide to IBM Information Management System*. 2. Aufl. Indianapolis, IN, USA: IBM Press, 13. März 2012. ISBN: 978-0-13-288687-1 (siehe S. 22–28).
- [50] Katie Kodes. *Intro to XML, JSON, & YAML*. London, England, UK: Payhip, 2019–4. Sep. 2020 (siehe S. 16–21, 64).
- [51] Tim Kraska und Michael Cafarella. "Introduction to Databases". In: *6.5830/6.5831: Database Systems*. Cambridge, MA, USA: Massachusetts Institute of Technology (MIT), Herbst 2024. Kap. 1. URL: <https://dsg.csail.mit.edu/6.5830/lectures/lec1.pdf> (besucht am 2025-01-08) (siehe S. 22–28).
- [52] Jay LaCroix. *Mastering Ubuntu Server*. 4. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Sep. 2022. ISBN: 978-1-80323-424-3 (siehe S. 63).
- [53] Joan Lambert und Curtis Frye. *Microsoft Office Step by Step (Office 2021 and Microsoft 365)*. Hoboken, NJ, USA: Microsoft Press, Pearson Education, Inc., Juni 2022. ISBN: 978-0-13-754493-6 (siehe S. 10–15, 62).
- [54] Kent D. Lee und Steve Hubbard. *Data Structures and Algorithms with Python*. Undergraduate Topics in Computer Science (UTICS). Cham, Switzerland: Springer, 2015. ISBN: 978-3-319-13071-2. doi:10.1007/978-3-319-13072-9 (siehe S. 62).
- [55] *LibreOffice – The Document Foundation*. Berlin, Germany: The Document Foundation, 2024. URL: <https://www.libreoffice.org> (besucht am 2024-12-12) (siehe S. 10–15, 61, 62).

References VII



- [56] Gloria Lotha, Aakanksha Gaur, Erik Gregersen, Swati Chopra und William L. Hosch. "Client-Server Architecture". In: *Encyclopaedia Britannica*. Hrsg. von The Editors of Encyclopaedia Britannica. Chicago, IL, USA: Encyclopædia Britannica, Inc., 3. Jan. 2025. URL: <https://www.britannica.com/technology/client-server-architecture> (besucht am 2025-01-20) (siehe S. 60).
- [57] Mark Lutz. *Learning Python*. 6. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2025. ISBN: 978-1-0981-7130-8 (siehe S. 62).
- [58] *MariaDB Server Documentation*. Milpitas, CA, USA: MariaDB, 2025. URL: <https://mariadb.com/kb/en/documentation> (besucht am 2025-04-24) (siehe S. 61).
- [59] Ron McFadyen und Cindy Miller. *Relational Databases and Microsoft Access*. 3. Aufl. Palatine, IL, USA: Harper College, 2014–2019. URL: <https://harpercollege.pressbooks.pub/relationaldatabases> (besucht am 2025-04-11) (siehe S. 61).
- [60] Jim Melton und Alan R. Simon. *SQL: 1999 – Understanding Relational Language Components*. The Morgan Kaufmann Series in Data Management Systems. Burlington, MA, USA/San Mateo, CA, USA: Morgan Kaufmann Publishers, Juni 2001. ISBN: 978-1-55860-456-8 (siehe S. 63).
- [61] *Microsoft Word*. Redmond, WA, USA: Microsoft Corporation, 2024. URL: <https://www.microsoft.com/en-us/microsoft-365/word> (besucht am 2024-12-12) (siehe S. 62).
- [62] Cameron Newham und Bill Rosenblatt. *Learning the Bash Shell – Unix Shell Programming: Covers Bash 3.0*. 3. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., 2005. ISBN: 978-0-596-00965-6 (siehe S. 60).
- [63] Kevin C. O'Kane. *The Mumps Programming Language*. North Charleston, SC, USA: CreateSpace Independent Publishing Platform, 19. Juni 2008. ISBN: 978-1-4382-4338-2 (siehe S. 22–28).
- [64] Kevin C. O'Kane. *The Mumps Programming Language*. Cedar Falls, IA, USA: University of Northern Iowa, 4. Apr. 2025. URL: <https://www.cs.uni.edu/~okane> (besucht am 2025-04-04) (siehe S. 22–28).
- [65] Regina O. Obe und Leo S. Hsu. *PostgreSQL: Up and Running*. 3. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., Okt. 2017. ISBN: 978-1-4919-6336-4 (siehe S. 62).
- [66] *Oracle CODASYL DBMS™ – Database Administration Reference Manual*. 7.4.1. Redwood Shores, CA, USA: Oracle Corporation, Aug. 2022. URL: <https://www.oracle.com/a/tech/docs/collateral/dbm0741-dbadmin-ref.pdf> (siehe S. 29–32).

References VIII



- [67] Robert Orfali, Dan Harkey und Jeri Edwards. *Client/Server Survival Guide*. 3. Aufl. Chichester, West Sussex, England, UK: John Wiley and Sons Ltd., 25. Jan. 1999. ISBN: **978-0-471-31615-2** (siehe S. 60).
- [68] Yasset Pérez-Riverol, Laurent Gatto, Rui Wang, Timo Sachsenberg, Julian Uszkoreit, Felipe da Veiga Leprevost, Christian Fufezan, Tobias Ternent, Stephen J. Eglén, Daniel S. Katz, Tom J. Pollard, Alexander Konovalov, Robert M. Flight, Kai Blin und Juan Antonio Vizcaíno. "Ten Simple Rules for Taking Advantage of Git and GitHub". *PLOS Computational Biology* 12(7), 14. Juli 2016. San Francisco, CA, USA: Public Library of Science (PLOS). ISSN: **1553-7358**. doi:**10.1371/JOURNAL.PCBI.1004947** (siehe S. 60).
- [69] *PostgreSQL Essentials: Leveling Up Your Data Work*. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2024 (siehe S. 62).
- [70] Abhishek Ratan, Eric Chou, Pradeeban Kathiravelu und Dr. M.O. Faruque Sarker. *Python Network Programming*. Birmingham, England, UK: Packt Publishing Ltd, Jan. 2019. ISBN: **978-1-78883-546-6** (siehe S. 60).
- [71] Federico Razzoli. *Mastering MariaDB*. Birmingham, England, UK: Packt Publishing Ltd, Sep. 2014. ISBN: **978-1-78398-154-0** (siehe S. 61).
- [72] Mike Reichardt, Michael Gundall und Hans D. Schotten. "Benchmarking the Operation Times of NoSQL and MySQL Databases for Python Clients". In: *47th Annual Conference of the IEEE Industrial Electronics Society (IECON'2021)*. 13.–15. Okt. 2021, Toronto, ON, Canada. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE), 2021, S. 1–8. ISSN: **2577-1647**. ISBN: **978-1-6654-3554-3**. doi:**10.1109/IECON48115.2021.9589382** (siehe S. 62).
- [73] Mark Richards und Neal Ford. *Fundamentals of Software Architecture: An Engineering Approach*. Sebastopol, CA, USA: O'Reilly Media, Inc., Jan. 2020. ISBN: **978-1-4920-4345-4** (siehe S. 60).
- [74] Heinz Schweppe und Manuel Scholz. "Introduction". In: *Einführung in die Datenbanksysteme. Datenbanken für die Bioinformatik*. Berlin, Germany: Freie Universität Berlin, Apr.–Okt. 2005. Kap. 1. URL: <https://www.inf.fu-berlin.de/lehre/SS05/19517-V/FolienEtc/dbs045-01-Intro-2.pdf> (besucht am 2025-01-08) (siehe S. 5–8).
- [75] Winfried Seimert. *LibreOffice 7.3 – Praxiswissen für Ein- und Umsteiger*. Blaufelden, Schwäbisch Hall, Baden-Württemberg, Germany: mitp Verlags GmbH & Co. KG, Apr. 2022. ISBN: **978-3-7475-0504-5** (siehe S. 10–15, 61, 62).

References IX



- [76] Yakov Shafranovich. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. Request for Comments (RFC) 4180. Wilmington, DE, USA: Internet Engineering Task Force (IETF), Okt. 2005. URL: <https://www.ietf.org/rfc/rfc4180.txt> (besucht am 2025-02-05) (siehe S. 10–15, 60).
- [77] Ellen Siever, Stephen Figgins, Robert Love und Arnold Robbins. *Linux in a Nutshell*. 6. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., Sep. 2009. ISBN: 978-0-596-15448-6 (siehe S. 61).
- [78] Anna Skoulikari. *Learning Git*. Sebastopol, CA, USA: O'Reilly Media, Inc., Mai 2023. ISBN: 978-1-0981-3391-7 (siehe S. 60).
- [79] John Miles Smith und Philip Yen-Tang Chang. "Optimizing the Performance of a Relational Algebra Database Interface". *Communications of the ACM (CACM)* 18(10):568–579, Okt. 1975. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 0001-0782. doi:10.1145/361020.361025 (siehe S. 62).
- [80] "SQL Commands". In: *PostgreSQL Documentation*. 17.4. The PostgreSQL Global Development Group (PGDG), 20. Feb. 2025. Kap. Part VI. Reference. URL: <https://www.postgresql.org/docs/17/sql-commands.html> (besucht am 2025-02-25) (siehe S. 63).
- [81] Ryan K. Stephens und Ronald R. Plew. *Sams Teach Yourself SQL in 21 Days*. 4. Aufl. Sams Tech Yourself. Indianapolis, IN, USA: SAMS Technical Publishing und Hoboken, NJ, USA: Pearson Education, Inc., Okt. 2002. ISBN: 978-0-672-32451-2 (siehe S. 57, 63).
- [82] Ryan K. Stephens, Ronald R. Plew, Bryan Morgan und Jeff Perkins. *SQL in 21 Tagen. Die Datenbank-Abfragesprache SQL vollständig erklärt (in 14/21 Tagen)*. 6. Aufl. Burghann, Bayern, Germany: Markt+Technik Verlag GmbH, Feb. 1998. ISBN: 978-3-8272-2020-2. Translation of⁸¹ (siehe S. 63).
- [83] Allen Taylor. *Introducing SQL and Relational Databases*. New York, NY, USA: Apress Media, LLC, Sep. 2018. ISBN: 978-1-4842-3841-7 (siehe S. 62, 63).
- [84] Robert W. Taylor und Randall L. Frank. "CODASYL Data-Base Management Systems". *ACM Computing Surveys (CSUR)* 8(1):67–103, März 1976. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 0360-0300. doi:10.1145/356662.356666 (siehe S. 29–32).

References X



- [85] Alkin Tezuysal und Ibrar Ahmed. *Database Design and Modeling with PostgreSQL and MySQL*. Birmingham, England, UK: Packt Publishing Ltd, Juli 2024. ISBN: **978-1-80323-347-5** (siehe S. **62**).
- [86] *The JSON Data Interchange Syntax*. Standard ECMA-404, 2nd Edition. Geneva, Switzerland: Ecma International, Dez. 2017. URL: <https://ecma-international.org/publications-and-standards/standards/ecma-404> (besucht am 2024-12-15) (siehe S. **16–21, 61**).
- [87] Linus Torvalds. "The Linux Edge". *Communications of the ACM (CACM)* 42(4):38–39, Apr. 1999. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: **0001-0782**. doi:[10.1145/299157.299165](https://doi.org/10.1145/299157.299165) (siehe S. **61**).
- [88] Mariot Tsitoara. *Beginning Git and GitHub: Version Control, Project Management and Teamwork for the New Developer*. New York, NY, USA: Apress Media, LLC, März 2024. ISBN: **979-8-8688-0215-7** (siehe S. **60, 63**).
- [89] Laurie A. Ulrich und Ken Cook. *Access For Dummies*. Hoboken, NJ, USA: For Dummies (Wiley), Dez. 2021. ISBN: **978-1-119-82908-9** (siehe S. **61**).
- [90] Sander van Vugt. *Linux Fundamentals*. 2. Aufl. Hoboken, NJ, USA: Pearson IT Certification, Juni 2022. ISBN: **978-0-13-792931-3** (siehe S. **61**).
- [91] Thomas Weise (汤卫思). *Databases*. Hefei, Anhui, China (中国安徽省合肥市): Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), Institute of Applied Optimization (应用优化研究所, IAO), 2025. URL: <https://thomasweise.github.io/databases> (besucht am 2025-01-05) (siehe S. **60–62**).
- [92] Thomas Weise (汤卫思). *Programming with Python*. Hefei, Anhui, China (中国安徽省合肥市): Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), Institute of Applied Optimization (应用优化研究所, IAO), 2024–2025. URL: <https://thomasweise.github.io/programmingWithPython> (besucht am 2025-01-05) (siehe S. **62**).
- [93] *What is a Relational Database?* Armonk, NY, USA: International Business Machines Corporation (IBM), 20. Okt. 2021–12. Dez. 2024. URL: <https://www.ibm.com/think/topics/relational-databases> (besucht am 2025-01-05) (siehe S. **62**).
- [94] Ulf Michael „Monty“ Widenius, David Axmark und Uppsala, Sweden: MySQL AB. *MySQL Reference Manual – Documentation from the Source*. Sebastopol, CA, USA: O'Reilly Media, Inc., 9. Juli 2002. ISBN: **978-0-596-00265-7** (siehe S. **62**).

References XI



- [95] Kinza Yasar und Craig S. Mullins. *Definition: Database Management System (DBMS)*. Newton, MA, USA: TechTarget, Inc., Juni 2024. URL: <https://www.techtarget.com/searchdatamanagement/definition/database-management-system> (besucht am 2025-01-11) (siehe S. 60).
- [96] Pavlo V. Zahorodko und Pavlo V. Merzlykin. "An Approach for Processing and Document Flow Automation for Microsoft Word and LibreOffice Writer File Formats". In: *4th Workshop for Young Scientists in Computer Science & Software Engineering (CS&SE@SW'2021)*. 18. Dez. 2021, Virtual Event and Kryvyi Rih, Ukraine. Hrsg. von Arnold E. Kiv, Serhiy O. Semerikov, Vladimir N. Soloviev und Andrii M. Striuk. Bd. 3077 der Reihe CEUR Workshop Proceedings (CEUR-WS.org). Aachen, Nordrhein-Westfalen, Germany: CEUR-WS Team, Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen, 2022, S. 66–82. ISSN: 1613-0073. URL: <https://ceur-ws.org/Vol-3077/paper12.pdf> (besucht am 2025-10-04) (siehe S. 61, 62).
- [97] Giorgio Zarrelli. *Mastering Bash*. Birmingham, England, UK: Packt Publishing Ltd, Juni 2017. ISBN: 978-1-78439-687-9 (siehe S. 60).

Glossary (in English) I



Bash is a the shell used under Ubuntu Linux, i.e., the program that „runs“ in the terminal and interprets your commands, allowing you to start and interact with other programs^{15,62,97}. Learn more at <https://www.gnu.org/software/bash>.

client In a client-server architecture, the client is a device or process that requests a service from the server. It initiates the communication with the server, sends a request, and receives the response with the result of the request. Typical examples for clients are web browsers in the internet as well as clients for database management systems (DBMSes), such as `psql`.

client-server architecture is a system design where a central server receives requests from one or multiple clients^{10,56,67,70,73}. These requests and responses are usually sent over network connections. A typical example for such a system is the World Wide Web (WWW), where web servers host websites and make them available to web browsers, the clients. Another typical example is the structure of database (DB) software, where a central server, the DBMS, offers access to the DB to the different clients. Here, the client can be some terminal software shipping with the DBMS, such as `psql`, or the different applications that access the DBs.

CSV *Comma-Separated Values* is a very common and simple text format for exchanging tabular or matrix data⁷⁶. Each row in the text file represents one row in the table or matrix. The elements in the row are separated by a fixed delimiter, usually a comma („,"), sometimes a semicolon („;"). Python offers some out-of-the-box CSV support in the `csv` module²⁵.

DB A *database* is an organized collection of structured information or data, typically stored electronically in a computer system. Databases are discussed in our book *Databases*⁹¹.

DBMS A *database management system* is the software layer located between the user or application and the DB. The DBMS allows the user/application to create, read, write, update, delete, and otherwise manipulate the data in the DB⁹⁵.

DBS A *database system* is the combination of a DB and a the corresponding DBMS, i.e., basically, an installation of a DBMS on a computer together with one or multiple DBs. DBS = DB + DBMS.

Git is a distributed Version Control Systems (VCS) which allows multiple users to work on the same code while preserving the history of the code changes^{78,88}. Learn more at <https://git-scm.com>.

GitHub is a website where software projects can be hosted and managed via the Git VCS^{68,88}. Learn more at <https://github.com>.

Glossary (in English) II



IT information technology

JavaScript JavaScript is the predominant programming language used in websites to develop interactive contents for display in browsers³³.

JSON *JavaScript Object Notation* is a data interchange format^{16,86} based on JavaScript³³ syntax.

LAMP Stack A system setup for web applications: Linux, Apache (a web server), MySQL, and the server-side scripting language PHP^{18,45}.

LibreOffice is an open source office suite^{36,55,75} which is a good and free alternative to Microsoft Office. It offers software such as LibreOffice Writer, LibreOffice Calc, and LibreOffice Base. See⁹¹ for more information and installation instructions.

LibreOffice Base is a DBMS that can work on stand-alone files but also connect to other popular relational databases^{34,75}. It is part of LibreOffice^{36,55,75} and has functionality that is comparable to Microsoft Access^{7,20,89}.

LibreOffice Calc is a spreadsheet software that allows you to arrange and perform calculations with data in a tabular grid. It is a free and open source spreadsheet software^{55,75}, i.e., an alternative to Microsoft Excel. It is part of LibreOffice^{36,55,75}.

LibreOffice Writer is a free and open source text writing program⁹⁶ and part of LibreOffice^{36,55,75}. It is a good alternative to Microsoft Word.

Linux is the leading open source operating system, i.e., a free alternative for Microsoft Windows^{5,43,77,87,90}. We recommend using it for this course, for software development, and for research. Learn more at <https://www.linux.org>. Its variant Ubuntu is particularly easy to use and install.

MariaDB An open source relational database management system that has forked off from MySQL^{1,2,6,32,58,71}. See <https://mariadb.org> for more information.

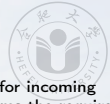
Microsoft Access is a DBMS that can work on DBs stored in single, stand-alone files but also connect to other popular relational databases^{7,20,59,89}. It is part of Microsoft Office. A free and open source alternative to this commercial software is LibreOffice Base.

Glossary (in English) III



- Microsoft Excel is a spreadsheet program that allows users to store, organize, manipulate, and calculate data in tabular structures^{12,37,53}. It is part of Microsoft Office. A free alternative to this commercial software is LibreOffice Calc^{55,75}.
- Microsoft Office is a commercial suite of office software, including Microsoft Excel, Microsoft Word, and Microsoft Access⁵³. LibreOffice is a free and open source alternative.
- Microsoft Windows is a commercial proprietary operating system¹⁴. It is widely spread, but we recommend using a Linux variant such as Ubuntu for software development and for our course. Learn more at <https://www.microsoft.com/windows>.
- Microsoft Word is one of the leading text writing programs^{30,61,96} and part of Microsoft Office. A free alternative to this commercial software is the LibreOffice Writer.
- MySQL An open source relational database management system^{13,32,72,85,94}. MySQL is famous for its use in the LAMP Stack. See <https://www.mysql.com> for more information.
- OSS Open source software, i.e., software that can freely be used, whose source code is made available in the internet, and which is usually developed cooperatively over the internet as well⁴⁶. Typical examples are Python, Linux, Git, and PostgreSQL.
- PostgreSQL An open source object-relational DBMS^{35,65,69,85}. See <https://postgresql.org> for more information.
- psql is the client program used to access the PostgreSQL DBMS server.
- Python The Python programming language^{47,54,57,92}, i.e., what you will learn about in our book⁹². Learn more at <https://python.org>.
- relational database A relational DB is a database that organizes data into rows (tuples, records) and columns (attributes), which collectively form tables (relations) where the data points are related to each other^{22,41,42,79,83,91,93}.

Glossary (in English) IV



server In a client-server architecture, the server is a process that fulfills the requests of the clients. It usually waits for incoming communication carrying the requests from the clients. For each request, it takes the necessary actions, performs the required computations, and then sends a response with the result of the request. Typical examples for servers are web servers¹⁸ in the internet as well as DBMSes. It is also common to refer to the computer running the server processes as server as well, i.e., to call it the „server computer“⁵².

SQL The *Structured Query Language* is basically a programming language for querying and manipulating relational databases^{19,26–28,48,60,80–83}. It is understood by many DBMSes. You find the SQL commands supported by PostgreSQL in the reference⁸⁰.

terminal A terminal is a text-based window where you can enter commands and execute them^{5,21}. Knowing what a terminal is and how to use it is very essential in any programming- or system administration-related task. If you want to open a terminal under Microsoft Windows, you can Druck auf  + , dann Schreiben von `cmd`, dann Druck auf . Under Ubuntu Linux,  +  +  opens a terminal, which then runs a Bash shell inside.

Ubuntu is a variant of the open source operating system Linux^{21,45}. We recommend that you use this operating system to follow this class, for software development, and for research. Learn more at <https://ubuntu.com>. If you are in China, you can download it from <https://mirrors.ustc.edu.cn/ubuntu-releases>.

VCS A *Version Control System* is a software which allows you to manage and preserve the historical development of your program code⁸⁸. A distributed VCS allows multiple users to work on the same code and upload their changes to the server, which then preserves the change history. The most popular distributed VCS is Git.

WWW World Wide Web^{9,29}

Glossary (in English) V



XML The *Extensible Markup Language* is a text-based language for storing and transporting of data^{17,24,50}. It allows you to define elements in the form `<myElement myAttr="x">...text...</myElement>`. Different from comma-separated values (CSV), elements in XML can be hierarchically nested, like `<a><b><c>test</c></b><b>bla</b></a>`, and thus easily represent tree structures. XML is one of most-used data interchange formats. To process XML in Python, use the `defusedxml` library⁴⁴, as it protects against several security issues.

YAML *YAML Ain't Markup Language*TM is a human-friendly data serialization language for all programming languages^{23,31,50}. It is widely used for configuration files in the DevOps environment. See <https://yaml.org> for more information.