



合肥大學
HEFEI UNIVERSITY



Datenbanken

1. Organisation

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. Willkommen
2. Lehrer
3. Zeitplan
4. Aspekte
5. Lehrmaterial
6. Ökosystem
7. Voraussetzungen
8. Zusammenfassung





Willkommen



Willkommen

- Willkommen zum Kurs **Datenbanken**.



Willkommen



- Willkommen zum Kurs **Datenbanken**.
- Wir befinden uns im Herbst-Winter-Semester 2025 an der Universität Hefei (合肥大学) in der schönen Stadt Hefei (合肥市) in der Provinz Anhui (安徽省) in China.

Willkommen



- Willkommen zum Kurs **Datenbanken**.
- Wir befinden uns im Herbst-Winter-Semester 2025 an der Universität Hefei (合肥大学) in der schönen Stadt Hefei (合肥市) in der Provinz Anhui (安徽省) in China.
- Dieser Kurs lehrt die Verwendung von Datenbanken.

Willkommen



- Willkommen zum Kurs **Datenbanken**.
- Wir befinden uns im Herbst-Winter-Semester 2025 an der Universität Hefei (合肥大学) in der schönen Stadt Hefei (合肥市) in der Provinz Anhui (安徽省) in China.
- Dieser Kurs lehrt die Verwendung von Datenbanken.
- Der Kurs wird in deutscher Sprache unterrichtet.



Lehrer



Lehrer

- Prof. Dr. Thomas WEISE (汤卫思)



Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



© Ewald Judd CC BY 4.0

- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



© Ewald Judd CC BY 4.0



© Kolossos CC BY-SA 3.0

Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



© Ewald Judt CC BY 4.0



© Ewald Judt CC BY 4.0



© Kolossos CC BY-SA 3.0

Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



© Ewald Judt CC BY 4.0



© Ewald Judt CC BY 4.0



© Kolossos CC BY-SA 3.0

Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



© Ewald Judt CC BY 4.0



© eagle77 Pixabay Content License



© Kolossos CC BY-SA 3.0

Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



© Ewald Judt CC BY 4.0



© eagle77 Pixabay Content License



© Kolossos CC BY-SA 3.0

© eagle77 Pixabay Content License

Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



© Ewald Judt CC BY 4.0

© eagle77 Pixabay Content License

© Firstclasspixel Pixabay Content License

© Kolossos CC BY-SA 3.0

© eagle77 Pixabay Content License

Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.

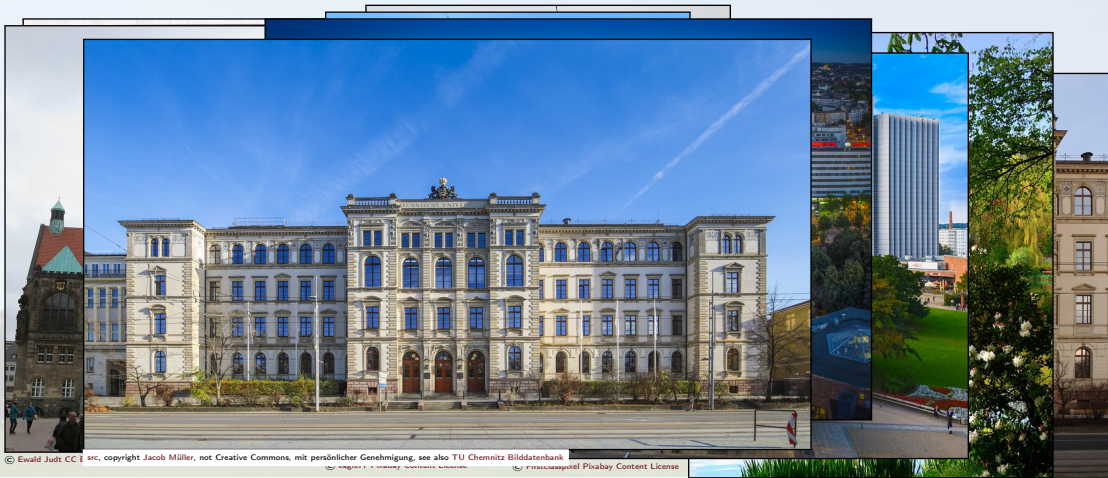


src, copyright Jacob Müller, not Creative Commons, mit persönlicher Genehmigung, see also TU Chemnitz Bilddatenbank

Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



src, copyright Jacob Müller, not Creative Commons, mit persönlicher Genehmigung, see also TU Chemnitz Bilddatenbank

© Ewald Judt CC BY-SA 3.0, src, copyright Jacob Müller, not Creative Commons, mit persönlicher Genehmigung, see also TU Chemnitz Bilddatenbank

© egle77 Pixabay Content License

© egle77 Pixabay Content License

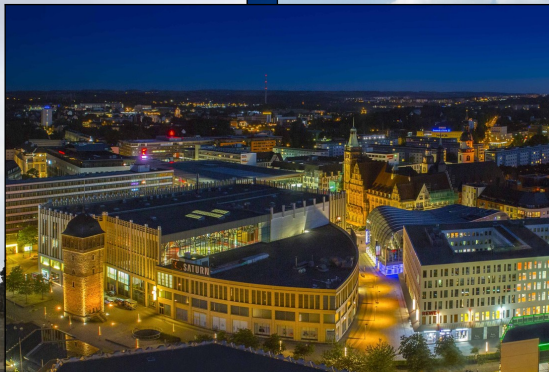
© Kolossos CC BY-SA 3.0

© egle77 Pixabay Content License

Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



src, copyright Jacob Müller, not Creative Commons, mit persönlicher Genehmigung, see also TU Chemnitz Bilddatenbank



Genehmigung, see also TU Chemnitz Bilddatenbank

© Ewald Judt CC BY-SA 3.0 src, copyright Jacob Müller, not Creative Commons, mit persönlicher Genehmigung, see also TU Chemnitz Bilddatenbank

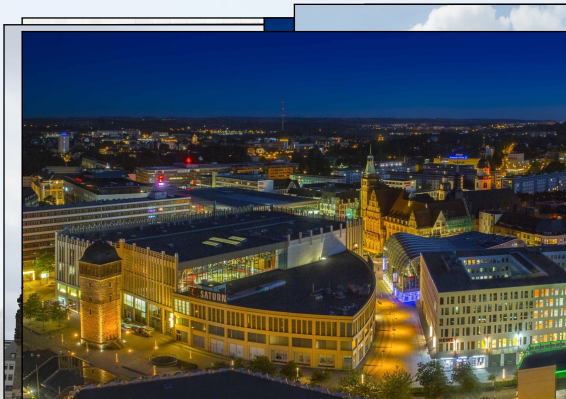
© Ewald Judt CC BY-SA 3.0

© Ewald Judt CC BY-SA 3.0

© Kolossos CC BY-SA 3.0

© eagle77 Pixabay Content License

- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



src, copyright Jacob Müller, not Creative Commons, mit persönlicher Genehmigung, see also TU Chemnitz Bilddatenbank



src, copyright Jacob Müller, not Creative Commons, mit persönlicher Genehmigung, see also TU Chemnitz Bilddatenbank

Genehmigung, see also TU Chemnitz Bilddatenbank



© Ewald Judt CC BY-SA 4.0

src, copyright Jacob Müller, not Creative Commons, mit persönlicher Genehmigung, see also TU Chemnitz Bilddatenbank

© Ewald Judt CC BY-SA 4.0

© Pixabay Content License



© Kolossos CC BY-SA 3.0

© eagle77 Pixabay Content License

Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



© till_westhof Pixabay Content License



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



© till_westhof Pixabay Content License



© webandi Pixabay Content License

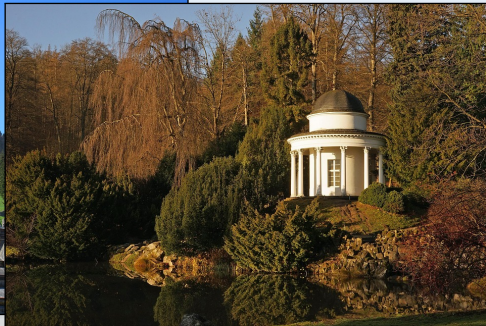
- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



© till_westhof Pixabay Content License



© webandi Pixabay Content License



© Barni1 Pixabay Content License

- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



© till_westhof Pixabay Content License



© webar

© Barn11 Pixabay Content License



ense

Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



U N I K A S S E L
V E R S I T Ä T

Copyright Universität Kassel, mit Genehmigung der Präseskone der Universität Kassel
© till_westhof Pixabay Content License

© webar

© Barni1 Pixabay Content License

- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.

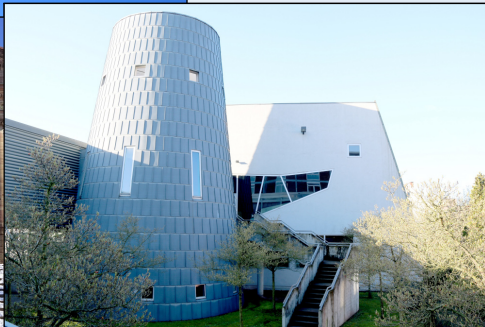


U N I K A S S E L
V E R S I T Ä T

Copyright Universität Kassel, mit Genehmigung der Pressestelle der Universität Kassel
© till_westhof Pixabay Content License



© webar



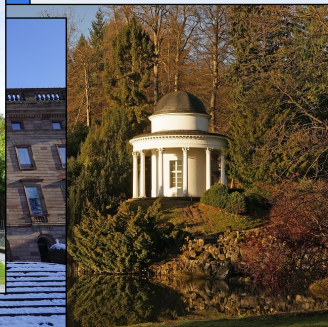
Copyright Universität Kassel, mit Genehmigung der Pressestelle der Universität Kassel

© Barni1 Pixabay Content License

Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



U N I K A S S
V E R S I T Ä T

Copyright Universität Kassel, mit Genehmigung der Pressestelle der Universität
© till_westhof Pixabay Content License

Copyright Universität Kassel, mit Genehmigung der Pressestelle der Universität

© webar

© Barni1 Pixabay Content License

Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



UNI KASSEL
VERSITÄT

Copyright Universität Kassel, mit Genehmigung der Pressestelle der Universität Kassel
© till_westhof Pixabay Content License



Copyright Universität Kassel, mit Genehmigung der

© webar



Copyright Universität Kassel, mit Genehmigung der Pressestelle der Universität Kassel

- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



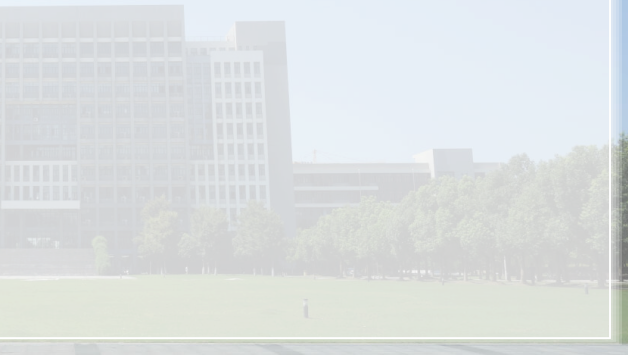
Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.

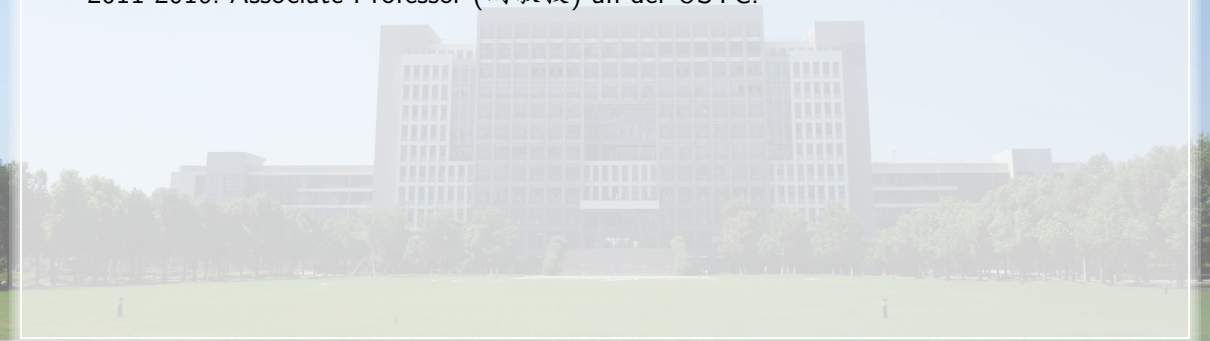


- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.





- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.
- 2011-2016: Associate Professor (副教授) an der USTC.





- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.
- 2011-2016: Associate Professor (副教授) an der USTC.
- Seit 2016: Full Professor (教授) und Direktor des Institute of Applied Optimization (应用优化研究所) der Universität Hefei (合肥大学) in Hefei, Anhui, China.





- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.
- 2011-2016: Associate Professor (副教授) an der USTC.
- Seit 2016: Full Professor (教授) und Direktor des Institute of Applied Optimization (应用优化研究所) der Universität Hefei (合肥大学) in Hefei, Anhui, China.
- Forschungsfeld: Optimierung, Operations Research.



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.
- 2011-2016: Associate Professor (副教授) an der USTC.
- Seit 2016: Full Professor (教授) und Direktor des Institute of Applied Optimization (应用优化研究所) der Universität Hefei (合肥大学) in Hefei, Anhui, China.
- Forschungsfeld: Optimierung, Operations Research.
- Kontakt: tweise@hfuu.edu.cn.



Zeitplan



Zeitplan: Datenbanken 2025



Vorlesung						Übung					
周次	月-日	星期	节	Zeit	Raum	周次	月-日	星期	节	Zeit	Raum
08	10-28	Di	3-4	10:20-12:00	36-212	08	10-28	Di	5-6	14:00-15:40	37-410
08	10-29	Mi	3-4	10:20-12:00	36-202						
08	10-31	Fr	3-4	10:20-12:00	36-202						
09	11-04	Di	3-4	10:20-12:00	36-212	09	11-04	Di	5-6	14:00-15:40	37-410
09	11-05	Mi	3-4	10:20-12:00	36-202						
09	11-07	Fr	3-4	10:20-12:00	36-202						
10	11-11	Di	3-4	10:20-12:00	36-212	10	11-11	Di	5-6	14:00-15:40	37-410
10	11-12	Mi	3-4	10:20-12:00	36-202	10	11-13	Do	5-6	14:00-15:40	37-410
11	11-17	Mo	3-4	10:20-12:00	36-204	11	11-18	Di	5-6	14:00-15:40	37-410
11	11-19	Mi	3-4	10:20-12:00	36-202						
12	11-24	Mo	3-4	10:20-12:00	36-204	12	11-25	Di	5-6	14:00-15:40	37-410
12	11-26	Mi	3-4	10:20-12:00	36-202						
13	12-01	Mo	3-4	10:20-12:00	36-204	13	12-02	Di	5-6	14:00-15:40	37-410
13	12-03	Mi	3-4	10:20-12:00	36-202						
14	12-08	Mo	3-4	10:20-12:00	36-204	14	12-09	Di	5-6	14:00-15:40	37-410
14	12-10	Mi	3-4	10:20-12:00	36-202						



Aspekte



Aspekte



- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite

- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite:
 1. Alles, was gelehrt wird, wird aktiv praktisch und selbstständig von den Studierenden ausprobiert.

- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite:
 1. Alles, was gelehrt wird, wird aktiv praktisch und selbstständig von den Studierenden ausprobiert. **Auf Ihrem eigenen Laptop-Computer.**



- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite:
 1. Alles, was gelehrt wird, wird aktiv praktisch und selbstständig von den Studierenden ausprobiert. **Auf Ihrem eigenen Laptop-Computer.**
 2. Umfangreiches unterstützendes Lehrmaterial wird zur Verfügung gestellt.



- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite:
 1. Alles, was gelehrt wird, wird aktiv praktisch und selbstständig von den Studierenden ausprobiert. **Auf Ihrem eigenen Laptop-Computer.**
 2. Umfangreiches unterstützendes Lehrmaterial wird zur Verfügung gestellt. Alles kann nachgelesen und nachvollzogen werden.

- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite:
 1. Alles, was gelehrt wird, wird aktiv praktisch und selbstständig von den Studierenden ausprobiert. **Auf Ihrem eigenen Laptop-Computer.**
 2. Umfangreiches unterstützendes Lehrmaterial wird zur Verfügung gestellt. Alles kann nachgelesen und nachvollzogen werden.
 3. Das ist keine Arbeit, kein Auswendiglernen.

- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite:
 1. Alles, was gelehrt wird, wird aktiv praktisch und selbstständig von den Studierenden ausprobiert. **Auf Ihrem eigenen Laptop-Computer.**
 2. Umfangreiches unterstützendes Lehrmaterial wird zur Verfügung gestellt. Alles kann nachgelesen und nachvollzogen werden.
 3. Das ist keine Arbeit, kein Auswendiglernen. Es soll nicht langweilig sein.

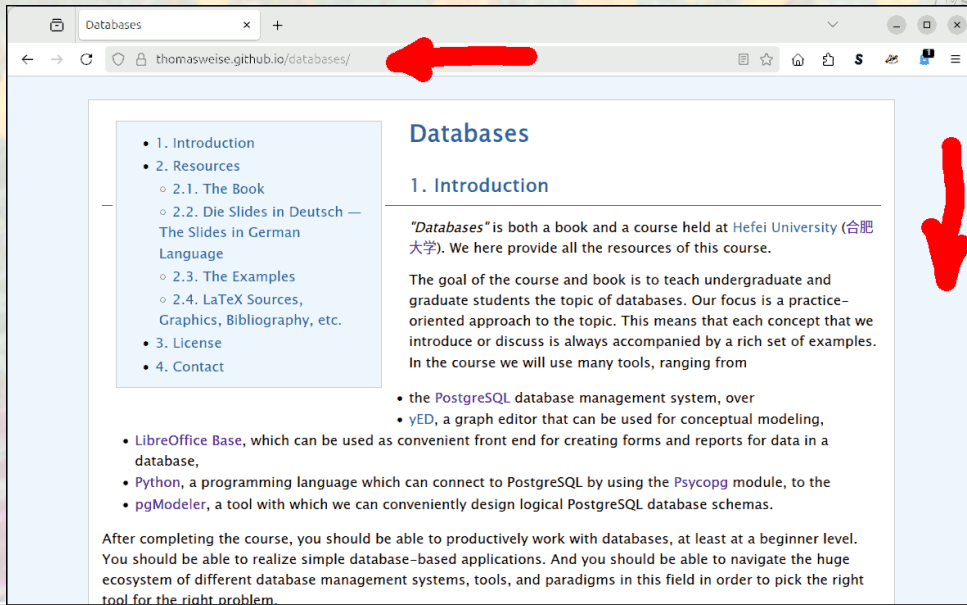
- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite:
 1. Alles, was gelehrt wird, wird aktiv praktisch und selbstständig von den Studierenden ausprobiert. **Auf Ihrem eigenen Laptop-Computer.**
 2. Umfangreiches unterstützendes Lehrmaterial wird zur Verfügung gestellt. Alles kann nachgelesen und nachvollzogen werden.
 3. Das ist keine Arbeit, kein Auswendiglernen. Es soll nicht langweilig sein. Das Ziel ist, dass Sie Freude beim Lernen und bei der Arbeit mit unserem Themengebiet fühlen.



Lehrmaterial







Databases

thomasweise.github.io/databases/

Databases

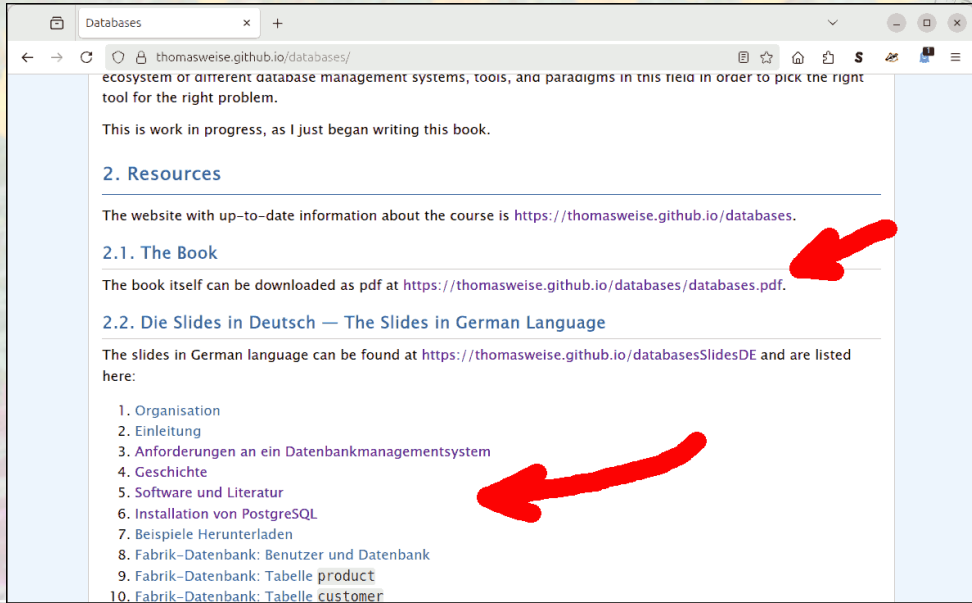
1. Introduction

"Databases" is both a book and a course held at [Hefei University](#) (合肥大学). We here provide all the resources of this course.

The goal of the course and book is to teach undergraduate and graduate students the topic of databases. Our focus is a practice-oriented approach to the topic. This means that each concept that we introduce or discuss is always accompanied by a rich set of examples. In the course we will use many tools, ranging from

- the [PostgreSQL](#) database management system, over
- [yED](#), a graph editor that can be used for conceptual modeling,
- [LibreOffice Base](#), which can be used as convenient front end for creating forms and reports for data in a database,
- [Python](#), a programming language which can connect to PostgreSQL by using the [Psycopg](#) module, to the
- [pgModeler](#), a tool with which we can conveniently design logical PostgreSQL database schemas.

After completing the course, you should be able to productively work with databases, at least at a beginner level. You should be able to realize simple database-based applications. And you should be able to navigate the huge ecosystem of different database management systems, tools, and paradigms in this field in order to pick the right tool for the right problem.



ecosystem or different database management systems, tools, and paradigms in this field in order to pick the right tool for the right problem.

This is work in progress, as I just began writing this book.

2. Resources

The website with up-to-date information about the course is <https://thomasweise.github.io/databases/>.

2.1. The Book

The book itself can be downloaded as pdf at <https://thomasweise.github.io/databases/databases.pdf>.

2.2. Die Slides in Deutsch — The Slides in German Language

The slides in German language can be found at <https://thomasweise.github.io/databasesSlidesDE> and are listed here:

1. Organisation
2. Einleitung
3. Anforderungen an ein Datenbankmanagementsystem
4. Geschichte
5. Software und Literatur
6. Installation von PostgreSQL
7. Beispiele Herunterladen
8. Fabrik-Datenbank: Benutzer und Datenbank
9. Fabrik-Datenbank: Tabelle product
10. Fabrik-Datenbank: Tabelle customer

Buch



- Ich habe für diesen Kurs ein Buch geschrieben: *Databases*¹¹⁷.

Buch



- Ich habe für diesen Kurs ein Buch geschrieben: *Databases*¹¹⁷.
- Das Buch ist in Englisch geschrieben.

Buch



- Ich habe für diesen Kurs ein Buch geschrieben: *Databases*¹¹⁷.
- Das Buch ist in Englisch geschrieben.
- Dieses Buch steht kostenlos im Internet.

Buch



- Ich habe für diesen Kurs ein Buch geschrieben: *Databases*¹¹⁷.
- Das Buch ist in Englisch geschrieben.
- Dieses Buch steht kostenlos im Internet.
- Es wird noch aktiv weiterentwickelt.

Buch



- Ich habe für diesen Kurs ein Buch geschrieben: *Databases*¹¹⁷.
- Das Buch ist in Englisch geschrieben.
- Dieses Buch steht kostenlos im Internet.
- Es wird noch aktiv weiterentwickelt.
- **Sie** können mir helfen, es weiterzuentwickeln, durch Fragen, Anregungen, das Finden von Fehlern, usw.

Buch



- Ich habe für diesen Kurs ein Buch geschrieben: *Databases*¹¹⁷.
- Das Buch ist in Englisch geschrieben.
- Dieses Buch steht kostenlos im Internet.
- Es wird noch aktiv weiterentwickelt.
- **Sie** können mir helfen, es weiterzuentwickeln, durch Fragen, Anregungen, das Finden von Fehlern, usw.

URL: <https://thomasweise.github.io/databases/databases.pdf>

Buch: Features

- Das Buch ist explizit für Anfänger auf dem Gebiet geschrieben.



Buch: Features



- Das Buch ist explizit für Anfänger auf dem Gebiet geschrieben.
- Es beinhaltet viele Beispiele, die Schritt-für-Schritt erklärt werden.

Buch: Features



- Das Buch ist explizit für Anfänger auf dem Gebiet geschrieben.
- Es beinhaltet viele Beispiele, die Schritt-für-Schritt erklärt werden.
- Fachbegriffe, die Sie vielleicht nicht kennen, werden in einem Glossary erklärt.

Buch: Features



- Das Buch ist explizit für Anfänger auf dem Gebiet geschrieben.
- Es beinhaltet viele Beispiele, die Schritt-für-Schritt erklärt werden.
- Fachbegriffe, die Sie vielleicht nicht kennen, werden in einem Glossary erklärt.
- Das Glossary beinhaltet auch Abkürzungen, Programme, mathematische Notationen, usw.

Buch: Features



- Das Buch ist explizit für Anfänger auf dem Gebiet geschrieben.
- Es beinhaltet viele Beispiele, die Schritt-für-Schritt erklärt werden.
- Fachbegriffe, die Sie vielleicht nicht kennen, werden in einem Glossary erklärt.
- Das Glossary beinhaltet auch Abkürzungen, Programme, mathematische Notationen, usw.
- Zum Beispiel `stdin`, UCS, UML, HTML, JSON, 1NF, breakpoint, `venv`, client-server architecture, $\mathbb{R}$, $\mathcal{NP}$ -hard, `psql`, `PyPI`, `Matplotlib`...

Buch: Features



- Das Buch ist explizit für Anfänger auf dem Gebiet geschrieben.
- Es beinhaltet viele Beispiele, die Schritt-für-Schritt erklärt werden.
- Fachbegriffe, die Sie vielleicht nicht kennen, werden in einem Glossary erklärt.
- Das Glossary beinhaltet auch Abkürzungen, Programme, mathematische Notationen, usw.
- Zum Beispiel `stdin`, UCS, UML, HTML, JSON, 1NF, breakpoint, `venv`, client-server architecture, $\mathbb{R}$, $\mathcal{NP}$ -hard, `psql`, `PyPI`, `Matplotlib`...
- Diese Terme sind jeweils im Text auf das Glossary verlinkt.

Buch: Features



- Das Buch ist explizit für Anfänger auf dem Gebiet geschrieben.
- Es beinhaltet viele Beispiele, die Schritt-für-Schritt erklärt werden.
- Fachbegriffe, die Sie vielleicht nicht kennen, werden in einem Glossary erklärt.
- Das Glossary beinhaltet auch Abkürzungen, Programme, mathematische Notationen, usw.
- Zum Beispiel `stdin`, UCS, UML, HTML, JSON, 1NF, breakpoint, `venv`, client-server architecture, $\mathbb{R}$, $\mathcal{NP}$ -hard, `psql`, `PyPI`, `Matplotlib`...
- Diese Terme sind jeweils im Text auf das Glossary verlinkt.
- Und im Glossary gibt es dann immer weiterführende Literaturhinweise.

Slides und Übungen/Fragen



- Die Slides zu diesem Kurs sind ebenfalls auf der Kurswebseite online.

Slides und Übungen/Fragen



- Die Slides zu diesem Kurs sind ebenfalls auf der Kurswebseite online.
- Ebenso finden Sie eine Liste mit Übungsfragen auf der Kurswebseite.
- URL: <https://thomasweise.github.io/databases>

Beispiele



- In diesem Kurs werden wir sehr viele Beispiele verwenden.

Beispiele



- In diesem Kurs werden wir sehr viele Beispiele verwenden.
 1. Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem^{38,73,79,100}.

Beispiele



- In diesem Kurs werden wir sehr viele Beispiele verwenden.
 1. Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem ^{38,73,79,100}.
 2. Wir werden echte Datenbanken damit erstellen und mit der Datenbanksprache SQL ^{17,69} darauf zugreifen.

Beispiele



- In diesem Kurs werden wir sehr viele Beispiele verwenden.
 1. Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem^{38,73,79,100}.
 2. Wir werden echte Datenbanken damit erstellen und mit der Datenbanksprache SQL^{17,69} darauf zugreifen.
 3. Wir werden die Strukturen von Datenbanken mit den Werkzeugen yEd^{86,124} und PgModeler² modellieren.

Beispiele



- In diesem Kurs werden wir sehr viele Beispiele verwenden.
 1. Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem^{38,73,79,100}.
 2. Wir werden echte Datenbanken damit erstellen und mit der Datenbanksprache SQL^{17,69} darauf zugreifen.
 3. Wir werden die Strukturen von Datenbanken mit den Werkzeugen yEd^{86,124} und PgModeler² modellieren.
 4. Wir werden von der Programmiersprache Python¹¹⁸ aus mit dem psycopg-Modul¹¹⁴ auf unsere PostgreSQL Datenbanken zugreifen.

Beispiele



- In diesem Kurs werden wir sehr viele Beispiele verwenden.
 1. Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem^{38,73,79,100}.
 2. Wir werden echte Datenbanken damit erstellen und mit der Datenbanksprache SQL^{17,69} darauf zugreifen.
 3. Wir werden die Strukturen von Datenbanken mit den Werkzeugen yEd^{86,124} und PgModeler² modellieren.
 4. Wir werden von der Programmiersprache Python¹¹⁸ aus mit dem psycopg-Modul¹¹⁴ auf unsere PostgreSQL Datenbanken zugreifen.
 5. Wir werden LibreOffice Base^{37,87} verwenden, um Formulare und Berichte für unsere PostgreSQL Datenbanken zu erstellen.

Beispiele



- In diesem Kurs werden wir sehr viele Beispiele verwenden.
 1. Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem^{38,73,79,100}.
 2. Wir werden echte Datenbanken damit erstellen und mit der Datenbanksprache SQL^{17,69} darauf zugreifen.
 3. Wir werden die Strukturen von Datenbanken mit den Werkzeugen yEd^{86,124} und PgModeler² modellieren.
 4. Wir werden von der Programmiersprache Python¹¹⁸ aus mit dem psycopg-Modul¹¹⁴ auf unsere PostgreSQL Datenbanken zugreifen.
 5. Wir werden LibreOffice Base^{37,87} verwenden, um Formulare und Berichte für unsere PostgreSQL Datenbanken zu erstellen.
- Alle Beispiele sind in dem GitHub Repository <https://github.com/thomasWeise/databasesCode> bereitgestellt.

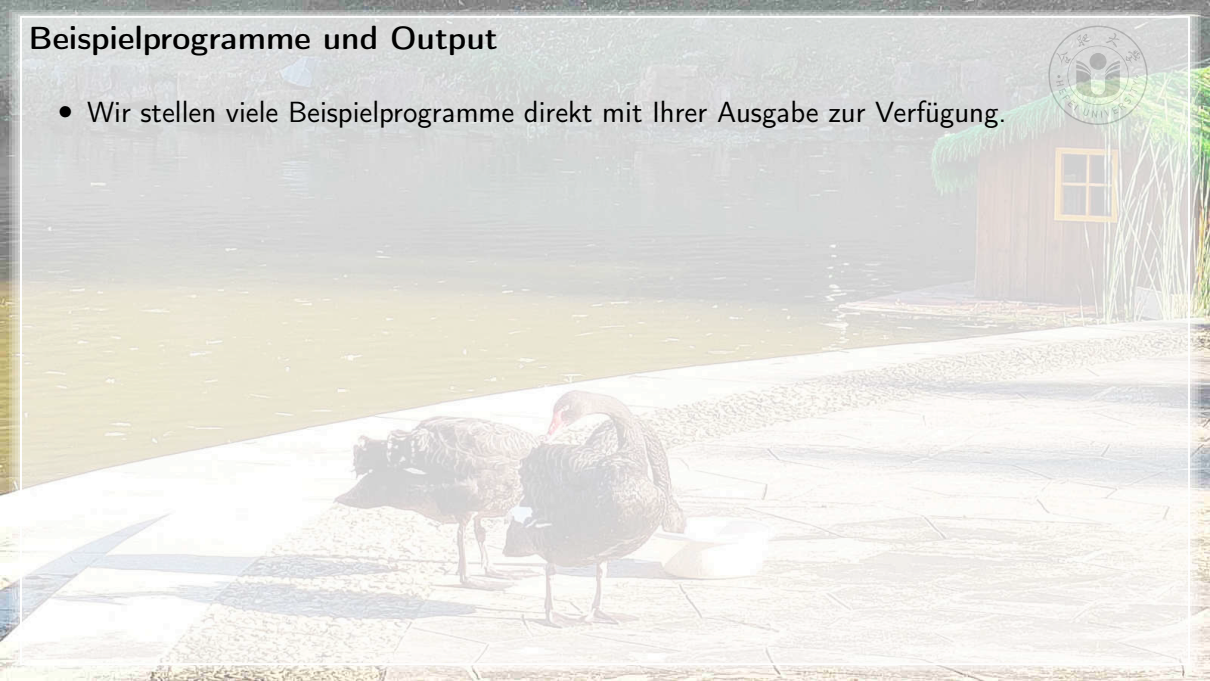
Beispiele



- In diesem Kurs werden wir sehr viele Beispiele verwenden.
 1. Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem^{38,73,79,100}.
 2. Wir werden echte Datenbanken damit erstellen und mit der Datenbanksprache SQL^{17,69} darauf zugreifen.
 3. Wir werden die Strukturen von Datenbanken mit den Werkzeugen yEd^{86,124} und PgModeler² modellieren.
 4. Wir werden von der Programmiersprache Python¹¹⁸ aus mit dem psycopg-Modul¹¹⁴ auf unsere PostgreSQL Datenbanken zugreifen.
 5. Wir werden LibreOffice Base^{37,87} verwenden, um Formulare und Berichte für unsere PostgreSQL Datenbanken zu erstellen.
- Alle Beispiele sind in dem GitHub Repository <https://github.com/thomasWeise/databasesCode> bereitgestellt.
- Sie können (und sollen!) also jeden Schritt jedes Beispiels genau nachvollziehen.

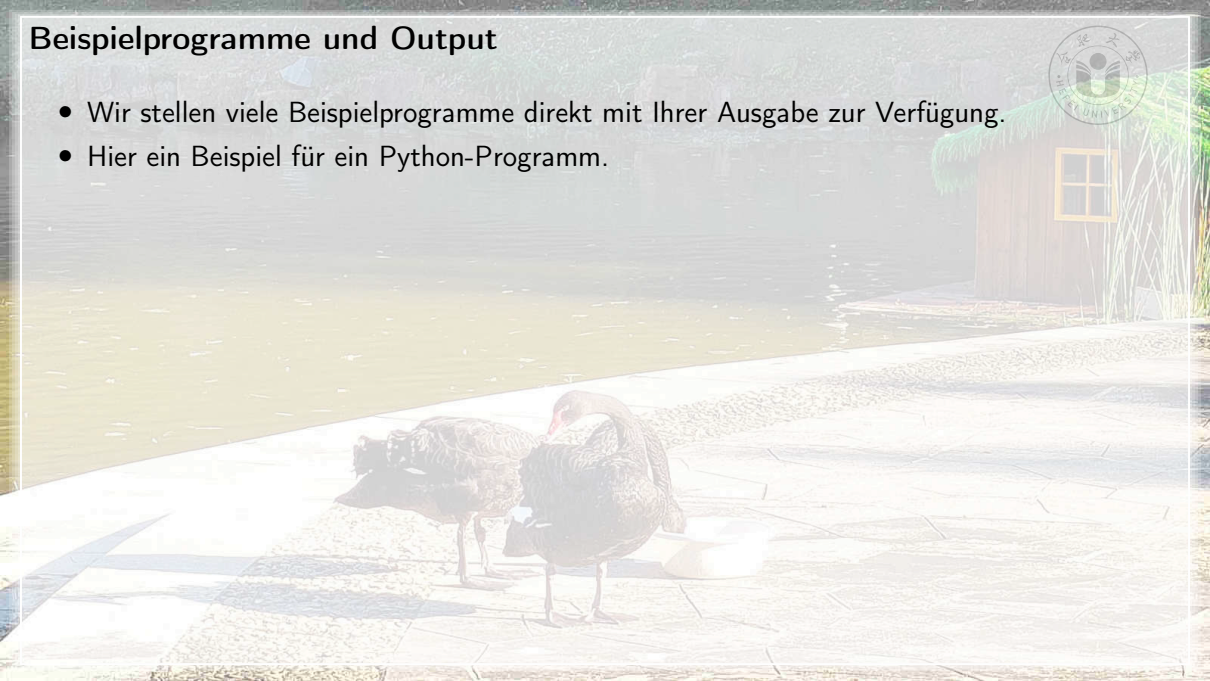
Beispielprogramme und Output

- Wir stellen viele Beispielprogramme direkt mit Ihrer Ausgabe zur Verfügung.



Beispielprogramme und Output

- Wir stellen viele Beispielprogramme direkt mit Ihrer Ausgabe zur Verfügung.
- Hier ein Beispiel für ein Python-Programm.



Beispielprogramme und Output



- Wir stellen viele Beispielprogramme direkt mit Ihrer Ausgabe zur Verfügung.
- Hier ein Beispiel für ein Python-Programm.

```
1 print("Hello World!")
```

↓ `python3 very_first_program.py` ↓

```
1 Hello World!
```

Beispielprogramme und Output



- Wir stellen viele Beispielprogramme direkt mit Ihrer Ausgabe zur Verfügung.
- Hier ein Beispiel für ein Python-Programm.
- Im Buch finden Sie jeweils im Titel des Programms gleich noch den Link zum Sourcecode auf GitHub.

```
1 print("Hello World!")
```

↓ `python3 very_first_program.py` ↓

```
1 Hello World!
```

Komplexere Beispielprogramme



Link zur Datei auf Github

- Für einige Beispiele mit komplexeren Kommandos werden der Programmcode, die Kommandozeile zu dessen Ausführung, der standard output stream (stdout)⁵⁷, der Exit-Code⁵⁵ und die Softwareversion mit angezeigt.

Listing 7.1: Using SQL to create a the user boss with password superboss123. (stored in file create_user.sql; output in Listing 7.2)

```
1  /* In this example, we create a new user for our database. */
2
3  -- On PostgreSQL, there is a table 'pg_catalog.pg_user' with all users.
4  -- We print the column 'username' with the user names.
5  SELECT username FROM pg_catalog.pg_user;
6
7  -- Create the user 'boss'.
8  -- He will be the owner of the database that we will create.
9  CREATE USER boss WITH ENCRYPTED PASSWORD 'superboss123';
10
11 -- Now there is a new user: 'boss'.
12 SELECT username FROM pg_catalog.pg_user;
```

Listing 7.2: The standard output stream (stdout) resulting from the SQL statements in create_user.sql given in Listing 7.1.

```
1  $ psql "postgres://postgres:XXX@localhost" -v ON_ERROR_STOP=1 -ebf
2  ↪ create_user.sql
3  username
4  -----
5  postgres
6  (1 row)
7
8  CREATE ROLE
9  username
10 -----
11 postgres
12 boss
13 (2 rows)
14
15 # psql 16.9 (Ubuntu 16.9-0ubuntu0.24.04.1) succeeded with exit code 0.
```

Kommando um die Datei auszuführen

Softwareversion, Ergebnis, und Exit Code

Komplexere Beispielprogramme



Link zur Datei auf Github

- Für einige Beispiele mit komplexeren Kommandos werden der Programmcode, die Kommandozeile zu dessen Ausführung, der standard output stream (stdout)⁵⁷, der Exit-Code⁵⁵ und die Softwareversion mit angezeigt.
- Natürlich mit einem Link zum Code auf GitHub.

Listing 7.1: Using SQL to create a the user boss with password superboss123. (stored in file create_user.sql; output in Listing 7.2)

```
1  /* In this example, we create a new user for our database. */
2
3  -- On PostgreSQL, there is a table 'pg_catalog.pg_user' with all users.
4  -- We print the column 'username' with the user names.
5  SELECT username FROM pg_catalog.pg_user;
6
7  -- Create the user 'boss'.
8  -- He will be the owner of the database that we will create.
9  CREATE USER boss WITH ENCRYPTED PASSWORD 'superboss123';
10
11 -- Now there is a new user: 'boss'.
12 SELECT username FROM pg_catalog.pg_user;
```

Listing 7.2: The standard output stream (stdout) resulting from the SQL statements in create_user.sql given in Listing 7.1.

```
1  $ psql "postgres://postgres:XXX@localhost" -v ON_ERROR_STOP=1 -ebf
2  ↪ create_user.sql
3  username
4  -----
5  postgres
6  (1 row)
7
8  CREATE ROLE
9  username
10 -----
11 postgres
12 boss
13 (2 rows)
14
15 # psql 16.9 (Ubuntu 16.9-0ubuntu0.24.04.1) succeeded with exit code 0.
```

Kommando um die Datei auszuführen

Softwareversion, Ergebnis, und Exit Code

- Wir nutzen viele verschiedene Werkzeuge.

Software



- Wir nutzen viele verschiedene Werkzeuge.
- Diese sind alle kostenlos im Internet verfügbar.

Software



- Wir nutzen viele verschiedene Werkzeuge.
- Diese sind alle kostenlos im Internet verfügbar.
- Die meisten sind Open Source software.



- Wir nutzen viele verschiedene Werkzeuge.
- Diese sind alle kostenlos im Internet verfügbar.
- Die meisten sind Open Source software.
- Für jedes Werkzeug haben wir Installationshinweise ausgearbeitet.



- Wir nutzen viele verschiedene Werkzeuge.
- Diese sind alle kostenlos im Internet verfügbar.
- Die meisten sind Open Source software.
- Für jedes Werkzeug haben wir Installationshinweise ausgearbeitet. Mit Screenshots für Microsoft Windows^{4,45,92,104,113} und Ubuntu Linux^{23,46}.

Literaturhinweise



- Haben Sie die kleinen roten Zahlen in dem Satz „*Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem* ^{38,73,79,100}.“ bemerkt?

Literaturhinweise



- Haben Sie die kleinen roten Zahlen in dem Satz „*Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem*^{38,73,79,100}.“ bemerkt?
- Dies sind Literaturhinweise, die am Ende der Slides gelistet werden.

Literaturhinweise



- Haben Sie die kleinen roten Zahlen in dem Satz „*Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem*^{38,73,79,100}.“ bemerkt?
- Dies sind Literaturhinweise, die am Ende der Slides gelistet werden.
- Im Buch *Databases*¹¹⁷ finden Sie ebenfalls viele Literaturhinweise, dort allerdings in eckigen Klammern, also eher so: [117].



- Haben Sie die kleinen roten Zahlen in dem Satz „*Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem*^{38,73,79,100}.“ bemerkt?
- Dies sind Literaturhinweise, die am Ende der Slides gelistet werden.
- Im Buch *Databases*¹¹⁷ finden Sie ebenfalls viele Literaturhinweise, dort allerdings in eckigen Klammern, also eher so: [117].
- Wir haben für alles was wir lehren, für jedes Werkzeug, das verwendet wird, für alle Befehle und Datenformate, die wir benutzen, Literaturhinweise herausgesucht.



- Haben Sie die kleinen roten Zahlen in dem Satz „*Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem*^{38,73,79,100}.“ bemerkt?
- Dies sind Literaturhinweise, die am Ende der Slides gelistet werden.
- Im Buch *Databases*¹¹⁷ finden Sie ebenfalls viele Literaturhinweise, dort allerdings in eckigen Klammern, also eher so: [117].
- Wir haben für alles was wir lehren, für jedes Werkzeug, das verwendet wird, für alle Befehle und Datenformate, die wir benutzen, Literaturhinweise herausgesucht.
- Wir beziehen uns auf Standards, wissenschaftliche Veröffentlichungen, Dokumentationen, Vorlesungen an anderen Universitäten und selbst auf aktuelle Gesetzgebung.



- Haben Sie die kleinen roten Zahlen in dem Satz „*Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem*^{38,73,79,100}.“ bemerkt?
- Dies sind Literaturhinweise, die am Ende der Slides gelistet werden.
- Im Buch *Databases*¹¹⁷ finden Sie ebenfalls viele Literaturhinweise, dort allerdings in eckigen Klammern, also eher so: [117].
- Wir haben für alles was wir lehren, für jedes Werkzeug, das verwendet wird, für alle Befehle und Datenformate, die wir benutzen, Literaturhinweise herausgesucht.
- Wir beziehen uns auf Standards, wissenschaftliche Veröffentlichungen, Dokumentationen, Vorlesungen an anderen Universitäten und selbst auf aktuelle Gesetzgebung.
- Wo immer möglich, werden die entsprechenden Quellen im Literaturverzeichnis verlinkt und ich verwende, wo immer möglich, Quellen, die online verfügbar sind.



- Haben Sie die kleinen roten Zahlen in dem Satz „*Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem*^{38,73,79,100}.“ bemerkt?
- Dies sind Literaturhinweise, die am Ende der Slides gelistet werden.
- Im Buch *Databases*¹¹⁷ finden Sie ebenfalls viele Literaturhinweise, dort allerdings in eckigen Klammern, also eher so: [117].
- Wir haben für alles was wir lehren, für jedes Werkzeug, das verwendet wird, für alle Befehle und Datenformate, die wir benutzen, Literaturhinweise herausgesucht.
- Wir beziehen uns auf Standards, wissenschaftliche Veröffentlichungen, Dokumentationen, Vorlesungen an anderen Universitäten und selbst auf aktuelle Gesetzgebung.
- Wo immer möglich, werden die entsprechenden Quellen im Literaturverzeichnis verlinkt und ich verwende, wo immer möglich, Quellen, die online verfügbar sind.
- Am Ende der Slides finden Sie oft ein Glossary (allerdings in Englisch).



- Haben Sie die kleinen roten Zahlen in dem Satz „*Wir verwenden zum Beispiel das PostgreSQL Datenbankmanagementsystem*^{38,73,79,100}.“ bemerkt?
- Dies sind Literaturhinweise, die am Ende der Slides gelistet werden.
- Im Buch *Databases*¹¹⁷ finden Sie ebenfalls viele Literaturhinweise, dort allerdings in eckigen Klammern, also eher so: [117].
- Wir haben für alles was wir lehren, für jedes Werkzeug, das verwendet wird, für alle Befehle und Datenformate, die wir benutzen, Literaturhinweise herausgesucht.
- Wir beziehen uns auf Standards, wissenschaftliche Veröffentlichungen, Dokumentationen, Vorlesungen an anderen Universitäten und selbst auf aktuelle Gesetzgebung.
- Wo immer möglich, werden die entsprechenden Quellen im Literaturverzeichnis verlinkt und ich verwende, wo immer möglich, Quellen, die online verfügbar sind.
- Am Ende der Slides finden Sie oft ein Glossary (allerdings in Englisch).
- Sie können *alles* nachlesen.

Alles ist da.



- Wir wissen, dass es nicht einfach ist, neue Dinge in einer fremden Sprache zu lernen.

Alles ist da.



- Wir wissen, dass es nicht einfach ist, neue Dinge in einer fremden Sprache zu lernen.
- Daher versuchen wir, alle Informationen so klar wie möglich darzustellen.

Alles ist da.



- Wir wissen, dass es nicht einfach ist, neue Dinge in einer fremden Sprache zu lernen.
- Daher versuchen wir, alle Informationen so klar wie möglich darzustellen.
- Wir stellen umfangreiches Lehrmaterial zur Verfügung.

Alles ist da.



- Wir wissen, dass es nicht einfach ist, neue Dinge in einer fremden Sprache zu lernen.
- Daher versuchen wir, alle Informationen so klar wie möglich darzustellen.
- Wir stellen umfangreiches Lehrmaterial zur Verfügung.
- Und wir machen den Kurs so praktisch wie möglich.



Ökosystem



Ökosystem

- Wir versuchen, ein Ökosystem von Lehrmaterial aufzubauen.



Ökosystem

- Wir versuchen, ein Ökosystem von Lehrmaterial aufzubauen.
- Aktuell gibt es folgende beiden Kurse



Ökosystem



- Wir versuchen, ein Ökosystem von Lehrmaterial aufzubauen.
- Aktuell gibt es folgende beiden Kurse:
 - Python, mit dem Textbuch *Thomas Weise (汤卫思). Programming with Python. 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>*

Ökosystem



- Wir versuchen, ein Ökosystem von Lehrmaterial aufzubauen.
- Aktuell gibt es folgende beiden Kurse:
 - Python, mit dem Textbuch *Thomas Weise (汤卫思). Programming with Python. 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> und*
 - Datenbanken, mit dem Textbuch *Thomas Weise (汤卫思). Databases. 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>.*

Ökosystem



- Wir versuchen, ein Ökosystem von Lehrmaterial aufzubauen.
- Aktuell gibt es folgende beiden Kurse:
 - Python, mit dem Textbuch *Thomas Weise (汤卫思). Programming with Python. 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> und*
 - Datenbanken, mit dem Textbuch *Thomas Weise (汤卫思). Databases. 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>.*
- Diese Kurse sind miteinander verzahnt und vom Material her gleich strukturiert.

Ökosystem



- Wir versuchen, ein Ökosystem von Lehrmaterial aufzubauen.
- Aktuell gibt es folgende beiden Kurse:
 - Python, mit dem Textbuch *Thomas Weise (汤卫思). Programming with Python. 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> und*
 - Datenbanken, mit dem Textbuch *Thomas Weise (汤卫思). Databases. 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>.*
- Diese Kurse sind miteinander verzahnt und vom Material her gleich strukturiert.
- Sie sind eingeladen, sich die Materialien beider Kurse anzuschauen.

Ökosystem



- Wir versuchen, ein Ökosystem von Lehrmaterial aufzubauen.
- Aktuell gibt es folgende beiden Kurse:
 - Python, mit dem Textbuch *Thomas Weise (汤卫思). Programming with Python. 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> und*
 - Datenbanken, mit dem Textbuch *Thomas Weise (汤卫思). Databases. 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>.*
- Diese Kurse sind miteinander verzahnt und vom Material her gleich strukturiert.
- Sie sind eingeladen, sich die Materialien beider Kurse anzuschauen.
- Beide werden aktiv weiterentwickelt.



Voraussetzungen



Voraussetzungen



- Bringen Sie Ihren Laptop mit.

Voraussetzungen



- Bringen Sie Ihren Laptop mit.
 1. Ihren eigenen Laptop.

Voraussetzungen



- Bringen Sie Ihren Laptop mit.
 1. Ihren eigenen Laptop.
 2. Immer.

Voraussetzungen



- Bringen Sie Ihren Laptop mit.
 1. Ihren eigenen Laptop.
 2. Immer.
 3. Vollziehen Sie unsere Beispiele nach.

Voraussetzungen



- Bringen Sie Ihren Laptop mit.
 1. Ihren eigenen Laptop.
 2. Immer.
 3. Vollziehen Sie unsere Beispiele nach.
 4. Am Besten während der Vorlesung.

Voraussetzungen



- Bringen Sie Ihren Laptop mit.
 1. Ihren eigenen Laptop.
 2. Immer.
 3. Vollziehen Sie unsere Beispiele nach.
 4. Am Besten während der Vorlesung.
 5. Auf Ihrem eigenen Laptop.

Voraussetzungen



- Bringen Sie Ihren Laptop mit.
 1. Ihren eigenen Laptop.
 2. Immer.
 3. Vollziehen Sie unsere Beispiele nach.
 4. Am Besten während der Vorlesung.
 5. Auf Ihrem eigenen Laptop.
- Das sind schon alle Voraussetzungen.

Voraussetzungen



- Bringen Sie Ihren Laptop mit.
 1. Ihren eigenen Laptop.
 2. Immer.
 3. Vollziehen Sie unsere Beispiele nach.
 4. Am Besten während der Vorlesung.
 5. Auf Ihrem eigenen Laptop.
- Das sind schon alle Voraussetzungen.
- Bonus: Im Idealfall arbeiten Sie mit dem Betriebssystem Ubuntu Linux^{23,46}.

Voraussetzungen



- Bringen Sie Ihren Laptop mit.
 1. Ihren eigenen Laptop.
 2. Immer.
 3. Vollziehen Sie unsere Beispiele nach.
 4. Am Besten während der Vorlesung.
 5. Auf Ihrem eigenen Laptop.
- Das sind schon alle Voraussetzungen.
- Bonus: Im Idealfall arbeiten Sie mit dem Betriebssystem Ubuntu Linux^{23,46}. Sie könnten es ja in einer virtuellen Maschine installieren.



Zusammenfassung



Kurs Datenbanken



- Willkommen zum Kurs **Datenbanken**.

Kurs Datenbanken



- Willkommen zum Kurs **Datenbanken**.
- Ich freue mich auf unsere Zusammenarbeit und hoffe, dass wir Spaß mit diesem Gebiet haben werden.



谢谢你们！
Thank you!
Vielen Dank!



References I



- [1] David J. Agans. *Debugging*. New York, NY, USA: AMACOM, Sep. 2002. ISBN: **978-0-8144-2678-4** (siehe S. **151**).
- [2] Raphael „rkhaotix“ Araújo e Silva. *pgModeler – PostgreSQL Database Modeler*. Palmas, Tocantins, Brazil, 2006–2025. URL: <https://pgmodeler.io> (besucht am 2025-04-12) (siehe S. **82–89**, **153**).
- [3] Richard Barker. *Case*Method: Entity Relationship Modelling (Oracle)*. 1. Aufl. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., Jan. 1990. ISBN: **978-0-201-41696-1** (siehe S. **151**).
- [4] 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. **96–100**, **152**, **154**).
- [5] Ben Beitler. *Hands-On Microsoft Access 2019*. Birmingham, England, UK: Packt Publishing Ltd, März 2020. ISBN: **978-1-83898-747-3** (siehe S. **152**).
- [6] 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. **151**, **155**).
- [7] 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. **150**).
- [8] Bernard Obeng Boateng. *Data Modeling with Microsoft Excel*. Birmingham, England, UK: Packt Publishing Ltd, Nov. 2023. ISBN: **978-1-80324-028-2** (siehe S. **152**).
- [9] Ethan Bommarito und Michael Bommarito. *An Empirical Analysis of the Python Package Index (PyPI)*. arXiv.org: Computing Research Repository (CoRR) abs/1907.11073. Ithaca, NY, USA: Cornell University Library, 26. Juli 2019. doi:[10.48550/arXiv.1907.11073](https://doi.org/10.48550/arXiv.1907.11073). URL: <https://arxiv.org/abs/1907.11073> (besucht am 2024-08-17). arXiv:1907.11073v2 [cs.SE] 26 Jul 2019 (siehe S. **153**).
- [10] Grady Booch, James Rumbaugh und Ivar Jacobson. *The Unified Modeling Language Reference Manual*. 1. Aufl. Reading, MA, USA: Addison-Wesley Professional, Jan. 1999. ISBN: **978-0-201-57168-4** (siehe S. **155**).
- [11] Ed Bott. *Windows 11 Inside Out*. Hoboken, NJ, USA: Microsoft Press, Pearson Education, Inc., Feb. 2023. ISBN: **978-0-13-769132-6** (siehe S. **152**).

References II



- [12] Ron Brash und Ganesh Naik. *Bash Cookbook*. Birmingham, England, UK: Packt Publishing Ltd, Juli 2018. ISBN: 978-1-78862-936-2 (siehe S. 150).
- [13] 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. 151).
- [14] Ben Brumm. "A Guide to the Entity Relationship Diagram (ERD)". In: *Database Star*. Armadale, VIC, Australia: Elevated Online Services PTY Ltd., 30. Juli 2019–23. Dez. 2023. URL: <https://www.databasestar.com/entity-relationship-diagram> (besucht am 2025-03-29) (siehe S. 151).
- [15] Jason Cannon. *High Availability for the LAMP Stack*. Shelter Island, NY, USA: Manning Publications, Juni 2022 (siehe S. 154).
- [16] Josh Centers. *Take Control of iOS 18 and iPadOS 18*. San Diego, CA, USA: Take Control Books, Dez. 2024. ISBN: 978-1-990783-55-5 (siehe S. 151).
- [17] 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. 82–89, 154).
- [18] Bo Chen, Chris N. Potts und Gerhard J. Woeginger. "A Review of Machine Scheduling: Complexity, Algorithms and Approximability". In: *Handbook of Combinatorial Optimization*. Hrsg. von Panos Miliadis Pardalos, Ding-Zhu Du und Ronald Lewis Graham. 1. Aufl. Boston, MA, USA: Springer, 1998, S. 1493–1641. ISBN: 978-1-4613-7987-4. doi:10.1007/978-1-4613-0303-9_25. See also pages 21–169 in volume 3/3 by Norwell, MA, USA: Kluwer Academic Publishers. (Siehe S. 156).
- [19] Peter Pin-Shan Chen. "Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned". In: *Software Pioneers: Contributions to Software Engineering*. Hrsg. von Manfred Broy und Ernst Denert. Berlin/Heidelberg, Germany: Springer-Verlag GmbH Germany, Feb. 2002, S. 296–310. doi:10.1007/978-3-642-59412-0_17. URL: http://bit.csc.lsu.edu/%7Echen/pdf/Chen_Pioneers.pdf (besucht am 2025-03-06) (siehe S. 151).

References III



- [20] Peter Pin-Shan Chen. "The Entity-Relationship Model – Toward a Unified View of Data". *ACM Transactions on Database Systems (TODS)* 1(1):9–36, März 1976. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: **0362-5915**. doi:[10.1145/320434.320440](https://doi.org/10.1145/320434.320440) (siehe S. **137**, **151**).
- [21] Peter Pin-Shan Chen. "The Entity-Relationship Model: Toward a Unified View of Data". In: *1st International Conference on Very Large Data Bases (VLDB'1975)*. 22.–24. Sep. 1975, Framingham, MA, USA. Hrsg. von Douglas S. Kerr. New York, NY, USA: Association for Computing Machinery (ACM), 1975, S. 173. ISBN: **978-1-4503-3920-9**. doi:[10.1145/1282480.1282492](https://doi.org/10.1145/1282480.1282492). See²⁰ for a more comprehensive introduction. (Siehe S. **151**).
- [22] 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. **152**).
- [23] 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. **96–100**, **122–130**, **154**, **155**).
- [24] Edgar Frank „Ted“ Codd. *Normalized Data Base Structure: A Brief Tutorial*. IBM Research Report RJ935. San Jose, CA, USA: IBM Research Laboratory, 1971. URL: [https://www.fsmwarden.com/Codd/Normalized%20data%20base%20structure_%20a%20brief%20tutorial\(1971,%20nov\).pdf](https://www.fsmwarden.com/Codd/Normalized%20data%20base%20structure_%20a%20brief%20tutorial(1971,%20nov).pdf) (besucht am 2025-05-04) (siehe S. **137**).
- [25] Edgar Frank „Ted“ Codd. "Normalized Data Base Structure: A Brief Tutorial". In: *ACM SIGFIDET Workshop on Data Description, Access, and Control*. 11.–12. Nov. 1971, San Diego, CA, USA. Hrsg. von Edgar Frank „Ted“ Codd und Albert L. Dean Jr. New York, NY, USA: Association for Computing Machinery (ACM), 1971, S. 1–17. ISBN: **978-1-4503-7300-5**. doi:[10.1145/1734714.1734716](https://doi.org/10.1145/1734714.1734716). See also²⁴ (siehe S. **150**).
- [26] 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](https://doi.org/10.1145/362384.362685). URL: <https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf> (besucht am 2025-01-05) (siehe S. **150**, **153**).

References IV



- [27] Edgar Frank „Ted“ Codd. *Further Normalization of the Data Base Relational Model*. IBM Research Report RJ909. San Jose, CA, USA: IBM Research Laboratory, 31. Aug. 1971. URL: <https://forum.thethirdmanifesto.com/wp-content/uploads/asgarosforum/987737/00-efc-further-normalization.pdf> (besucht am 2025-05-04). Reprinted in and presented at⁸⁴ (siehe S. 150).
- [28] Stephen Arthur Cook. "The Complexity of Theorem-Proving Procedures". In: *Third Annual ACM Symposium on Theory of Computing (STOC'1971)*. 3.–5. Mai 1971, Shaker Heights, OH, USA. Hrsg. von Michael A. Harrison, Ranan B. Banerji und Jeffrey D. Ullman. New York, NY, USA: Association for Computing Machinery (ACM), 1971, S. 151–158. ISBN: 978-1-4503-7464-4. doi:10.1145/800157.805047 (siehe S. 156).
- [29] *Database Language SQL*. Techn. Ber. ANSI X3.135-1986. Washington, D.C., USA: American National Standards Institute (ANSI), 1986 (siehe S. 154).
- [30] Christopher J. Date. *An Introduction to Database Systems*. 8. Aufl. Hoboken, NJ, USA: Pearson Education, Inc., Juli 2003. ISBN: 978-0-321-19784-9 (siehe S. 150, 153).
- [31] 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. 154).
- [32] *Database Language SQL*. International Standard ISO 9075-1987. Geneva, Switzerland: International Organization for Standardization (ISO), 1987 (siehe S. 154).
- [33] Paul Deitel, Harvey Deitel und Abbey Deitel. *Internet & World Wide Web: How to Program*. 5. Aufl. Hoboken, NJ, USA: Pearson Education, Inc., Nov. 2011. ISBN: 978-0-13-299045-5 (siehe S. 155).
- [34] 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. 152).
- [35] *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. 151).

References V



- [36] Ramez Elmasri und Shamkant Navathe. *Fundamentals of Database Systems*. 7. Aufl. Hoboken, NJ, USA: Pearson Education, Inc., Juni 2015. ISBN: **978-0-13-397077-7** (siehe S. **150, 153**).
- [37] Steve Fanning, Vasudev Narayanan, „flywire“, Olivier Hallot, Jean Hollis Weber, Jenna Sargent, Pulkit Krishna, Dan Lewis, Peter Schofield, Jochen Schiffrers, Robert Großkopf, Jost Lange, Martin Fox, Hazel Russman, Steve Schwettman, Alain Romedenne, 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. **82–89, 152**).
- [38] 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. **82–89, 101–108, 153**).
- [39] 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 (IFIPACT). 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. **151, 152**).
- [40] Michael T. Goodrich. *A Gentle Introduction to NP-Completeness*. Irvine, CA, USA: University of California, Irvine, Apr. 2022. URL: <https://ics.uci.edu/~goodrich/teach/cs165/notes/NPComplete.pdf> (besucht am 2025-08-01) (siehe S. **156**).
- [41] Michael Goodwin. *What is an API?* Armonk, NY, USA: International Business Machines Corporation (IBM), 9. Apr. 2024. URL: <https://www.ibm.com/topics/api> (besucht am 2024-12-12) (siehe S. **150**).
- [42] 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. **152**).
- [43] 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. **153**).
- [44] 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. **153**).

References VI



- [45] Michael Hausenblas. *Learning Modern Linux*. Sebastopol, CA, USA: O'Reilly Media, Inc., Apr. 2022. ISBN: **978-1-0981-0894-6** (siehe S. **96–100, 152**).
- [46] 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. **96–100, 122–130, 155**).
- [47] Ian Hickson, Robin Berjon, Steve Faulkner, Travis Leithead, Erika Doyle Navara, Theresa O'Connor und Silvia Pfeiffer, Hrsg. *HTML5: A Vocabulary and Associated APIs for HTML and XHTML*. W3C Recommendation. Wakefield, MA, USA: World Wide Web Consortium (W3C), 28. Okt. 2014. URL: <http://www.w3.org/TR/2014/REC-html5-20141028> (besucht am 2024-12-17) (siehe S. **151**).
- [48] 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](https://doi.org/10.1007/978-3-031-35122-8) (siehe S. **153**).
- [49] John D. Hunter. "Matplotlib: A 2D Graphics Environment". *Computing in Science & Engineering* 9(3):90–95, Mai–Juni 2007. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE). ISSN: **1521-9615**. doi:[10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55) (siehe S. **152**).
- [50] John D. Hunter, Darren Dale, Eric Firing, Michael Droettboom und The Matplotlib Development Team. *Matplotlib: Visualization with Python*. Austin, TX, USA: NumFOCUS, Inc., 2012–2025. URL: <https://matplotlib.org> (besucht am 2025-02-02) (siehe S. **152**).
- [51] *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. **154**).
- [52] *Information Technology – Universal Coded Character Set (UCS)*. International Standard ISO/IEC 10646:2020. Geneva, Switzerland: International Organization for Standardization (ISO) und International Electrotechnical Commission (IEC), Dez. 2020 (siehe S. **155**).
- [53] *Python 3 Documentation. Installing Python Modules*. Beaverton, OR, USA: Python Software Foundation (PSF), 2001–2025. URL: <https://docs.python.org/3/installing> (besucht am 2024-08-17) (siehe S. **153**).

References VII



- [54] Robert Johansson. *Numerical Python: Scientific Computing and Data Science Applications with NumPy, SciPy and Matplotlib*. New York, NY, USA: Apress Media, LLC, Dez. 2018. ISBN: 978-1-4842-4246-9 (siehe S. 152).
- [55] „exit – Terminate a Process“. In: *POSIX.1-2024: The Open Group Base Specifications Issue 8, IEEE Std 1003.1™-2024 Edition*. Hrsg. von Andrew Josey. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE) und San Francisco, CA, USA: The Open Group, 8. Aug. 2024. URL: <https://pubs.opengroup.org/onlinepubs/979919799/functions/exit.html> (besucht am 2024-10-30) (siehe S. 94, 95).
- [56] Andrew Josey, Hrsg. *POSIX.1-2024: The Open Group Base Specifications Issue 8, IEEE Std 1003.1™-2024 Edition*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE) und San Francisco, CA, USA: The Open Group, 8. Aug. 2024. URL: <https://pubs.opengroup.org/onlinepubs/979919799> (besucht am 2024-10-30).
- [57] „stderr, stdin, stdout – Standard I/O Streams“. In: *POSIX.1-2024: The Open Group Base Specifications Issue 8, IEEE Std 1003.1™-2024 Edition*. Hrsg. von Andrew Josey. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE) und San Francisco, CA, USA: The Open Group, 8. Aug. 2024. URL: <https://pubs.opengroup.org/onlinepubs/979919799/functions/stdin.html> (besucht am 2024-10-30) (siehe S. 94, 95, 154).
- [58] Shannon Kempe und Paul Williams. *A Short History of the ER Diagram and Information Modeling*. Studio City, CA, USA: Dataversity Digital LLC, 25. Sep. 2012. URL: <https://www.dataversity.net/a-short-history-of-the-er-diagram-and-information-modeling> (besucht am 2025-03-06) (siehe S. 151).
- [59] William (Bill) Kent. “A Simple Guide to Five Normal Forms in Relational Database Theory”. *Communications of the ACM (CACM)* 26(2):120–125, Sep. 1982–Feb. 1983. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 0001-0782. doi:10.1145/358024.358054. URL: <https://www.cs.dartmouth.edu/~cs61/Resources/Papers/CACM%20Kent%20Five%20Normal%20Forms.pdf> (besucht am 2025-05-03) (siehe S. 150, 153).
- [60] Jay LaCroix. *Mastering Ubuntu Server*. 4. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Sep. 2022. ISBN: 978-1-80323-424-3 (siehe S. 154).
- [61] 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. 152).

References VIII



- [62] Eugene Leighton Lawler, Jan Karel Lenstra, Alexander Hendrik George Rinnooy Kan und David B. Shmoys. "Sequencing and Scheduling: Algorithms and Complexity". In: *Production Planning and Inventory*. Hrsg. von Stephen C. Graves, Alexander Hendrik George Rinnooy Kan und Paul H. Zipkin. Bd. IV der Reihe Handbooks of Operations Research and Management Science. Amsterdam, The Netherlands: Elsevier B.V., 1993. Kap. 9, S. 445–522. ISSN: 0927-0507. ISBN: 978-0-444-87472-6. doi:10.1016/S0927-0507(05)80189-6. URL: <http://alexandria.tue.nl/repository/books/339776.pdf> (besucht am 2023-12-06) (siehe S. 156).
- [63] 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. 153).
- [64] Marc-André Lemburg. *Python Database API Specification v2.0*. Python Enhancement Proposal (PEP) 249. Beaverton, OR, USA: Python Software Foundation (PSF), 12. Apr. 1999. URL: <https://peps.python.org/pep-0249> (besucht am 2025-02-02) (siehe S. 153).
- [65] LibreOffice – The Document Foundation. Berlin, Germany: The Document Foundation, 2024. URL: <https://www.libreoffice.org> (besucht am 2024-12-12) (siehe S. 151, 152).
- [66] 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. 150).
- [67] Mark Lutz. *Learning Python*. 6. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2025. ISBN: 978-1-0981-7130-8 (siehe S. 153).
- [68] 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. 152).
- [69] 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. 82–89, 154).
- [70] Carl Meyer. *Python Virtual Environments*. Python Enhancement Proposal (PEP) 405. Beaverton, OR, USA: Python Software Foundation (PSF), 13. Juni 2011–24. Mai 2012. URL: <https://peps.python.org/pep-0405> (besucht am 2024-12-25) (siehe S. 155).

References IX



- [71] *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. 152).
- [72] 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. 150).
- [73] 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. 82–89, 101–108, 153).
- [74] *OMG® Unified Modeling Language® (OMG UML®). Version 2.5.1*. OMG Document formal/2017-12-05. Milford, MA, USA: Object Management Group, Inc. (OMG), Dez. 2017. URL: <https://www.omg.org/spec/UML/2.5.1/PDF> (besucht am 2025-03-30) (siehe S. 155).
- [75] 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. 150).
- [76] Ashwin Pajankar. *Hands-on Matplotlib: Learn Plotting and Visualizations with Python 3*. New York, NY, USA: Apress Media, LLC, Nov. 2021. ISBN: 978-1-4842-7410-1 (siehe S. 152).
- [77] 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 Kononov, Robert M. Flight, Kai Blin und Juan Antonio Vizcaino. "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. 151).
- [78] pip Developers. *pip Documentation v24.3.1*. Beaverton, OR, USA: Python Software Foundation (PSF), 27. Okt. 2024. URL: <https://pip.pypa.io> (besucht am 2024-12-25) (siehe S. 153).
- [79] *PostgreSQL Essentials: Leveling Up Your Data Work*. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2024 (siehe S. 82–89, 101–108, 153).
- [80] 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. 150).

References X



- [81] 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. **150**).
- [82] Kristian Rother. *Pro Python Best Practices: Debugging, Testing and Maintenance*. New York, NY, USA: Apress Media, LLC, März 2017. ISBN: **978-1-4842-2241-6** (siehe S. **151**).
- [83] Ernest E. Rothman, Rich Rosen und Brian Jepson. *Mac OS X for Unix Geeks*. 4. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., Sep. 2008. ISBN: **978-0-596-52062-5** (siehe S. **152**).
- [84] Randall Rustin, Hrsg. *Data Base Systems: Courant Computer Science Symposium 6*. 24.–25. Mai 1971, New York, NY, USA. Englewood Cliffs, NJ, USA: Prentice-Hall, 1972. ISBN: **978-0-13-196741-0** (siehe S. **138**).
- [85] Ahmad Sahar. *iOS 26 Programming for Beginners*. 10. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Nov. 2025. ISBN: **978-1-80602-393-6** (siehe S. **155**).
- [86] Matthias Sedlmeier und Martin Gogolla. "Model Driven ActiveRecord with yEd". In: *25th International Conference on Information Modelling and Knowledge Bases XXVII (EJC'2015)*. 8.–12. Juni 2015, Maribor, Štajerska, Podravska, Slovenia. Hrsg. von Tatjana Welzer, Hannu Jaakkola, Bernhard Thalheim, Yasushi Kiyoki und Naofumi Yoshida. Bd. 280 der Reihe Frontiers in Artificial Intelligence and Applications. Amsterdam, The Netherlands: IOS Press BV, 2015, S. 65–76. ISSN: **0922-6389**. ISBN: **978-1-61499-610-1**. doi:**10.3233/978-1-61499-611-8-65** (siehe S. **82–89, 156**).
- [87] 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. **82–89, 151, 152**).
- [88] Yuriy Shamshin. "Conceptual Database Model. Entity Relationship Diagram (ERD)". In: *Databases*. Riga, Latvia: ISMA University of Applied Sciences, Mai 2024. Kap. 04. URL: https://dbs.academy.lv/lection/dbs_LS04EN_erd.pdf (besucht am 2025-03-29) (siehe S. **151**).
- [89] Yuriy Shamshin. *Databases*. Riga, Latvia: ISMA University of Applied Sciences, Mai 2024. URL: <https://dbs.academy.lv> (besucht am 2025-01-11).

References XI



- [90] Yuriy Shamshin. "Normalization". In: *Databases*. Riga, Latvia: ISMA University of Applied Sciences, Mai 2024. Kap. 07a. URL: https://dbs.academy.lv/lection/dbs_LS07ENa_normalization.pdf (besucht am 2025-05-03) (siehe S. 153).
- [91] Yuriy Shamshin. "RDM Normalization. Data Anomalies. Functional Dependency. Normal Forms.". In: *Databases*. Riga, Latvia: ISMA University of Applied Sciences, Mai 2024. Kap. 07. URL: https://dbs.academy.lv/lection/dbs_LS07EN_normalization.pdf (besucht am 2025-05-03) (siehe S. 153).
- [92] 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. 96–100, 152).
- [93] Anna Skoulikari. *Learning Git*. Sebastopol, CA, USA: O'Reilly Media, Inc., Mai 2023. ISBN: 978-1-0981-3391-7 (siehe S. 151).
- [94] Drew Smith. *Modern Apple Platform Administration – macOS and iOS Essentials (2025)*. Birmingham, England, UK: Packt Publishing Ltd, Feb. 2025. ISBN: 978-1-80580-309-6 (siehe S. 151, 152).
- [95] 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. 153).
- [96] "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. 154).
- [97] 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. 145, 154).
- [98] 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. Burghthann, Bayern, Germany: Markt+Technik Verlag GmbH, Feb. 1998. ISBN: 978-3-8272-2020-2. Translation of ⁹⁷ (siehe S. 154).
- [99] Allen Taylor. *Introducing SQL and Relational Databases*. New York, NY, USA: Apress Media, LLC, Sep. 2018. ISBN: 978-1-4842-3841-7 (siehe S. 153, 154).

References XII



- [100] 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. **82–89, 101–108, 153**).
- [101] *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. **151**).
- [102] *The Python Package Index (PyPI)*. Beaverton, OR, USA: Python Software Foundation (PSF), 2024. URL: <https://pypi.org> (besucht am 2024-08-17) (siehe S. **153**).
- [103] *The Unicode Standard, Version 15.1: Archived Code Charts*. South San Francisco, CA, USA: The Unicode Consortium, 25. Aug. 2023. URL: <https://www.unicode.org/Public/15.1.0/charts/CodeCharts.pdf> (besucht am 2024-07-26) (siehe S. **155**).
- [104] 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** (siehe S. **96–100, 152**).
- [105] Brad Traversy. *Modern HTML & CSS From The Beginning 2.0*. 2. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Juli 2024. ISBN: **978-1-83588-056-2** (siehe S. **151**).
- [106] 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. **151, 155**).
- [107] 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. **152**).
- [108] *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 3. Aufl. The Addison-Wesley Object Technology Series. Reading, MA, USA: Addison-Wesley Professional, Sep. 2003. ISBN: **978-0-321-19368-1** (siehe S. **155**).

References XIII



- [109] *UML Notation Guide*. Version 1.1. Santa Clara, CA, USA: Rational Software Corporation, Redmond, WA, USA: Microsoft Corporation, Palo Alto, CA, USA: Hewlett-Packard Company, Redwood Shores, CA, USA: Oracle Corporation, Dallas, TX, USA: Sterling Software, Ottawa, ON, Canada: MCI Systemhouse Corporation, Blue Bell, PA, USA: Unisys Corporation, Blue Bell, PA, USA: ICON Computing, Santa Clara, CA, USA: IntelliCorp, Burlington, MA, USA: i-Logix, Armonk, NY, USA: International Business Machines Corporation (IBM), Kanata, ON, Canada: ObjectTime Limited, Chicago, IL, USA: Platinum Technology Inc., Boston, MA, USA: Ptech Inc., Orlando, FL, USA: Taskon A/S, Paoli, PA, USA: Reich Technologies und Paris, Île-de-France, France: Softeam, 1. Sep. 1997. URL: <https://web.cse.msu.edu/~cse870/Materials/uml-notation-guide-9-97.pdf> (besucht am 2025-03-30) (siehe S. 155).
- [110] *Unicode® 15.1.0*. South San Francisco, CA, USA: The Unicode Consortium, 12. Sep. 2023. ISBN: 978-1-936213-33-7. URL: <https://www.unicode.org/versions/Unicode15.1.0> (besucht am 2024-07-26) (siehe S. 155).
- [111] Marat Valiev, Bogdan Vasilescu und James D. Herbsleb. "Ecosystem-Level Determinants of Sustained Activity in Open-Source Projects: A Case Study of the PyPI Ecosystem". In: *ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/SIGSOFT FSE'2018)*. 4.–9. Nov. 2018, Lake Buena Vista, FL, USA. Hrsg. von Gary T. Leavens, Alessandro F. Garcia und Corina S. Păsăreanu. New York, NY, USA: Association for Computing Machinery (ACM), 2018, S. 644–655. ISBN: 978-1-4503-5573-5. doi:10.1145/3236024.3236062 (siehe S. 153).
- [112] Bruce M. Van Horn II und Quan Nguyen. *Hands-On Application Development with PyCharm*. 2. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Okt. 2023. ISBN: 978-1-83763-235-0 (siehe S. 153).
- [113] Sander van Vugt. *Linux Fundamentals*. 2. Aufl. Hoboken, NJ, USA: Pearson IT Certification, Juni 2022. ISBN: 978-0-13-792931-3 (siehe S. 96–100, 152).
- [114] Daniele „dvarrazzo“ Varrazzo, Federico „fogzot“ Di Gregorio und Jason „jerickso“ Erickson. *Psycopg*. London, England, UK: The Psycopg Team, 2010–2023. URL: <https://www.psycopg.org> (besucht am 2025-02-02) (siehe S. 82–89, 153).
- [115] "venv – Creation of Virtual Environments". In: *Python 3 Documentation. The Python Standard Library*. Beaverton, OR, USA: Python Software Foundation (PSF), 2001–2025. URL: <https://docs.python.org/3/library/venv.html> (besucht am 2024-12-25) (siehe S. 155).
- [116] "Virtual Environments and Packages". In: *Python 3 Documentation. The Python Tutorial*. Beaverton, OR, USA: Python Software Foundation (PSF), 2001–2025. Kap. 12. URL: <https://docs.python.org/3/tutorial/venv.html> (besucht am 2024-12-24) (siehe S. 155).

References XIV



- [117] 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. 67–72, 94, 95, 101–108, 114–120, 150, 151, 153).
- [118] 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. 82–89, 114–120, 151, 153).
- [119] Matthew West. *Developing High Quality Data Models*. Version: 2.0, Issue: 2.1. London, England, UK: Shell International Limited und European Process Industries STEP Technical Liaison Executive (EPISTLE); Burlington, MA, USA/San Mateo, CA, USA: Morgan Kaufmann Publishers, 8. Dez. 1995–Dez. 2010. ISBN: 978-0-12-375107-2. URL: <https://www.researchgate.net/publication/286610894> (besucht am 2025-03-24). Edited by Julian Fowler (siehe S. 151).
- [120] *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. 153).
- [121] Kevin Wilson. *Python Made Easy*. Birmingham, England, UK: Packt Publishing Ltd, Aug. 2024. ISBN: 978-1-83664-615-0 (siehe S. 151, 153).
- [122] Martin Yanev. *PyCharm Productivity and Debugging Techniques*. Birmingham, England, UK: Packt Publishing Ltd, Okt. 2022. ISBN: 978-1-83763-244-2 (siehe S. 153).
- [123] 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. 150).
- [124] *yEd Graph Editor Manual*. Tübingen, Baden-Württemberg, Germany: yWorks GmbH, 2011–2025. URL: <https://yed.yworks.com/support/manual/index.html> (besucht am 2025-03-31) (siehe S. 82–89, 156).

References XV



- [125] 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](https://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. 152).
- [126] Giorgio Zarrelli. *Mastering Bash*. Birmingham, England, UK: Packt Publishing Ltd, Juni 2017. ISBN: 978-1-78439-687-9 (siehe S. 150).

Glossary (in English) I



1NF The first normal form (NF) in relational databases (DBs)^{26,30,36,59}.

2NF The second normal form (NF) in relational DBs^{25,27,30,36,59}.

3NF The third normal form (NF) in relational DBs^{25,27,30,36,59}.

API An *Application Programming Interface* is a set of rules or protocols that enables one software application or component to use or communicate with another⁴¹.

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^{12,72,126}. Learn more at <https://www.gnu.org/software/bash>.

breakpoint A breakpoint is a mark in a line of code in an Integrated Development Environment (IDE) at which the debugger will pause the execution of a program.

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^{7,66,75,80,81}. 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 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.

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*¹¹⁷.

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 DB¹²³.

Glossary (in English) II



- debugger** A debugger is a tool that lets you execute a program step-by-step while observing the current values of variables. This allows you to find errors in the code more easily^{1,82,121}. Learn more about debugging in¹¹⁸.
- ERD** Entity relationship diagrams show the relationships between objects, e.g., between the tables in a DB and how they reference each other^{3,14,19–21,58,88,119}.
- 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^{93,106}. Learn more at <https://git-scm.com>.
- GitHub** is a website where software projects can be hosted and managed via the Git VCS^{77,106}. Learn more at <https://github.com>.
- HTML** The Hyper Text Markup Language (HTML) is the text format used by the WWW^{6,47,105}.
- IDE** An *Integrated Developer Environment* is a program that allows the user do multiple different activities required for software development in one single system. It often offers functionality such as editing source code, debugging, testing, or interaction with a distributed version control system. For Python, we recommend using PyCharm. On Apple systems, Xcode is often used.
- iOS** is the operating system that powers Apple iPhones^{16,94}. Learn more at <https://www.apple.com/ios>.
- JavaScript** JavaScript is the predominant programming language used in websites to develop interactive contents for display in browsers³⁵.
- JSON** *JavaScript Object Notation* is a data interchange format^{13,101} based on JavaScript³⁵ syntax.
- LibreOffice** is on open source office suite^{39,65,87} which is a good and free alternative to Microsoft Office. It offers software such as LibreOffice Writer, LibreOffice Calc, and LibreOffice Base. See¹¹⁷ for more information and installation instructions.

Glossary (in English) III



- LibreOffice Base is a DBMS that can work on stand-alone files but also connect to other popular relational databases^{37,87}. It is part of LibreOffice^{39,65,87} and has functionality that is comparable to Microsoft Access^{5,22,107}.
- 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 spread sheet software^{65,87}, i.e., an alternative to Microsoft Excel. It is part of LibreOffice^{39,65,87}.
- LibreOffice Writer is a free and open source text writing program¹²⁵ and part of LibreOffice^{39,65,87}. It is a good alternative to Microsoft Word.
- Linux is the leading open source operating system, i.e., a free alternative for Microsoft Windows^{4,45,92,104,113}. 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.
- macOS or Mac OS is the operating system that powers Apple Mac(intosh) computers^{83,94}. Learn more at <https://www.apple.com/macos>.
- Matplotlib is a Python package for plotting diagrams and charts^{49,50,54,76}. Learn more at <https://matplotlib.org>⁵⁰.
- Microsoft Access is a DBMS that can work on DBs stored in single, stand-alone files but also connect to other popular relational databases^{5,22,68,107}. It is part of Microsoft Office. A free and open source alternative to this commercial software is LibreOffice Base.
- Microsoft Excel is a spreadsheet program that allows users to store, organize, manipulate, and calculate data in tabular structures^{8,42,61}. It is part of Microsoft Office. A free alternative to this commercial software is LibreOffice Calc^{65,87}.
- Microsoft Office is a commercial suite of office software, including Microsoft Excel, Microsoft Word, and Microsoft Access⁶¹. LibreOffice is a free and open source alternative.
- Microsoft Windows is a commercial proprietary operating system¹¹. 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^{34,71,125} and part of Microsoft Office. A free alternative to this commercial software is the LibreOffice Writer.

Glossary (in English) IV



NF The *normal forms* define guidelines for the design of relational DBs with the goal to avoid redundancy and to prevent inconsistencies and anomalies^{30,36,59,90,91}. There are several normal forms, 1NF, second normal form (2NF), third normal form (3NF), and so on, each more restrictive than the other.

PgModeler the PostgreSQL DB modeler is a tool that allows for graphical modeling of logical schemas for DBs using an entity relationship diagram (ERD)-like notation². Learn more at <https://pgmodeler.io>.

pip is the standard tool to install Python software packages from the PyPI repository^{53,78}. To install a package `thepackage` hosted on PyPI, type `pip install thepackage` into the terminal. Learn more at <https://packaging.python.org/installing>.

PostgreSQL An open source object-relational DBMS^{38,73,79,100}. See <https://postgresql.org> for more information.

psql is the client program used to access the PostgreSQL DBMS server.

psycopg or, more exactly, `psycopg 3`, is the most popular PostgreSQL adapter for Python, implementing the Python DB API 2.0 specification⁶⁴. Learn more at <https://www.psycopg.org>¹¹⁴.

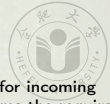
PyCharm is the convenient Python IDE that we recommend for this course^{112,121,122}. It comes in a free community edition, so it can be downloaded and used at no cost. Learn more at <https://www.jetbrains.com/pycharm>.

PyPI The Python Package Index (PyPI) is an online repository that provides the software packages that you can install with pip^{9,102,111}. Learn more at <https://pypi.org>.

Python The Python programming language^{48,63,67,118}, i.e., what you will learn about in our book¹¹⁸. 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^{26,43,44,95,99,117,120}.

Glossary (in English) V



- 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 servers¹⁵ 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“⁶⁰.
- SQL** The *Structured Query Language* is basically a programming language for querying and manipulating relational databases^{17,29,31,32,51,69,96–99}. It is understood by many DBMSes. You find the SQL commands supported by PostgreSQL in the reference⁹⁶.
- stderr** The *standard error stream* is one of the three pre-defined streams of a console process (together with the `stdin` and the `stdout`)⁵⁷. It is the text stream to which the process writes information about errors and exceptions. If an uncaught `Exception` is raised in Python and the program terminates, then this information is written to standard error stream (`stderr`). If you run a program in a terminal, then the text that a process writes to its `stderr` appears in the console.
- stdin** The *standard input stream* is one of the three pre-defined streams of a console process (together with the `stdout` and the `stderr`)⁵⁷. It is the text stream from which the process reads its input text, if any. The Python instruction `input` reads from this stream. If you run a program in a terminal, then the text that you type into the terminal while the process is running appears in this stream.
- stdout** The *standard output stream* is one of the three pre-defined streams of a console process (together with the `stdin` and the `stderr`)⁵⁷. It is the text stream to which the process writes its normal output. The `print` instruction of Python writes text to this stream. If you run a program in a terminal, then the text that a process writes to its `stdout` appears in the console.
- terminal** A terminal is a text-based window where you can enter commands and execute them^{4,23}. 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 `Win` + `R`, dann Schreiben von `cmd`, dann Druck auf `↵`. Under Ubuntu Linux, `Ctrl` + `Alt` + `T` opens a terminal, which then runs a Bash shell inside.

Glossary (in English) VI



- Ubuntu is a variant of the open source operating system Linux^{23,46}. 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>.
- UCS Universal Coded Character Set, see Unicode
- UML The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems^{10,74,108,109}
- Unicode A standard for assigning characters to numbers^{52,103,110}. The Unicode standard supports basically all characters from all languages that are currently in use, as well as many special symbols. It is the predominantly used way to represent characters in computers and is regularly updated and improved.
- VCS A *Version Control System* is a software which allows you to manage and preserve the historical development of your program code¹⁰⁶. 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.
- venv is a Python module and tool for creating virtual environments¹¹⁵. Learn more at <https://docs.python.org/3/library/venv.html>¹¹⁵.
- virtual environment A virtual environment is a directory that contains a local Python installation^{70,116}. It comes with its own package installation directory. Multiple different virtual environments can be installed on a system. This allows different applications to use different versions of the same packages without conflict, because we can simply install these applications into different virtual environments.
- WWW World Wide Web^{6,33}
- Xcode is offers the tools for developing, testing, and distributing applications as well as an IDE for Apple platforms such as macOS and iOS⁸⁵.

Glossary (in English) VII



yEd is a graph editor for high-quality graph-based diagrams^{86,124}, suitable to draw, e.g., technology-independent ERDs, control flow charts, or UML class diagrams. An online version of the editor is available at <https://www.yworks.com/yed-live>. Learn more at <https://www.yworks.com/products/yed>.

$\mathcal{NP}$ is the class of computational problems that can be solved in polynomial time by a non-deterministic machine and can be verified in polynomial time by a deterministic machine (such as a normal computer)⁴⁰.

$\mathcal{NP}$ -hard Algorithms that guarantee to find the correct solutions of $\mathcal{NP}$ -hard problems^{18,28,62} need a runtime that is exponential in the problem scale in the worst case. A problem is $\mathcal{NP}$ -hard if all problems in $\mathcal{NP}$ are reducible to it in polynomial time⁴⁰.

$\mathbb{R}$ the set of the real numbers.