



合肥大學
HEFEI UNIVERSITY



Datenbanken

2. Einleitung

Thomas Weise (汤卫思)
tweise@hfuu.edu.cn

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

应用优化研究所
人工智能与大数据学院
合肥大学
中国安徽省合肥市

Databases



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

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



Outline



1. Einleitung
2. Daten
3. Datenbanken
4. Zusammenfassung





Einleitung



Einleitung

- Dieser Kurs lehrt die Verwendung von Datenbanken.



Einleitung

- Dieser Kurs lehrt die Verwendung von Datenbanken.
- Unser Fokus liegt auf einem praxisorientierten Ansatz.



Einleitung



- Dieser Kurs lehrt die Verwendung von Datenbanken.
- Unser Fokus liegt auf einem praxisorientierten Ansatz.
- Das bedeutet, dass alle Konzepte, die wir diskutieren, immer von einer reichen Auswahl von praktischen Beispielen begleitet werden.

Einleitung



- Dieser Kurs lehrt die Verwendung von Datenbanken.
- Unser Fokus liegt auf einem praxisorientierten Ansatz.
- Das bedeutet, dass alle Konzepte, die wir diskutieren, immer von einer reichen Auswahl von praktischen Beispielen begleitet werden.
- In diesem Kurs werden wir daher auch viele Werkzeuge und echte Datenbanken verwenden.

Einleitung



- Dieser Kurs lehrt die Verwendung von Datenbanken.
- Unser Fokus liegt auf einem praxisorientierten Ansatz.
- Das bedeutet, dass alle Konzepte, die wir diskutieren, immer von einer reichen Auswahl von praktischen Beispielen begleitet werden.
- In diesem Kurs werden wir daher auch viele Werkzeuge und echte Datenbanken verwenden.
 - das PostgreSQL Datenbankmanagementsystem^{38,64,68,84},

Einleitung



- Dieser Kurs lehrt die Verwendung von Datenbanken.
- Unser Fokus liegt auf einem praxisorientierten Ansatz.
- Das bedeutet, dass alle Konzepte, die wir diskutieren, immer von einer reichen Auswahl von praktischen Beispielen begleitet werden.
- In diesem Kurs werden wir daher auch viele Werkzeuge und echte Datenbanken verwenden.
 - das PostgreSQL Datenbankmanagementsystem^{38,64,68,84},
 - yEd, ein Graph-Editor mit dem konzeptuelle Schemata erarbeitet werden können^{73,99},

Einleitung



- Dieser Kurs lehrt die Verwendung von Datenbanken.
- Unser Fokus liegt auf einem praxisorientierten Ansatz.
- Das bedeutet, dass alle Konzepte, die wir diskutieren, immer von einer reichen Auswahl von praktischen Beispielen begleitet werden.
- In diesem Kurs werden wir daher auch viele Werkzeuge und echte Datenbanken verwenden.
 - das PostgreSQL Datenbankmanagementsystem^{38,64,68,84},
 - yEd, ein Graph-Editor mit dem konzeptuelle Schemata erarbeitet werden können^{73,99},
 - LibreOffice Base, das als bequeme Oberfläche verwendet werden kann, um mit Datenbanken über Formulare und Berichten zu interagieren^{37,74},

Einleitung



- Dieser Kurs lehrt die Verwendung von Datenbanken.
- Unser Fokus liegt auf einem praxisorientierten Ansatz.
- Das bedeutet, dass alle Konzepte, die wir diskutieren, immer von einer reichen Auswahl von praktischen Beispielen begleitet werden.
- In diesem Kurs werden wir daher auch viele Werkzeuge und echte Datenbanken verwenden.
 - das PostgreSQL Datenbankmanagementsystem^{38,64,68,84},
 - yEd, ein Graph-Editor mit dem konzeptuelle Schemata erarbeitet werden können^{73,99},
 - LibreOffice Base, das als bequeme Oberfläche verwendet werden kann, um mit Datenbanken über Formulare und Berichten zu interagieren^{37,74},
 - Python⁹⁴, eine Programmiersprache, für die das psycopg-Modul⁹² zum Verbinden mit PostgreSQL zur Verfügung steht, bis hin zum

Einleitung



- Dieser Kurs lehrt die Verwendung von Datenbanken.
- Unser Fokus liegt auf einem praxisorientierten Ansatz.
- Das bedeutet, dass alle Konzepte, die wir diskutieren, immer von einer reichen Auswahl von praktischen Beispielen begleitet werden.
- In diesem Kurs werden wir daher auch viele Werkzeuge und echte Datenbanken verwenden.
 - das PostgreSQL Datenbankmanagementsystem^{38,64,68,84},
 - yEd, ein Graph-Editor mit dem konzeptuelle Schemata erarbeitet werden können^{73,99},
 - LibreOffice Base, das als bequeme Oberfläche verwendet werden kann, um mit Datenbanken über Formulare und Berichten zu interagieren^{37,74},
 - Python⁹⁴, eine Programmiersprache, für die das psycopg-Modul⁹² zum Verbinden mit PostgreSQL zur Verfügung steht, bis hin zum
 - PgModeler, einem Werkzeug, mit dem bequem logische Schemata für PostgreSQL-Datenbanken entwickelt werden können¹.

Einleitung



- Dieser Kurs lehrt die Verwendung von Datenbanken.
- Unser Fokus liegt auf einem praxisorientierten Ansatz.
- Das bedeutet, dass alle Konzepte, die wir diskutieren, immer von einer reichen Auswahl von praktischen Beispielen begleitet werden.
- In diesem Kurs werden wir daher auch viele Werkzeuge und echte Datenbanken verwenden.
- Nach dem Abschluss des Kurses sollten Sie in der Lage sein, produktiv mit Datenbanken zu arbeiten.

Einleitung



- Dieser Kurs lehrt die Verwendung von Datenbanken.
- Unser Fokus liegt auf einem praxisorientierten Ansatz.
- Das bedeutet, dass alle Konzepte, die wir diskutieren, immer von einer reichen Auswahl von praktischen Beispielen begleitet werden.
- In diesem Kurs werden wir daher auch viele Werkzeuge und echte Datenbanken verwenden.
- Nach dem Abschluss des Kurses sollten Sie in der Lage sein, produktiv mit Datenbanken zu arbeiten.
- Sie sollten in der Lage sein, einfache Datenbankapplikationen zu entwickeln.

Einleitung



- Dieser Kurs lehrt die Verwendung von Datenbanken.
- Unser Fokus liegt auf einem praxisorientierten Ansatz.
- Das bedeutet, dass alle Konzepte, die wir diskutieren, immer von einer reichen Auswahl von praktischen Beispielen begleitet werden.
- In diesem Kurs werden wir daher auch viele Werkzeuge und echte Datenbanken verwenden.
- Nach dem Abschluss des Kurses sollten Sie in der Lage sein, produktiv mit Datenbanken zu arbeiten.
- Sie sollten in der Lage sein, einfache Datenbankapplikationen zu entwickeln.
- Sie sollten in der Lage sein, das gewaltige Ökosystem verschiedener Datenbankmanagementsystem, Werkzeuge und Paradigmen dieses Gebiets zu navigieren und die richtigen Lösungen für die richtigen Probleme auszuwählen.



Daten



Daten



- Daten sind überall.



Daten



- Daten sind überall.
- Name, Adressen, Bankkonten und -transaktionen, Onlineeinkäufe, Zugtickets, Mobiltelefonnummern, WeChat-Chats, Webseiten, Bücher, Gebrauchsanleitungen, Programm Quelltexte, Karten, Schulnoten, Untersuchungsergebnisse und medizinische Historien, Steuerdaten, berufliche Werdegänge, Spielergebnisse...

Daten



- Daten sind überall.
- Name, Adressen, Bankkonten und -transaktionen, Onlineeinkäufe, Zugtickets, Mobiltelefonnummern, WeChat-Chats, Webseiten, Bücher, Gebrauchsanleitungen, Programm Quelltexte, Karten, Schulnoten, Untersuchungsergebnisse und medizinische Historien, Steuerdaten, berufliche Werdegänge, Spielergebnisse. . . Alles sind Daten.

- Daten sind überall.
- Name, Adressen, Bankkonten und -transaktionen, Onlineeinkäufe, Zugtickets, Mobiltelefonnummern, WeChat-Chats, Webseiten, Bücher, Gebrauchsanleitungen, Programm Quelltexte, Karten, Schulnoten, Untersuchungsergebnisse und medizinische Historien, Steuerdaten, berufliche Werdegänge, Spielergebnisse. . . Alles sind Daten.
- Daten sind vielleicht die wichtigste Ressource unseres digitalen Zeitalters.

- Daten sind überall.
- Name, Adressen, Bankkonten und -transaktionen, Onlineeinkäufe, Zugtickets, Mobiltelefonnummern, WeChat-Chats, Webseiten, Bücher, Gebrauchsanleitungen, Programm Quelltexte, Karten, Schulnoten, Untersuchungsergebnisse und medizinische Historien, Steuerdaten, berufliche Werdegänge, Spielergebnisse... Alles sind Daten.
- Daten sind vielleicht die wichtigste Ressource unseres digitalen Zeitalters.
- Daten müssen gespeichert werden, sortiert, wiedergefunden, gesichert, zusammengefasst, erneuert und verwaltet werden.

- Daten sind überall.
- Name, Adressen, Bankkonten und -transaktionen, Onlineeinkäufe, Zugtickets, Mobiltelefonnummern, WeChat-Chats, Webseiten, Bücher, Gebrauchsanleitungen, Programm Quelltexte, Karten, Schulnoten, Untersuchungsergebnisse und medizinische Historien, Steuerdaten, berufliche Werdegänge, Spielergebnisse... Alles sind Daten.
- Daten sind vielleicht die wichtigste Ressource unseres digitalen Zeitalters.
- Daten müssen gespeichert werden, sortiert, wiedergefunden, gesichert, zusammengefasst, erneuert und verwaltet werden.
- Es gibt ganz verschiedene Arten von Daten.

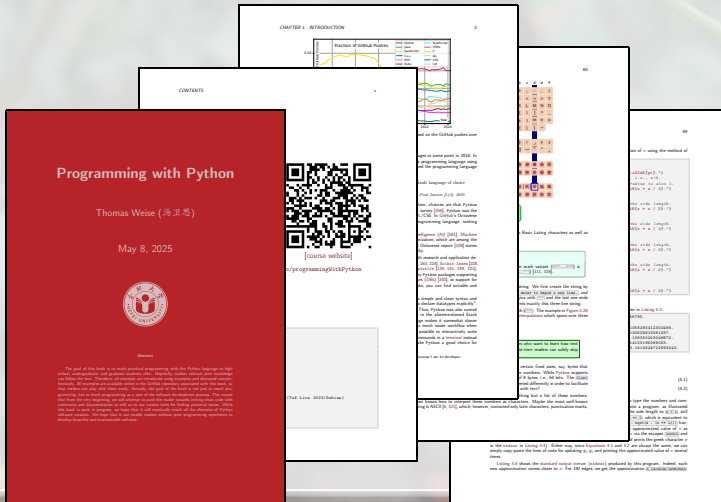
Unstrukturierte Daten



- Unstrukturierte Daten wie z. B. Bücher, Bachelor- und Masterarbeiten, Berichte, Budgetpläne und Forschungsanträge werden oft als einzelne Dokumente gespeichert.



- Unstrukturierte Daten wie z. B. Bücher, Bachelor- und Masterarbeiten, Berichte, Budgetpläne und Forschungsanträge werden oft als einzelne Dokumente gespeichert.



Unstrukturierte Daten



- Unstrukturierte Daten wie z. B. Bücher, Bachelor- und Masterarbeiten, Berichte, Budgetpläne und Forschungsanträge werden oft als einzelne Dokumente gespeichert.
- Oft gibt es Vorlagen für die Struktur solcher Dokumente, aber darüber hinaus können diese sehr unterschiedlich sein.



Unstrukturierte Daten



- Unstrukturierte Daten wie z. B. Bücher, Bachelor- und Masterarbeiten, Berichte, Budgetpläne und Forschungsanträge werden oft als einzelne Dokumente gespeichert.
- Oft gibt es Vorlagen für die Struktur solcher Dokumente, aber darüber hinaus können diese sehr unterschiedlich sein.
- Sie können in Katalogen organisiert werden, in dem man Metainformationen wie Autoren, Jahrgang und Stichworte getrennt abspeichert.

Unstrukturierte Daten



- Unstrukturierte Daten wie z. B. Bücher, Bachelor- und Masterarbeiten, Berichte, Budgetpläne und Forschungsanträge werden oft als einzelne Dokumente gespeichert.
- Oft gibt es Vorlagen für die Struktur solcher Dokumente, aber darüber hinaus können diese sehr unterschiedlich sein.
- Sie können in Katalogen organisiert werden, in dem man Metainformationen wie Autoren, Jahrgang und Stichworte getrennt abspeichert.
- Viel mehr kann man aber nicht tun.

Tabellarische Daten



- Für tabellarische Daten gibe es einfache Textformate wie comma-separated values (CSV)⁷⁵ und Programme wie Microsoft Excel^{10,41} und LibreOffice Calc^{56,74}.

Tabellarische Daten



- Für tabellarische Daten gebe es einfache Textformate wie comma-separated values (CSV)⁷⁵ und Programme wie Microsoft Excel^{10,41} und LibreOffice Calc^{56,74}.

	A	B	C	D	E	F	G	H	I
	algorithm	instance	objective	encoding	n	bestF.m/n	bestF.mean	bestF.med	bestF.geom
1	algorithm	instance	objective	encoding	n	bestF.m/n	bestF.mean	bestF.med	bestF.geom
2	fea1p1_swap2	bra24	weightBasedPrioritization	GameEncoding	7	511704978	537039600.285714	533000499	536879304.80311
3	fea1p1_swap2	circ10	weightBasedPrioritization	GameEncoding	7	266	276	276	275.96424714448
4	fea1p1_swap2	circ12	weightBasedPrioritization	GameEncoding	7	498	501.142857142857	500	501.128206430223
5	fea1p1_swap2	circ14	weightBasedPrioritization	GameEncoding	7	854	863.428571428571	864	863.404254888884
6	fea1p1_swap2	circ16	weightBasedPrioritization	GameEncoding	7	1428	1444.85714285714	1446	1444.78905377176
7	fea1p1_swap2	circ18	weightBasedPrioritization	GameEncoding	7	2260	5645.42857142857	2428	3899.17499073968
8	fea1p1_swap2	circ20	weightBasedPrioritization	GameEncoding	7	98946	110399	114949	109845.831199988
9	fea1p1_swap2	circ22	weightBasedPrioritization	GameEncoding	7	344392	368742.857142857	365623	367980.04183661
10	fea1p1_swap2	circ24	weightBasedPrioritization	GameEncoding	7	806235	853588.428571429	861429	853237.666055972
11	fea1p1_swap2	circ26	weightBasedPrioritization	GameEncoding	7	1516609	1642138.85714286	1657193	1641155.93526768
12	fea1p1_swap2	circ28	weightBasedPrioritization	GameEncoding	7	2552726	2740632	2771943	2739219.59794411
13	fea1p1_swap2	circ30	weightBasedPrioritization	GameEncoding	7	4380409	4527255.71428572	4489041	4526178.6922413
14	fea1p1_swap2	circ32	weightBasedPrioritization	GameEncoding	7	6757897	6972900.85714286	6888851	6971266.88170685
15	fea1p1_swap2	circ34	weightBasedPrioritization	GameEncoding	7	9677587	10114965.1428571	10148595	10111560.1871562
16	fea1p1_swap2	circ36	weightBasedPrioritization	GameEncoding	7	14285985	14698708.8571429	14845057	14696766.2689776
17	fea1p1_swap2	circ38	weightBasedPrioritization	GameEncoding	7	20092941	20594347.5714286	20641476	20592928.8917216
18	fea1p1_swap2	circ4	weightBasedPrioritization	GameEncoding	7	20	20	20	20
19	fea1p1_swap2	circ40	weightBasedPrioritization	GameEncoding	7	27274645	27822677.2857143	27658176	27818114.3972958
20	fea1p1_swap2	circ6	weightBasedPrioritization	GameEncoding	7	64	64	64	64
21	fea1p1_swap2	circ8	weightBasedPrioritization	GameEncoding	7	132	134.571428571429	136	134.559842651306
22	fea1p1_swap2	con10	weightBasedPrioritization	GameEncoding	7	126	127.428571428571	128	127.42648254324
23	fea1p1_swap2	con12	weightBasedPrioritization	GameEncoding	7	188	189	189	188.999244133033
24	fea1p1_swap2	con14	weightBasedPrioritization	GameEncoding	7	263	266.428571428571	266	266.421664745328
25	fea1p1_swap2	con16	weightBasedPrioritization	GameEncoding	7	359	359.714285714286	359	359.714285714286

Hierarchische Daten

- Dann gibt es hierarchische Datenstrukturen.



Hierarchische Daten



- Dann gibt es hierarchische Datenstrukturen.
- Dafür gibt es Formate wie Extensible Markup Language (XML)^{16,27,51}

Hierarchische Daten



- Dann gibt es hierarchische Datenstrukturen.
- Dafür gibt es Formate wie Extensible Markup Language (XML)^{16,27,51}

```
yamlexample.yaml  jsonexample.json  xmlexample.xml x
1  <?xml version="1.0" encoding="UTF-8"?>
   Bind to grammar/schema...
i 2  <course titleEN="Databases" titleCN="数据库原理与应用" year="2025">
3    <teacher name="Weise" />
4    <students>
5      <student studentID="1234567890" name="Bibbo" score="85" />
6      <student studentID="1234567891" name="Bebbo" score="73" />
7      <student studentID="1234567892" name="Bibboto" score="90" />
8      <student studentID="1234567894" name="Bibboba" score="97" />
9    </students>
10 </course>
11
```



Hierarchische Daten

- Dann gibt es hierarchische Datenstrukturen.
- Dafür gibt es Formate wie Extensible Markup Language (XML)^{16,27,51}, JavaScript Object Notation (JSON)^{15,85}

```
yamlexample.yaml  jsonexample.json  xmlexample.xml x
1 <?xml version="1.0" encoding="UTF-8"?>
  <!-- to grammar/schema... -->
2 <?course titleEN="Databases" titleCN="数据库原理与应用" year="2025" >
3   <teacher name="Weise" />
4 </course>
5 <!-- student -->
6 <student studentID="1234567890" name="Bibbo" score="85" />
7 <student studentID="1234567891" name="Bebbo" score="73" />
8 <student studentID="1234567892" name="Bibboto" score="90" />
9 <student studentID="1234567894" name="Bibboba" score="97" />
10 </students>
11 </course>
12
```

```
yamlexample.yaml  jsonexample.json x  xmlexample.xml
1 { "course": {
2   "titleEN": "Databases",
3   "titleCN": "数据库原理与应用",
4   "year": 2025,
5   "teacher": {
6     "name": "Weise"
7   },
8   "students": [
9     { "studentID": 1234567890,
10      "name": "Bibbo",
11      "score": 85 },
12     { "studentID": 1234567891,
13      "name": "Bebbo",
14      "score": 73 },
15     { "studentID": 1234567892,
16      "name": "Bibboto",
17      "score": 90 },
18     { "studentID": 1234567894,
19      "name": "Bibboba"
```

Hierarchische Daten



- Dann gibt es hierarchische Datenstrukturen.
- Dafür gibt es Formate wie Extensible Markup Language (XML)^{16,27,51}, JavaScript Object Notation (JSON)^{15,85} und YAML Ain't Markup Language™ (YAML)^{26,34,51}.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- kind to primary school... -->
3 <?course titleEN="Databases" titleCN="数据库原理与应用" year="2025"?>
4 <?teacher name="Weise"?>
5 <?students?>
6 <?student studentID="1234567890" name="Bibbo" score="85"?>
7 <?student studentID="1234567891" name="Bebbo" score="73"?>
8 <?student studentID="1234567892" name="Bibboto" score="90"?>
9 <?student studentID="1234567894" name="Bibboba" score="97"?>
10 </students?>
11 </course?>
```

```
1 { "course": {
2   "titleEN": "Databases",
3   "titleCN": "数据库原理与应用",
4   "year": 2025,
5   "teacher": {
6     "name": "Weise"
7   },
8   "students": [
9     { "studentID": "1234567890",
10      "name": "Bibbo",
11      "score": 85 },
12     { "studentID": "1234567891",
13      "name": "Bebbo",
14      "score": 73 },
15     { "studentID": "1234567892",
16      "name": "Bibboto",
17      "score": 90 },
18     { "studentID": "1234567894",
19      "name": "Bibboba",
20      "score": 97 }
21   ]
22 }
23 }
```

```
1 course:
2   titleEN: Databases
3   titleCN: 数据库原理与应用
4   year: '2025'
5   teacher:
6     name: Weise
7   students:
8     student:
9       - studentID: '1234567890'
10         name: Bibbo
11         score: '85'
12       - studentID: '1234567891'
13         name: Bebbo
14         score: '73'
15       - studentID: '1234567892'
16         name: Bibboto
17         score: '90'
18       - studentID: '1234567894'
19         name: Bibboba
```

Strukturierte Daten



- Diese tabellarische und hierarchische Datenformate haben gemeinsam, dass sie Informationen in einzelnen Dokumenten speichern.



Strukturierte Daten



- Diese tabellarische und hierarchische Datenformate haben gemeinsam, dass sie Informationen in einzelnen Dokumenten speichern.
- Sie bieten klare und strenge Regeln, wie Daten definiert, strukturiert, gespeichert und geladen werden kann.

Strukturierte Daten



- Diese tabellarische und hierarchische Datenformate haben gemeinsam, dass sie Informationen in einzelnen Dokumenten speichern.
- Sie bieten klare und strenge Regeln, wie Daten definiert, strukturiert, gespeichert und geladen werden kann.
- Die Formate sind oft offen und nicht an einzelne Anbieter gebunden.

Strukturierte Daten



- Diese tabellarische und hierarchische Datenformate haben gemeinsam, dass sie Informationen in einzelnen Dokumenten speichern.
- Sie bieten klare und strenge Regeln, wie Daten definiert, strukturiert, gespeichert und geladen werden kann.
- Die Formate sind oft offen und nicht an einzelne Anbieter gebunden.
- Aber sie sind eben nur nützlich, wenn wir die Daten in einzelnen Dateien speichern können.

Strukturierte Daten



- Diese tabellarische und hierarchische Datenformate haben gemeinsam, dass sie Informationen in einzelnen Dokumenten speichern.
- Sie bieten klare und strenge Regeln, wie Daten definiert, strukturiert, gespeichert und geladen werden kann.
- Die Formate sind oft offen und nicht an einzelne Anbieter gebunden.
- Aber sie sind eben nur nützlich, wenn wir die Daten in einzelnen Dateien speichern können.
- Sie sind ungeeignet, um sehr große Datenmengen zu speichern.

Strukturierte Daten



- Diese tabellarische und hierarchische Datenformate haben gemeinsam, dass sie Informationen in einzelnen Dokumenten speichern.
- Sie bieten klare und strenge Regeln, wie Daten definiert, strukturiert, gespeichert und geladen werden kann.
- Die Formate sind oft offen und nicht an einzelne Anbieter gebunden.
- Aber sie sind eben nur nützlich, wenn wir die Daten in einzelnen Dateien speichern können.
- Sie sind ungeeignet, um sehr große Datenmengen zu speichern.
- Sie sind auch nicht geeignet, um komplexe Zusammenhänge zwischen verschiedenen Arten von Daten zu modellieren.

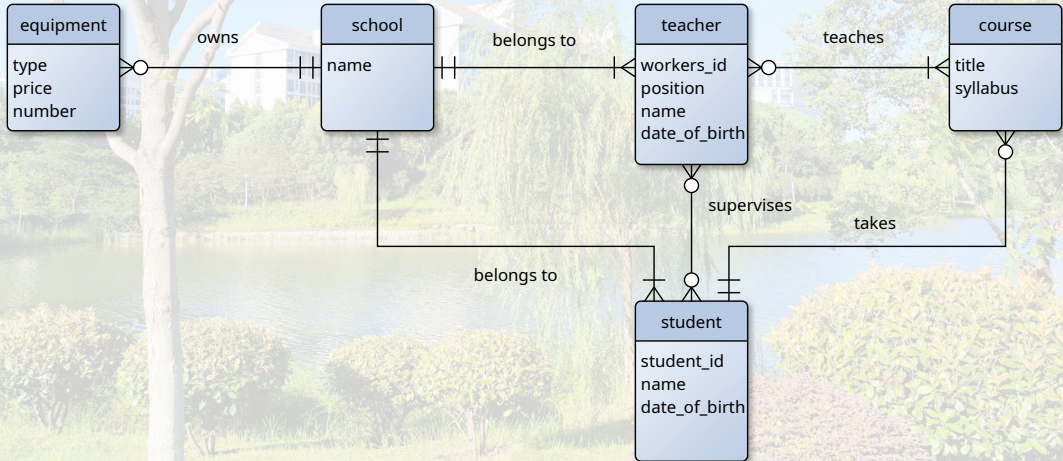
Relationale Daten

- Sagen wir, Sie wollen Daten über das Personal unserer Universität speichern, über die einzelnen Fakultäten, die Studenten und die Geräte.



Relationale Daten

- Sagen wir, Sie wollen Daten über das Personal unserer Universität speichern, über die einzelnen Fakultäten, die Studenten und die Geräte.



Relationale Daten



- Sagen wir, Sie wollen Daten über das Personal unserer Universität speichern, über die einzelnen Fakultäten, die Studenten und die Geräte.
- Ein einzelnes Dokument müsste nicht nur die Informationen speichern, sondern auch wie diese zusammenhängen.

Relationale Daten



- Sagen wir, Sie wollen Daten über das Personal unserer Universität speichern, über die einzelnen Fakultäten, die Studenten und die Geräte.
- Ein einzelnes Dokument müsste nicht nur die Informationen speichern, sondern auch wie diese zusammenhängen.
- Welcher Student besucht welchen Kurs? Welcher Student wird von welchem Lehrer betreut? Welcher Mitarbeiter gehört zu welcher Fakultät?

Relationale Daten



- Sagen wir, Sie wollen Daten über das Personal unserer Universität speichern, über die einzelnen Fakultäten, die Studenten und die Geräte.
- Ein einzelnes Dokument müsste nicht nur die Informationen speichern, sondern auch wie diese zusammenhängen.
- Welcher Student besucht welchen Kurs? Welcher Student wird von welchem Lehrer betreut? Welcher Mitarbeiter gehört zu welcher Fakultät?
- Das kann man durchaus mit XML, YAML, oder JSON machen.

Relationale Daten



- Sagen wir, Sie wollen Daten über das Personal unserer Universität speichern, über die einzelnen Fakultäten, die Studenten und die Geräte.
- Ein einzelnes Dokument müsste nicht nur die Informationen speichern, sondern auch wie diese zusammenhängen.
- Welcher Student besucht welchen Kurs? Welcher Student wird von welchem Lehrer betreut? Welcher Mitarbeiter gehört zu welcher Fakultät?
- Das kann man durchaus mit XML, YAML, oder JSON machen.
- Aber das Dokument wäre riesig.

Relationale Daten



- Sagen wir, Sie wollen Daten über das Personal unserer Universität speichern, über die einzelnen Fakultäten, die Studenten und die Geräte.
- Ein einzelnes Dokument müsste nicht nur die Informationen speichern, sondern auch wie diese zusammenhängen.
- Welcher Student besucht welchen Kurs? Welcher Student wird von welchem Lehrer betreut? Welcher Mitarbeiter gehört zu welcher Fakultät?
- Das kann man durchaus mit XML, YAML, oder JSON machen.
- Aber das Dokument wäre riesig.
- Wie findet man etwas in so einem Dokument?

Relationale Daten



- Sagen wir, Sie wollen Daten über das Personal unserer Universität speichern, über die einzelnen Fakultäten, die Studenten und die Geräte.
- Ein einzelnes Dokument müsste nicht nur die Informationen speichern, sondern auch wie diese zusammenhängen.
- Welcher Student besucht welchen Kurs? Welcher Student wird von welchem Lehrer betreut? Welcher Mitarbeiter gehört zu welcher Fakultät?
- Das kann man durchaus mit XML, YAML, oder JSON machen.
- Aber das Dokument wäre riesig.
- Wie findet man etwas in so einem Dokument?
- Ein Fehler, irgendwo und das Dokument ist kaput.

Relationale Daten



- Sagen wir, Sie wollen Daten über das Personal unserer Universität speichern, über die einzelnen Fakultäten, die Studenten und die Geräte.
- Ein einzelnes Dokument müsste nicht nur die Informationen speichern, sondern auch wie diese zusammenhängen.
- Welcher Student besucht welchen Kurs? Welcher Student wird von welchem Lehrer betreut? Welcher Mitarbeiter gehört zu welcher Fakultät?
- Das kann man durchaus mit XML, YAML, oder JSON machen.
- Aber das Dokument wäre riesig.
- Wie findet man etwas in so einem Dokument?
- Ein Fehler, irgendwo und das Dokument ist kaput.
- Sobald mehrere Leute gleichzeitig mit den Daten arbeiten müssen, fällt alles zusammen.

Relationale Daten



- Sagen wir, Sie wollen Daten über das Personal unserer Universität speichern, über die einzelnen Fakultäten, die Studenten und die Geräte.
- Ein einzelnes Dokument müsste nicht nur die Informationen speichern, sondern auch wie diese zusammenhängen.
- Welcher Student besucht welchen Kurs? Welcher Student wird von welchem Lehrer betreut? Welcher Mitarbeiter gehört zu welcher Fakultät?
- Das kann man durchaus mit XML, YAML, oder JSON machen.
- Aber das Dokument wäre riesig.
- Wie findet man etwas in so einem Dokument?
- Ein Fehler, irgendwo und das Dokument ist kaput.
- Sobald mehrere Leute gleichzeitig mit den Daten arbeiten müssen, fällt alles zusammen.
- Wir brauchen eine andere Lösung.



Datenbanken



Datenbanken



- Willkommen zum Kurs *Datenbanken*⁹³.

Datenbanken



- Willkommen zum Kurs *Datenbanken*⁹³.
- Hier lernen wir über genau so eine Methode

Datenbanken



- Willkommen zum Kurs *Datenbanken*⁹³.
- Hier lernen wir über genau so eine Methode, eine Methode, um strukturierte Daten zu speichern

Datenbanken



- Willkommen zum Kurs *Datenbanken*⁹³.
- Hier lernen wir über genau so eine Methode, eine Methode, um strukturierte Daten zu speichern, so daß mehrere Nutzer gleichzeitig damit arbeiten können.

Datenbanken



- Willkommen zum Kurs *Datenbanken*⁹³.
- Hier lernen wir über genau so eine Methode, eine Methode, um strukturierte Daten zu speichern, so daß mehrere Nutzer gleichzeitig damit arbeiten können.
- Beginnen wir mit ein paar Definitionen.

- Willkommen zum Kurs *Datenbanken*⁹³.
- Hier lernen wir über genau so eine Methode, eine Methode, um strukturierte Daten zu speichern, so daß mehrere Nutzer gleichzeitig damit arbeiten können.
- Beginnen wir mit ein paar Definitionen.

Definition: Datenbank

Eine *Datenbank* (DB) ist eine Kollektion von in gegenseitiger Beziehung stehenden Daten, die in einem Computer gespeichert sind, potentiell viele verschiedene Typen haben können und auf die viele Nutzern und Applikationen gleichzeitig zugreifen können.

Datenbanken

- Es gibt viele verschiedene Datenbanken.



Datenbanken

- Es gibt viele verschiedene Datenbanken.
- Datenbanken für Dokumente



Datenbanken

- Es gibt viele verschiedene Datenbanken.
- Datenbanken für Dokumente, für geographische Daten



Datenbanken



- Es gibt viele verschiedene Datenbanken.
- Datenbanken für Dokumente, für geographische Daten, für komplexe Objekte, usw.



Datenbanken



- Es gibt viele verschiedene Datenbanken.
- Datenbanken für Dokumente, für geographische Daten, für komplexe Objekte, usw.
- Wir fokussieren uns auf Daten, die in (mehreren) Tabellen gespeichert werden, die in festen logischen Beziehungen (Relationen) zu einander stehen.



Datenbanken



- Es gibt viele verschiedene Datenbanken.
- Datenbanken für Dokumente, für geographische Daten, für komplexe Objekte, usw.
- Wir fokussieren uns auf Daten, die in (mehreren) Tabellen gespeichert werden, die in festen logischen Beziehungen (Relationen) zu einander stehen.

Definition: Relationale Datenbank

Eine relationale Datenbank () ist eine Datenbank, die Daten in Zeilen (Rows, Tupel, Datensätze, Records) und Spalten (Columns, Attribute) organisiert, die zusammen Tabellen (Relationen) bilden wobei die Datenpunkte zu einander in Beziehungen stehen^{25,79,83,96}.

Datenbanken



- Es gibt viele verschiedene Datenbanken.
- Datenbanken für Dokumente, für geographische Daten, für komplexe Objekte, usw.
- Wir fokussieren uns auf Daten, die in (mehreren) Tabellen gespeichert werden, die in festen logischen Beziehungen (Relationen) zu einander stehen.

Definition: Relationale Datenbank

Eine relationale Datenbank () ist eine Datenbank, die Daten in Zeilen (Rows, Tupel, Datensätze, Records) und Spalten (Columns, Attribute) organisiert, die zusammen Tabellen (Relationen) bilden wobei die Datenpunkte zu einander in Beziehungen stehen^{25,79,83,96}.

Definition: Datensatz

Ein Datensatz (Record) ist eine Gruppe von zusammenhängenden Datenelementen die von einer Anwendung als Einheit behandelt werden.

Relational Daten



- Ein Studenten-Datensatz könnte z. B. den Namen des Studenten/in speichern, seinen/ihren Geburtstag, die Ausweisnummer und die Mobiltelefonnummer.

Relational Daten



- Ein Studenten-Datensatz könnte z. B. den Namen des Studenten/in speichern, seinen/ihren Geburtstag, die Ausweisnummer und die Mobiltelefonnummer.
- Datensätze sind die grundlegende Einheit von Daten, die in einer Datenbank gespeichert werden.

Relational Daten



- Ein Studenten-Datensatz könnte z. B. den Namen des Studenten/in speichern, seinen/ihren Geburtstag, die Ausweisnummer und die Mobiltelefonnummer.
- Datensätze sind die grundlegende Einheit von Daten, die in einer Datenbank gespeichert werden.
- Das zweite Kernelement sind die Beziehungen der Datensätze untereinander.

Relational Daten



- Ein Studenten-Datensatz könnte z. B. den Namen des Studenten/in speichern, seinen/ihren Geburtstag, die Ausweisnummer und die Mobiltelefonnummer.
- Datensätze sind die grundlegende Einheit von Daten, die in einer Datenbank gespeichert werden.
- Das zweite Kernelement sind die Beziehungen der Datensätze untereinander.
- Diese Beziehungen schützen die Korrektheit und Integrität der Daten.



- Ein Studenten-Datensatz könnte z. B. den Namen des Studenten/in speichern, seinen/ihren Geburtstag, die Ausweisnummer und die Mobiltelefonnummer.
- Datensätze sind die grundlegende Einheit von Daten, die in einer Datenbank gespeichert werden.
- Das zweite Kernelement sind die Beziehungen der Datensätze untereinander.
- Diese Beziehungen schützen die Korrektheit und Integrität der Daten.
- Ein Studenten-Datensatz könnte z. B. mit einem Lehrer-Datensatz verbinden sein.



- Ein Studenten-Datensatz könnte z. B. den Namen des Studenten/in speichern, seinen/ihren Geburtstag, die Ausweisnummer und die Mobiltelefonnummer.
- Datensätze sind die grundlegende Einheit von Daten, die in einer Datenbank gespeichert werden.
- Das zweite Kernelement sind die Beziehungen der Datensätze untereinander.
- Diese Beziehungen schützen die Korrektheit und Integrität der Daten.
- Ein Studenten-Datensatz könnte z. B. mit einem Lehrer-Datensatz verbinden sein.
- Vielleicht speichert unsere Datenbank ja auch, welcher Student von welchem Lehrer betreut wird.



- Ein Studenten-Datensatz könnte z. B. den Namen des Studenten/in speichern, seinen/ihren Geburtstag, die Ausweisnummer und die Mobiltelefonnummer.
- Datensätze sind die grundlegende Einheit von Daten, die in einer Datenbank gespeichert werden.
- Das zweite Kernelement sind die Beziehungen der Datensätze untereinander.
- Diese Beziehungen schützen die Korrektheit und Integrität der Daten.
- Ein Studenten-Datensatz könnte z. B. mit einem Lehrer-Datensatz verbinden sein.
- Vielleicht speichert unsere Datenbank ja auch, welcher Student von welchem Lehrer betreut wird.
- Solche miteinander in Beziehung stehenden Daten, das ist womit wir arbeiten werden.

Datenbankmanagementsysteme



- Natürlich brauchen wir eine Art Software, um mit solchen Daten umzugehen.

Datenbankmanagementsysteme



- Natürlich brauchen wir eine Art Software, um mit solchen Daten umzugehen.

Definition: Datenbankmanagementsystem

Ein Datenbankmanagementsystem (DBMS) ist ein Softwaresystem um mit Daten arbeiten zu können. Es bietet die Möglichkeit, Datenbanken, Tabellen und Datensätze zu erstellen, zu speichern, zu verändern und zu löschen. Es verwaltet auch die Zugriffsrechte auf die Datenbanken und Tabellen.

Datenbankmanagementsysteme



- Natürlich brauchen wir eine Art Software, um mit solchen Daten umzugehen.

Definition: Datenbankmanagementsystem

Ein Datenbankmanagementsystem (DBMS) ist ein Softwaresystem um mit Daten arbeiten zu können. Es bietet die Möglichkeit, Datenbanken, Tabellen und Datensätze zu erstellen, zu speichern, zu verändern und zu löschen. Es verwaltet auch die Zugriffsrechte auf die Datenbanken und Tabellen.

- DBMSes können beliebig komplexe Softwareprogramme sein.

- Natürlich brauchen wir eine Art Software, um mit solchen Daten umzugehen.

Definition: Datenbankmanagementsystem

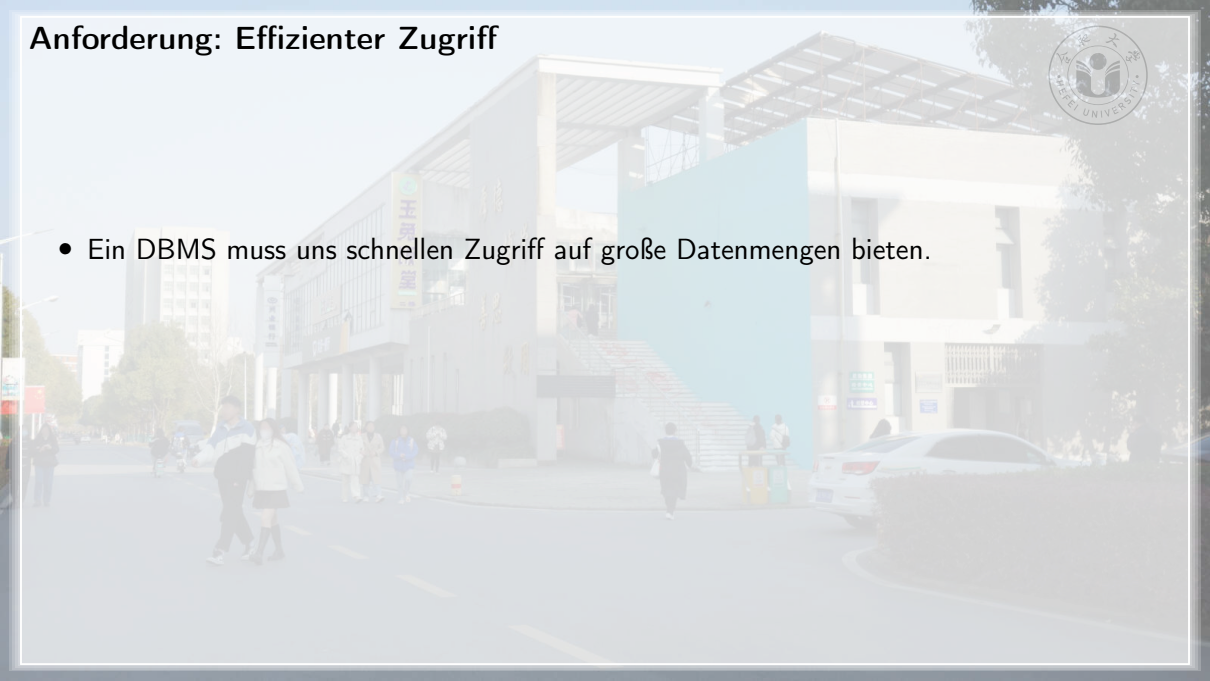
Ein Datenbankmanagementsystem (DBMS) ist ein Softwaresystem um mit Daten arbeiten zu können. Es bietet die Möglichkeit, Datenbanken, Tabellen und Datensätze zu erstellen, zu speichern, zu verändern und zu löschen. Es verwaltet auch die Zugriffsrechte auf die Datenbanken und Tabellen.

- DBMSes können beliebig komplexe Softwareprogramme sein.
- Wenn Sie über die Anforderungen an solche Software nachdenken, wird Ihnen sofort klar, warum.

Anforderung: Effizienter Zugriff



- Ein DBMS muss uns schnellen Zugriff auf große Datenmengen bieten.



Anforderung: Effizienter Zugriff



- Ein DBMS muss uns schnellen Zugriff auf große Datenmengen bieten.
- Sie kennen ja effiziente Datenstrukturen wie Hashes, Suchbäume, sortierte Listen, usw.

Anforderung: Effizienter Zugriff



- Ein DBMS muss uns schnellen Zugriff auf große Datenmengen bieten.
- Sie kennen ja effiziente Datenstrukturen wie Hashes, Suchbäume, sortierte Listen, usw.
- Hier arbeiten wir jedoch mit Datenstructure, die auf der Festplatte gespeichert werden müssen...

Anforderung: Effizienter Zugriff



- Ein DBMS muss uns schnellen Zugriff auf große Datenmengen bieten.
- Sie kennen ja effiziente Datenstrukturen wie Hashes, Suchbäume, sortierte Listen, usw.
- Hier arbeiten wir jedoch mit Datenstructure, die auf der Festplatte gespeichert werden müssen...
- ...und das macht alles schwieriger.

Anforderung: Datenintegrität



- Ein DBMS muss es uns erlauben, Einschränkungen (EN: *constraints*) und Beziehungen für Daten zu spezifizieren.

Anforderung: Datenintegrität



- Ein DBMS muss es uns erlauben, Einschränkungen (EN: *constraints*) und Beziehungen für Daten zu spezifizieren.
- Vielleicht erlauben wir nur Geburtsdaten zwischen dem 1. 1. 1900 und dem 31. 12. 2020 für alle Studenten und Universitätsmitarbeiter.

Anforderung: Datenintegrität



- Ein DBMS muss es uns erlauben, Einschränkungen (EN: *constraints*) und Beziehungen für Daten zu spezifizieren.
- Vielleicht erlauben wir nur Geburtsdaten zwischen dem 1. 1. 1900 und dem 31. 12. 2020 für alle Studenten und Universitätsmitarbeiter.
- Vielleicht wollen wir erzwingen, das jeder Universitätsmitarbeiter immer genau einer Fakultät zugeordnet ist.

Anforderung: Datenintegrität



- Ein DBMS muss es uns erlauben, Einschränkungen (EN: *constraints*) und Beziehungen für Daten zu spezifizieren.
- Vielleicht erlauben wir nur Geburtsdaten zwischen dem 1. 1. 1900 und dem 31. 12. 2020 für alle Studenten und Universitätsmitarbeiter.
- Vielleicht wollen wir erzwingen, das jeder Universitätsmitarbeiter immer genau einer Fakultät zugeordnet ist.
- Das DBMS muss es uns ermöglichen, solche Einschränkungen und Beziehungen zu definieren und muss dann dafür sorgen, dass diese immer eingehalten werden.

Anforderung: Zugriffskontrolle



- Ein DBMS muss es uns auch erlauben, Regeln für den Datenzugriff zu formulieren.

Anforderung: Zugriffskontrolle



- Ein DBMS muss es uns auch erlauben, Regeln für den Datenzugriff zu formulieren.
- Vielleicht speichert unsere Datenbank alle Informationen über unsere Universität und darf von allen Studenten und Mitarbeitern genutzt werden.

Anforderung: Zugriffskontrolle



- Ein DBMS muss es uns auch erlauben, Regeln für den Datenzugriff zu formulieren.
- Vielleicht speichert unsere Datenbank alle Informationen über unsere Universität und darf von allen Studenten und Mitarbeitern genutzt werden.
- Das bedeutet aber nicht, dass alle auch auf alle Daten zugreifen können.

Anforderung: Zugriffskontrolle



- Ein DBMS muss es uns auch erlauben, Regeln für den Datenzugriff zu formulieren.
- Vielleicht speichert unsere Datenbank alle Informationen über unsere Universität und darf von allen Studenten und Mitarbeitern genutzt werden.
- Das bedeutet aber nicht, dass alle auch auf alle Daten zugreifen können.
- Und nur weil jemand bestimmte Daten auslesen darf, bedeutet das noch nicht, dass sie diese Daten auch verändern darf.

Anforderung: Gleichzeitiger Zugriff



- Ein DBMS muss es mehreren Personen und Applikationen ermöglichen, gleichzeitig auf die Daten zuzugreifen.



Anforderung: Gleichzeitiger Zugriff



- Ein DBMS muss es mehreren Personen und Applikationen ermöglichen, gleichzeitig auf die Daten zuzugreifen.
- Es muss verhindern, dass die Datenintegrität verletzt wird, wenn gleichzeitig mehrere Personen die Daten verändern.

Anforderung: Gleichzeitiger Zugriff



- Ein DBMS muss es mehreren Personen und Applikationen ermöglichen, gleichzeitig auf die Daten zuzugreifen.
- Es muss verhindern, dass die Datenintegrität verletzt wird, wenn gleichzeitig mehrere Personen die Daten verändern.
- Manche Datenbanken sind so groß, dass sie über mehrere Computer in einem Cluster verteilt werden müssen.

Anforderung: Gleichzeitiger Zugriff



- Ein DBMS muss es mehreren Personen und Applikationen ermöglichen, gleichzeitig auf die Daten zuzugreifen.
- Es muss verhindern, dass die Datenintegrität verletzt wird, wenn gleichzeitig mehrere Personen die Daten verändern.
- Manche Datenbanken sind so groß, dass sie über mehrere Computer in einem Cluster verteilt werden müssen.
- Das DBMS muss dann die Datenintegrität über alle Datenbankteile hinweg sicherstellen.

- Weil Datenbanken also zwangsweise schon etwas kompliziert sind, gibt es in vielen Organisationen eine Person, die/der sich damit beschäftigt:

- Weil Datenbanken also zwangsweise schon etwas kompliziert sind, gibt es in vielen Organisationen eine Person, die/der sich damit beschäftigt:

Definition: Datenbankadministrator

Datenbankadministratoren (DBAs) sind die Person oder Gruppe von Personen, die für die effektive Nutzung von Datenbanktechnologien in einer Organisation oder einem Unternehmen zuständig ist.



- Weil Datenbanken also zwangsweise schon etwas kompliziert sind, gibt es in vielen Organisationen eine Person, die/der sich damit beschäftigt:

Definition: Datenbankadministrator

Datenbankadministratoren (DBAs) sind die Person oder Gruppe von Personen, die für die effektive Nutzung von Datenbanktechnologien in einer Organisation oder einem Unternehmen zuständig ist.

- Datebanken werden in vielen Organisationen und Abteilungen verwendet.

- Weil Datenbanken also zwangsweise schon etwas kompliziert sind, gibt es in vielen Organisationen eine Person, die/der sich damit beschäftigt:

Definition: Datenbankadministrator

Datenbankadministratoren (DBAs) sind die Person oder Gruppe von Personen, die für die effektive Nutzung von Datenbanktechnologien in einer Organisation oder einem Unternehmen zuständig ist.

- Datenbanken werden in vielen Organisationen und Abteilungen verwendet.
- Daher gibt es auch viele DBAs.



Zusammenfassung



Zusammenfassung



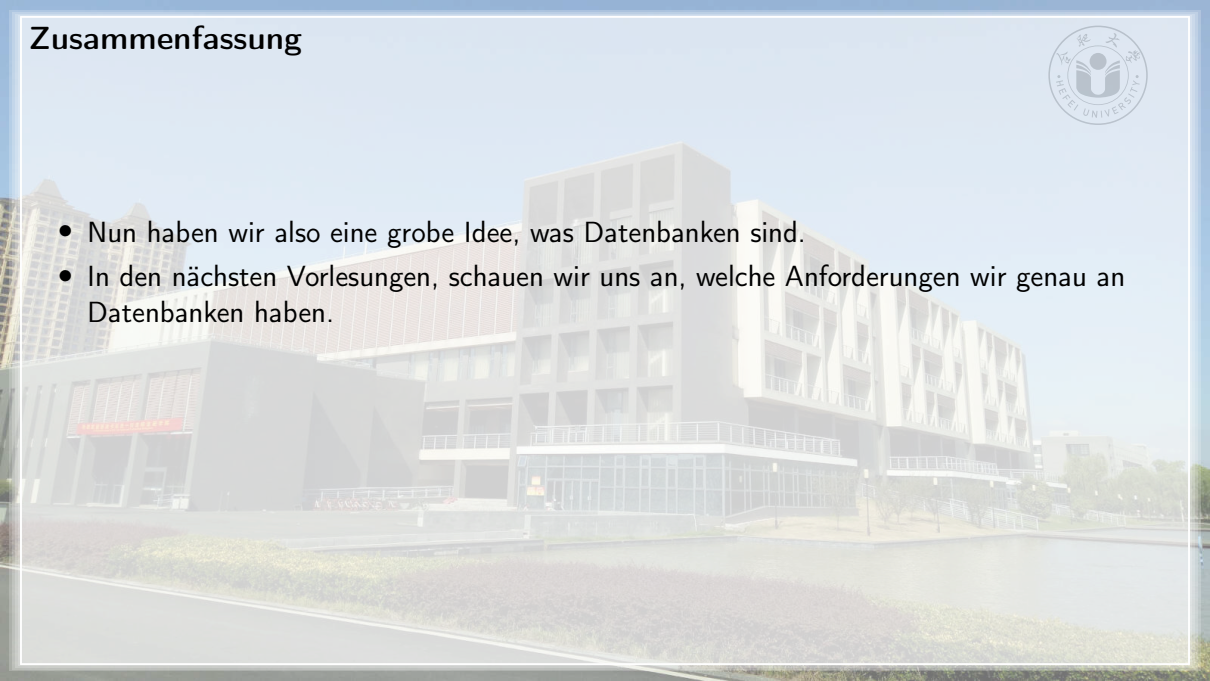
- Nun haben wir also eine grobe Idee, was Datenbanken sind.



Zusammenfassung



- Nun haben wir also eine grobe Idee, was Datenbanken sind.
- In den nächsten Vorlesungen, schauen wir uns an, welche Anforderungen wir genau an Datenbanken haben.



Zusammenfassung



- Nun haben wir also eine grobe Idee, was Datenbanken sind.
- In den nächsten Vorlesungen, schauen wir uns an, welche Anforderungen wir genau an Datenbanken haben.
- Wir schauen uns die Geschichte von Datenbanken an.

Zusammenfassung



- Nun haben wir also eine grobe Idee, was Datenbanken sind.
- In den nächsten Vorlesungen, schauen wir uns an, welche Anforderungen wir genau an Datenbanken haben.
- Wir schauen uns die Geschichte von Datenbanken an.
- Wir schauen uns an, welche DBMSe es so gibt.

- Nun haben wir also eine grobe Idee, was Datenbanken sind.
- In den nächsten Vorlesungen, schauen wir uns an, welche Anforderungen wir genau an Datenbanken haben.
- Wir schauen uns die Geschichte von Datenbanken an.
- Wir schauen uns an, welche DBMSe es so gibt.
- Und dann legen wir richtig los.



谢谢你们！
Thank you!
Vielen Dank!



References I



- [1] 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. 9–13, 118).
- [2] Adam Aspin und Karine Aspin. *Query Answers with MariaDB – Volume I: Introduction to SQL Queries*. Tetras Publishing, Okt. 2018. ISBN: 978-1-9996172-4-0. See also³ (siehe S. 104, 118).
- [3] Adam Aspin und Karine Aspin. *Query Answers with MariaDB – Volume II: In-Depth Querying*. Tetras Publishing, Okt. 2018. ISBN: 978-1-9996172-5-7. See also² (siehe S. 104, 118).
- [4] 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. 117).
- [5] Daniel J. Barrett. *Efficient Linux at the Command Line*. Sebastopol, CA, USA: O'Reilly Media, Inc., Feb. 2022. ISBN: 978-1-0981-1340-7 (siehe S. 118, 119).
- [6] Daniel Bartholomew. *Learning the MariaDB Ecosystem: Enterprise-level Features for Scalability and Availability*. New York, NY, USA: Apress Media, LLC, Okt. 2019. ISBN: 978-1-4842-5514-8 (siehe S. 118).
- [7] Ben Beitler. *Hands-On Microsoft Access 2019*. Birmingham, England, UK: Packt Publishing Ltd, März 2020. ISBN: 978-1-83898-747-3 (siehe S. 117, 118).
- [8] 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. 120).
- [9] 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. 116).
- [10] Bernard Obeng Boateng. *Data Modeling with Microsoft Excel*. Birmingham, England, UK: Packt Publishing Ltd, Nov. 2023. ISBN: 978-1-80324-028-2 (siehe S. 29, 30, 118).
- [11] 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. 120).

References II



- [12] Silvia Botros und Jeremy Tinley. *High Performance MySQL*. 4. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., Nov. 2021. ISBN: **978-1-4920-8051-0** (siehe S. **118**).
- [13] Ed Bott. *Windows 11 Inside Out*. Hoboken, NJ, USA: Microsoft Press, Pearson Education, Inc., Feb. 2023. ISBN: **978-0-13-769132-6** (siehe S. **118**).
- [14] Ron Brash und Ganesh Naik. *Bash Cookbook*. Birmingham, England, UK: Packt Publishing Ltd, Juli 2018. ISBN: **978-1-78862-936-2** (siehe S. **116**).
- [15] 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. **31–35**, **117**).
- [16] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen und Eve Maler, Hrsg. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. W3C Recommendation. Wakefield, MA, USA: World Wide Web Consortium (W3C), 26. Nov. 2008–7. Feb. 2013. URL: <http://www.w3.org/TR/2008/REC-xml-20081126> (besucht am 2024-12-15) (siehe S. **31–35**, **120**).
- [17] 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. **117**).
- [18] Jason Cannon. *High Availability for the LAMP Stack*. Shelter Island, NY, USA: Manning Publications, Juni 2022 (siehe S. **117**, **119**).
- [19] Donald D. Chamberlin. "50 Years of Queries". *Communications of the ACM (CACM)* 67(8):110–121, Aug. 2024. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: **0001-0782**. doi:[10.1145/3649887](https://doi.org/10.1145/3649887). URL: <https://cacm.acm.org/research/50-years-of-queries> (besucht am 2025-01-09) (siehe S. **119**).
- [20] 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](https://doi.org/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. **117**).

References III



- [21] 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 (siehe S. 106, 117).
- [22] 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. See²¹ for a more comprehensive introduction. (Siehe S. 117).
- [23] 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. 117, 118).
- [24] 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. 119, 120).
- [25] Edgar Frank „Ted“ Codd. "A Relational Model of Data for Large Shared Data Banks". *Communications of the ACM (CACM)* 13(6):377–387, Juni 1970. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 0001-0782. doi:10.1145/362384.362685. URL: <https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf> (besucht am 2025-01-05) (siehe S. 59–65, 119).
- [26] Coding Gears und Train Your Brain. *YAML Fundamentals for DevOps, Cloud and IaC Engineers*. Birmingham, England, UK: Packt Publishing Ltd, März 2022. ISBN: 978-1-80324-243-9 (siehe S. 31–35, 120).
- [27] Timothy W. Cole und Myung-Ja K. Han. *XML for Catalogers and Metadata Librarians (Third Millennium Cataloging)*. 1. Aufl. Dublin, OH, USA: Libraries Unlimited, 23. Mai 2013. ISBN: 978-1-59884-519-8 (siehe S. 31–35, 120).
- [28] "[csv](#) – CSV File Reading and Writing". In: *Python 3 Documentation. The Python Standard Library*. Beaverton, OR, USA: Python Software Foundation (PSF), 2001–2025. URL: <https://docs.python.org/3/library/csv.html> (besucht am 2024-11-14) (siehe S. 116).
- [29] *Database Language SQL*. Techn. Ber. ANSI X3.135-1986. Washington, D.C., USA: American National Standards Institute (ANSI), 1986 (siehe S. 119).

References IV



- [30] 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. 119).
- [31] *Database Language SQL*. International Standard ISO 9075-1987. Geneva, Switzerland: International Organization for Standardization (ISO), 1987 (siehe S. 119).
- [32] Paul Deitel, Harvey Deitel und Abbey Deitel. *Internet & World Wide WebW[: How to Program*. 5. Aufl. Hoboken, NJ, USA: Pearson Education, Inc., Nov. 2011. ISBN: 978-0-13-299045-5 (siehe S. 120).
- [33] 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. 118).
- [34] Ingy döt Net, Tina Müller, Pantelis Antoniou, Eemeli Aro, Thomas Smith, Oren Ben-Kiki und Clark C. Evans. *YAML Ain't Markup Language (YAML™) version 1.2*. Revision 1.2.2. Seattle, WA, USA: YAML Language Development Team, 1. Okt. 2021. URL: <https://yaml.org/spec/1.2.2> (besucht am 2025-01-05) (siehe S. 31–35, 120).
- [35] Russell J.T. Dyer. *Learning MySQL and MariaDB*. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2015. ISBN: 978-1-4493-6290-4 (siehe S. 118).
- [36] *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. 117).
- [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. 9–13, 117).
- [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. 9–13, 119).

References V



- [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 (IFIPAICT). Berlin/Heidelberg, Germany: Springer-Verlag GmbH Germany, 2012, S. 29–47. ISSN: 1868-4238. ISBN: 978-3-642-33441-2. doi:10.1007/978-3-642-33442-9_3 (siehe S. 117).
- [40] 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. 116).
- [41] 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. 29, 30, 118).
- [42] 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. 119).
- [43] 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. 119).
- [44] Michael Hausenblas. *Learning Modern Linux*. Sebastopol, CA, USA: O'Reilly Media, Inc., Apr. 2022. ISBN: 978-1-0981-0894-6 (siehe S. 118).
- [45] Christian Heimes. "`defusedxml` 0.7.1: XML Bomb Protection for Python stdlib Modules". In: 8. März 2021. URL: <https://pypi.org/project/defusedxml> (besucht am 2024-12-15) (siehe S. 120).
- [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. 117, 120).
- [47] Manuel Hoffmann, Frank Nagle und Yanuo Zhou. *The Value of Open Source Software*. Working Paper 24-038. Boston, MA, USA: Harvard Business School, 1. Jan. 2024. URL: https://www.hbs.edu/ris/Publication%20Files/24-038_51f8444f-502c-4139-8bf2-56eb4b65c58a.pdf (besucht am 2025-06-04) (siehe S. 118).

References VI



- [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 (siehe S. 119).
- [49] *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. 119).
- [50] 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. 117).
- [51] Katie Kodes. *Intro to XML, JSON, & YAML*. London, England, UK: Payhip, 2019–4. Sep. 2020 (siehe S. 31–35, 120).
- [52] Jay LaCroix. *Mastering Ubuntu Server*. 4. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Sep. 2022. ISBN: 978-1-80323-424-3 (siehe S. 119).
- [53] Joan Lambert und Curtis Frye. *Microsoft Office Step by Step (Office 2021 and Microsoft 365)*. Hoboken, NJ, USA: Microsoft Press, Pearson Education, Inc., Juni 2022. ISBN: 978-0-13-754493-6 (siehe S. 118).
- [54] Kent D. Lee und Steve Hubbard. *Data Structures and Algorithms with Python*. Undergraduate Topics in Computer Science (UTICS). Cham, Switzerland: Springer, 2015. ISBN: 978-3-319-13071-2. doi:10.1007/978-3-319-13072-9 (siehe S. 119).
- [55] 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. 119).
- [56] *LibreOffice – The Document Foundation*. Berlin, Germany: The Document Foundation, 2024. URL: <https://www.libreoffice.org> (besucht am 2024-12-12) (siehe S. 29, 30, 117, 118).

References VII



- [57] 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. 116).
- [58] Mark Lutz. *Learning Python*. 6. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2025. ISBN: 978-1-0981-7130-8 (siehe S. 119).
- [59] *MariaDB Server Documentation*. Milpitas, CA, USA: MariaDB, 2025. URL: <https://mariadb.com/kb/en/documentation> (besucht am 2025-04-24) (siehe S. 118).
- [60] 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. 118).
- [61] 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. 119).
- [62] *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. 118).
- [63] 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. 116).
- [64] 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. 9–13, 119).
- [65] *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. 120).
- [66] 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. 116).

References VIII



- [67] 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 Vizcaíno. "Ten Simple Rules for Taking Advantage of Git and GitHub". *PLOS Computational Biology* 12(7), 14. Juli 2016. San Francisco, CA, USA: Public Library of Science (PLOS). ISSN: **1553-7358**. doi:[10.1371/JOURNAL.PCBI.1004947](https://doi.org/10.1371/JOURNAL.PCBI.1004947) (siehe S. **117**).
- [68] *PostgreSQL Essentials: Leveling Up Your Data Work*. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2024 (siehe S. **9–13, 119**).
- [69] 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. **116**).
- [70] Federico Razzoli. *Mastering MariaDB*. Birmingham, England, UK: Packt Publishing Ltd, Sep. 2014. ISBN: **978-1-78398-154-0** (siehe S. **118**).
- [71] Mike Reichardt, Michael Gundall und Hans D. Schotten. "Benchmarking the Operation Times of NoSQL and MySQL Databases for Python Clients". In: *47th Annual Conference of the IEEE Industrial Electronics Society (IECON'2021)*. 13.–15. Okt. 2021, Toronto, ON, Canada. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE), 2021, S. 1–8. ISSN: **2577-1647**. ISBN: **978-1-6654-3554-3**. doi:[10.1109/IECON48115.2021.9589382](https://doi.org/10.1109/IECON48115.2021.9589382) (siehe S. **118**).
- [72] 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. **116**).
- [73] 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](https://doi.org/10.3233/978-1-61499-611-8-65) (siehe S. **9–13, 120**).
- [74] 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. **9–13, 29, 30, 117, 118**).

References IX



- [75] Yakov Shafranovich. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. Request for Comments (RFC) 4180. Wilmington, DE, USA: Internet Engineering Task Force (IETF), Okt. 2005. URL: <https://www.ietf.org/rfc/rfc4180.txt> (besucht am 2025-02-05) (siehe S. 29, 30, 116).
- [76] 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. 117).
- [77] Ellen Siever, Stephen Figgins, Robert Love und Arnold Robbins. *Linux in a Nutshell*. 6. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., Sep. 2009. ISBN: 978-0-596-15448-6 (siehe S. 118).
- [78] Anna Skoulikari. *Learning Git*. Sebastopol, CA, USA: O'Reilly Media, Inc., Mai 2023. ISBN: 978-1-0981-3391-7 (siehe S. 117).
- [79] John Miles Smith und Philip Yen-Tang Chang. "Optimizing the Performance of a Relational Algebra Database Interface". *Communications of the ACM (CACM)* 18(10):568–579, Okt. 1975. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 0001-0782. doi:10.1145/361020.361025 (siehe S. 59–65, 119).
- [80] "SQL Commands". In: *PostgreSQL Documentation*. 17.4. The PostgreSQL Global Development Group (PGDG), 20. Feb. 2025. Kap. Part VI. Reference. URL: <https://www.postgresql.org/docs/17/sql-commands.html> (besucht am 2025-02-25) (siehe S. 119).
- [81] Ryan K. Stephens und Ronald R. Plew. *Sams Teach Yourself SQL in 21 Days*. 4. Aufl. Sams Tech Yourself. Indianapolis, IN, USA: SAMS Technical Publishing und Hoboken, NJ, USA: Pearson Education, Inc., Okt. 2002. ISBN: 978-0-672-32451-2 (siehe S. 112, 119).
- [82] Ryan K. Stephens, Ronald R. Plew, Bryan Morgan und Jeff Perkins. *SQL in 21 Tagen. Die Datenbank-Abfragesprache SQL vollständig erklärt (in 14/21 Tagen)*. 6. Aufl. Burgthann, Bayern, Germany: Markt+Technik Verlag GmbH, Feb. 1998. ISBN: 978-3-8272-2020-2. Translation of⁸¹ (siehe S. 119).
- [83] Allen Taylor. *Introducing SQL and Relational Databases*. New York, NY, USA: Apress Media, LLC, Sep. 2018. ISBN: 978-1-4842-3841-7 (siehe S. 59–65, 119).

References X



- [84] 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. **9–13, 118, 119**).
- [85] *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. **31–35, 117**).
- [86] 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. **118**).
- [87] 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. **117, 120**).
- [88] 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. **117, 118**).
- [89] *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. **120**).
- [90] *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: ObjecTime 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. **120**).
- [91] Sander van Vugt. *Linux Fundamentals*. 2. Aufl. Hoboken, NJ, USA: Pearson IT Certification, Juni 2022. ISBN: **978-0-13-792931-3** (siehe S. **118**).
- [92] 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. **9–13, 119**).

References XI



- [93] 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. 53–58, 116, 117, 119).
- [94] 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. 9–13, 25, 119).
- [95] 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. 117).
- [96] *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. 59–65, 119).
- [97] Ulf Michael „Monty“ Widenius, David Axmark und Uppsala, Sweden: MySQL AB. *MySQL Reference Manual – Documentation from the Source*. Sebastopol, CA, USA: O'Reilly Media, Inc., 9. Juli 2002. ISBN: 978-0-596-00265-7 (siehe S. 118).
- [98] 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. 116).
- [99] *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. 9–13, 120).
- [100] Pavlo V. Zahorodko und Pavlo V. Merzlykin. "An Approach for Processing and Document Flow Automation for Microsoft Word and LibreOffice Writer File Formats". In: *4th Workshop for Young Scientists in Computer Science & Software Engineering (CS&SE@SW'2021)*. 18. Dez. 2021, Virtual Event and Kryvyi Rih, Ukraine. Hrsg. von Arnold E. Kiv, Serhiy O. Semerikov, Vladimir N. Soloviev und Andrii M. Striuk. Bd. 3077 der Reihe CEUR Workshop Proceedings (CEUR-WS.org). Aachen, Nordrhein-Westfalen, Germany: CEUR-WS Team, Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen, 2022, S. 66–82. ISSN: 1613-0073. URL: <https://ceur-ws.org/Vol-3077/paper12.pdf> (besucht am 2025-10-04) (siehe S. 117, 118).

References XII

- [101] Giorgio Zarrelli. *Mastering Bash*. Birmingham, England, UK: Packt Publishing Ltd, Juni 2017. ISBN: 978-1-78439-687-9 (siehe S. 116).



Glossary (in English) I



API An *Application Programming Interface* is a set of rules or protocols that enables one software application or component to use or communicate with another⁴⁰.

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^{14,63,101}. Learn more at <https://www.gnu.org/software/bash>.

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

client-server architecture is a system design where a central server receives requests from one or multiple clients^{9,57,66,69,72}. These requests and responses are usually sent over network connections. A typical example for such a system is the World Wide Web (WWW), where web servers host websites and make them available to web browsers, the clients. Another typical example is the structure of database (DB) software, where a central server, the DBMS, offers access to the DB to the different clients. Here, the client can be some terminal software shipping with the DBMS, such as `psql`, or the different applications that access the DBs.

CSV *Comma-Separated Values* is a very common and simple text format for exchanging tabular or matrix data⁷⁵. Each row in the text file represents one row in the table or matrix. The elements in the row are separated by a fixed delimiter, usually a comma („,"), sometimes a semicolon („;"). Python offers some out-of-the-box CSV support in the `csv` module²⁸.

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*⁹³.

DBA A *database administrator* is the person or group responsible for the effective use of database technology in an organization or enterprise.

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 DB⁹⁸.

Glossary (in English) II



ERD Entity relationship diagrams show the relationships between objects, e.g., between the tables in a DB and how they reference each other^{4,17,20–22,50,76,95}

Git is a distributed Version Control Systems (VCS) which allows multiple users to work on the same code while preserving the history of the code changes^{78,87}. Learn more at <https://git-scm.com>.

GitHub is a website where software projects can be hosted and managed via the Git VCS^{67,87}. Learn more at <https://github.com>.

IT information technology

JavaScript JavaScript is the predominant programming language used in websites to develop interactive contents for display in browsers³⁶.

JSON *JavaScript Object Notation* is a data interchange format^{15,85} based on JavaScript³⁶ syntax.

LAMP Stack A system setup for web applications: Linux, Apache (a web server), MySQL, and the server-side scripting language PHP^{18,46}.

LibreOffice is an open source office suite^{39,56,74} which is a good and free alternative to Microsoft Office. It offers software such as LibreOffice Writer, LibreOffice Calc, and LibreOffice Base. See⁹³ for more information and installation instructions.

LibreOffice Base is a DBMS that can work on stand-alone files but also connect to other popular relational databases^{37,74}. It is part of LibreOffice^{39,56,74} and has functionality that is comparable to Microsoft Access^{7,23,88}.

LibreOffice Calc is a spreadsheet software that allows you to arrange and perform calculations with data in a tabular grid. It is a free and open source spreadsheet software^{56,74}, i.e., an alternative to Microsoft Excel. It is part of LibreOffice^{39,56,74}.

LibreOffice Writer is a free and open source text writing program¹⁰⁰ and part of LibreOffice^{39,56,74}. It is a good alternative to Microsoft Word.

Glossary (in English) III



Linux is the leading open source operating system, i.e., a free alternative for Microsoft Windows^{5,44,77,86,91}. We recommend using it for this course, for software development, and for research. Learn more at <https://www.linux.org>. Its variant Ubuntu is particularly easy to use and install.

MariaDB An open source relational database management system that has forked off from MySQL^{2,3,6,35,59,70}. See <https://mariadb.org> for more information.

Microsoft Access is a DBMS that can work on DBs stored in single, stand-alone files but also connect to other popular relational databases^{7,23,60,88}. 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^{10,41,53}. It is part of Microsoft Office. A free alternative to this commercial software is LibreOffice Calc^{56,74}.

Microsoft Office is a commercial suite of office software, including Microsoft Excel, Microsoft Word, and Microsoft Access⁵³. LibreOffice is a free and open source alternative.

Microsoft Windows is a commercial proprietary operating system¹³. 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^{33,62,100} and part of Microsoft Office. A free alternative to this commercial software is the LibreOffice Writer.

MySQL An open source relational database management system^{12,35,71,84,97}. MySQL is famous for its use in the LAMP Stack. See <https://www.mysql.com> for more information.

OSS Open source software, i.e., software that can freely be used, whose source code is made available in the internet, and which is usually developed cooperatively over the internet as well⁴⁷. Typical examples are Python, Linux, Git, and PostgreSQL.

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 notation¹. Learn more at <https://pgmodeler.io>.

Glossary (in English) IV



PostgreSQL An open source object-relational DBMS^{38,64,68,84}. 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 specification⁵⁵. Learn more at <https://www.psycopg.org>⁹².

Python The Python programming language^{48,54,58,94}, i.e., what you will learn about in our book⁹⁴. 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^{25,42,43,79,83,93,96}.

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

SQL The *Structured Query Language* is basically a programming language for querying and manipulating relational databases^{19,29–31,49,61,80–83}. It is understood by many DBMSes. You find the Structured Query Language (SQL) commands supported by PostgreSQL in the reference⁸⁰.

terminal A terminal is a text-based window where you can enter commands and execute them^{5,24}. Knowing what a terminal is and how to use it is very essential in any programming- or system administration-related task. If you want to open a terminal under Microsoft Windows, you can Druck auf  + , dann Schreiben von `cmd`, dann Druck auf . Under Ubuntu Linux,  +  +  opens a terminal, which then runs a Bash shell inside.

Glossary (in English) V



- Ubuntu** is a variant of the open source operating system Linux^{24,46}. 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>.
- UML** The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems^{11,65,89,90}
- VCS** A *Version Control System* is a software which allows you to manage and preserve the historical development of your program code⁸⁷. 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.
- WeChat** 微信, produced by Tencent (腾讯公司) is the most-used messenger application in China. It integrates payment capabilities, video and voice chats, as well as social plugins and location-based services. See also <https://weixin.qq.com/>.
- WWW** World Wide Web^{8,32}
- XML** The *Extensible Markup Language* is a text-based language for storing and transporting of data^{16,27,51}. It allows you to define elements in the form `<myElement myAttr="x">...text...</myElement>`. Different from CSV, elements in XML can be hierarchically nested, like `<a><b><c>test</c></b><b>bla</b></a>`, and thus easily represent tree structures. XML is one of most-used data interchange formats. To process XML in Python, use the `defusedxml` library⁴⁵, as it protects against several security issues.
- YAML** *YAML Ain't Markup Language™* is a human-friendly data serialization language for all programming languages^{26,34,51}. It is widely used for configuration files in the DevOps environment. See <https://yaml.org> for more information.
- yEd** is a graph editor for high-quality graph-based diagrams^{73,99}, suitable to draw, e.g., technology-independent ERDs, control flow charts, or Unified Modeling Language (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>.