



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-Kode 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^{10,14,15,21},



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^{10,14,15,21},
 - yEd, ein Graph-Editor mit dem konzeptuelle Schemata erarbeitet werden können^{16,27},



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^{10,14,15,21},
 - yEd, ein Graph-Editor mit dem konzeptuelle Schemata erarbeitet werden können^{16,27},
 - LibreOffice Base, das als bequeme Oberfläche verwendet werden kann, um mit Datenbanken über Formulare und Berichten zu interagieren^{9,17},

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^{10,14,15,21},
 - yEd, ein Graph-Editor mit dem konzeptuelle Schemata erarbeitet werden können^{16,27},
 - LibreOffice Base, das als bequeme Oberfläche verwendet werden kann, um mit Datenbanken über Formulare und Berichten zu interagieren^{9,17},
 - Python²⁵, eine Programmiersprache, für die das psycopg-Modul²³ 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^{10,14,15,21},
 - yEd, ein Graph-Editor mit dem konzeptuelle Schemata erarbeitet werden können^{16,27},
 - LibreOffice Base, das als bequeme Oberfläche verwendet werden kann, um mit Datenbanken über Formulare und Berichten zu interagieren^{9,17},
 - Python²⁵, eine Programmiersprache, für die das psycopg-Modul²³ 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.
- Names, Adressen, Bankkonten und -transaktionen, Onlineeinkäufe, Zugtickets, Mobiltelefonnummern, WeChat-Chats, Webseiten, Bücker, Gebrauchsanleitungen, Programmquelltexte, Karten, Schulnoten, Untersuchungsergebnisse und medizinische Historien, Steuerdaten, berufliche Werdegänge, Spielergebnisse... .

Daten



- Daten sind überall.
- Names, Adressen, Bankkonten und -transaktionen, Onlineeinkäufe, Zugtickets, Mobiltelefonnummern, WeChat-Chats, Webseiten, Bücker, Gebrauchsanleitungen, Programmquelltexte, Karten, Schulnoten, Untersuchungsergebnisse und medizinische Historien, Steuerdaten, berufliche Werdegänge, Spielergebnisse... Alles sind Daten.

Daten



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

Daten



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

Daten



- Daten sind überall.
- Names, Adressen, Bankkonten und -transaktionen, Onlineeinkäufe, Zugtickets, Mobiltelefonnummern, WeChat-Chats, Webseiten, Bücker, Gebrauchsanleitungen, Programmquelltexte, Karten, Schulnoten, Untersuchungsergebnisse und medizinische Historien, Steuerdaten, berufliche Werdegänge, Spielergebnisse... Alles sind Daten.
- Data 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

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

The image displays a composite of several elements related to Python programming:

- Book Cover:** A red book cover for "Programming with Python" by Thomas Weise. The cover features the title in white, the author's name, and a small logo of a university building.
- QR Code:** A black and white QR code located next to the book cover, which links to the course website.
- Code Editor Screenshots:** Two side-by-side screenshots of a code editor showing Python code and its output.
 - Left Screenshot:** Shows a snippet of code with annotations. It includes a plot titled "CHAPTER 1 - INTRODUCTION" showing the fraction of GitHub pull requests over time, and a section on basic Latin characters.
 - Right Screenshot:** Shows another snippet of code with annotations, including a string manipulation example and a section on list comprehensions.





Unstrukturierte Daten

- Unstrukturierte Daten wie z.B. Bücher, Bachelor- und Masterarbeiten, Berichte, Bugetpläne, 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)¹⁸ und Programme wie Microsoft Excel^{2,11} und LibreOffice Calc^{13,17}.

Tabellarische Daten



- Für tabellarische Daten gibe es einfache Textformate wie comma-separated values (CSV)¹⁸ und Programme wie Microsoft Excel^{2,11} und LibreOffice Calc^{13,17}.

Screenshot of LibreOffice Calc showing a table of data. The table has columns labeled A through I. Row 1 contains column headers: algorithm, instance, objective, encoding, n, bestF_min, bestF_mean, bestF_med, bestF_geom, and bestF_std. Rows 2 through 25 contain data points, with row 4 highlighted in red. The formula bar at the top shows the value 501.128206430223. The status bar at the bottom indicates an average of 501.128206430223 and a sum of 501.128206430223.

	A	B	C	D	E	F	G	H	I	J
1	algorithm	instance	objective	encoding	n	bestF_min	bestF_mean	bestF_med	bestF_geom	bestF_std
2	fea1p1_swap2	bra24	weightBasedPrioritization	GameEncoding	7	511704976	537039600.285714	533000499	536879304.80311	100.000000
3	fea1p1_swap2	circ10	weightBasedPrioritization	GameEncoding	7	266	276	276	275.96424714448	0.000000
4	fea1p1_swap2	circ12	weightBasedPrioritization	GameEncoding	7	496	501.142857142857	500	501.128206430223	0.000000
5	fea1p1_swap2	circ14	weightBasedPrioritization	GameEncoding	7	854	863.428571428571	864	863.404254888884	0.000000
6	fea1p1_swap2	circ16	weightBasedPrioritization	GameEncoding	7	1428	1444.85714285714	1446	1444.78905377176	0.000000
7	fea1p1_swap2	circ18	weightBasedPrioritization	GameEncoding	7	2260	5645.42857142857	2428	3899.17499073968	100.000000
8	fea1p1_swap2	circ20	weightBasedPrioritization	GameEncoding	7	98946	110399	114949	109845.831199988	100.000000
9	fea1p1_swap2	circ22	weightBasedPrioritization	GameEncoding	7	344392	368742.857142857	365623	367980.04183661	100.000000
10	fea1p1_swap2	circ24	weightBasedPrioritization	GameEncoding	7	806259	853588.428571429	861429	853237.666055972	100.000000
11	fea1p1_swap2	circ26	weightBasedPrioritization	GameEncoding	7	1516600	1642138.85714288	1657193	1641155.93526768	100.000000
12	fea1p1_swap2	circ28	weightBasedPrioritization	GameEncoding	7	2552726	2740632	2771943	2739219.59794411	100.000000
13	fea1p1_swap2	circ30	weightBasedPrioritization	GameEncoding	7	4380409	4527255.71428572	4489041	4526178.6922413	100.000000
14	fea1p1_swap2	circ32	weightBasedPrioritization	GameEncoding	7	6757897	6972900.85714286	6888851	6971266.88170685	100.000000
15	fea1p1_swap2	circ34	weightBasedPrioritization	GameEncoding	7	9677587	10114965.1428571	10148595	10111560.1871562	100.000000
16	fea1p1_swap2	circ36	weightBasedPrioritization	GameEncoding	7	14285985	14698708.8571429	14845057	14696766.2689776	100.000000
17	fea1p1_swap2	circ38	weightBasedPrioritization	GameEncoding	7	20092941	20594347.5714286	20641476	20592928.8917216	100.000000
18	fea1p1_swap2	circ4	weightBasedPrioritization	GameEncoding	7	20	20	20	20	0.000000
19	fea1p1_swap2	circ40	weightBasedPrioritization	GameEncoding	7	27274645	27822677.2857143	27658176	27818114.3972958	100.000000
20	fea1p1_swap2	circ6	weightBasedPrioritization	GameEncoding	7	64	64	64	64	0.000000
21	fea1p1_swap2	circ8	weightBasedPrioritization	GameEncoding	7	132	134.571428571429	136	134.559842651306	0.000000
22	fea1p1_swap2	con10	weightBasedPrioritization	GameEncoding	7	126	127.428571428571	128	127.42648254324	0.000000
23	fea1p1_swap2	con12	weightBasedPrioritization	GameEncoding	7	188	189	189	188.999244133033	0.000000
24	fea1p1_swap2	con14	weightBasedPrioritization	GameEncoding	7	263	266.428571428571	266	266.421664745328	0.000000
25	fea1p1_swap2	con16	weightBasedPrioritization	GameEncoding	7	260	260.71428571429	260	260.711222403041	0.000000

Hierarchische Daten

- Dann gibt es hierarchische Datenstrukturen.



Hierarchische Daten

- Dann gibt es hierarchische Datenstrukturen.
- Dafür gibt es Formate wie Extensible Markup Language (XML)^{4,7,12}



Hierarchische Daten



- Dann gibt es hierarchische Datenstrukturen.
- Dafür gibt es Formate wie Extensible Markup Language (XML)^{4,7,12}

A screenshot of a code editor window. At the top, there are three tabs: 'yamlexample.yaml' (selected), 'jsonexample.json', and 'xmlexample.xml'. The 'xmlexample.xml' tab has a red border around it. Below the tabs, the XML code is displayed.

```
1 <?xml version="1.0" encoding="UTF-8"?>
Bind to grammar/schema...
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)^{4,7,12}, JavaScript Object Notation (JSON)^{3,22}

```
yamlexample.yaml jsonexample.json xmlexample.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml version="1.0" encoding="UTF-8"?>
3 <?xml version="1.0" encoding="UTF-8"?>
4 <?xml version="1.0" encoding="UTF-8"?>
5 <?xml version="1.0" encoding="UTF-8"?>
6 <?xml version="1.0" encoding="UTF-8"?>
7 <?xml version="1.0" encoding="UTF-8"?>
8 <?xml version="1.0" encoding="UTF-8"?>
9 <?xml version="1.0" encoding="UTF-8"?>
10 <?xml version="1.0" encoding="UTF-8"?>
11 <?xml version="1.0" encoding="UTF-8"?>
```

```
yaml example.yaml jsonexample.json xml example.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)^{4,7,12}, JavaScript Object Notation (JSON)^{3,22}, und YAML Ain't Markup Language™ (YAML)^{6,8,12}.

```
yamlexample.yaml jsonexample.json xmlexample.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <course>
3   <titleEN>Database</titleEN> <titleCN>数据库原理与应用</titleCN>
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="99" />
9     <student studentID="1234567894" name="Bibboha" score="97" />
10   </students>
11 </course>
```

```
yamlexample.yaml jsonexample.json xmlexample.xml
1 { "course": {
2   "titleEN": "Database",
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": 99 },
18     { "studentID": "1234567894",
19       "name": "Bibboha",
20       "score": 97 }
21   ]
22 }
```

```
yamlexample.yaml jsonexample.json 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 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: Bibboha
```

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.

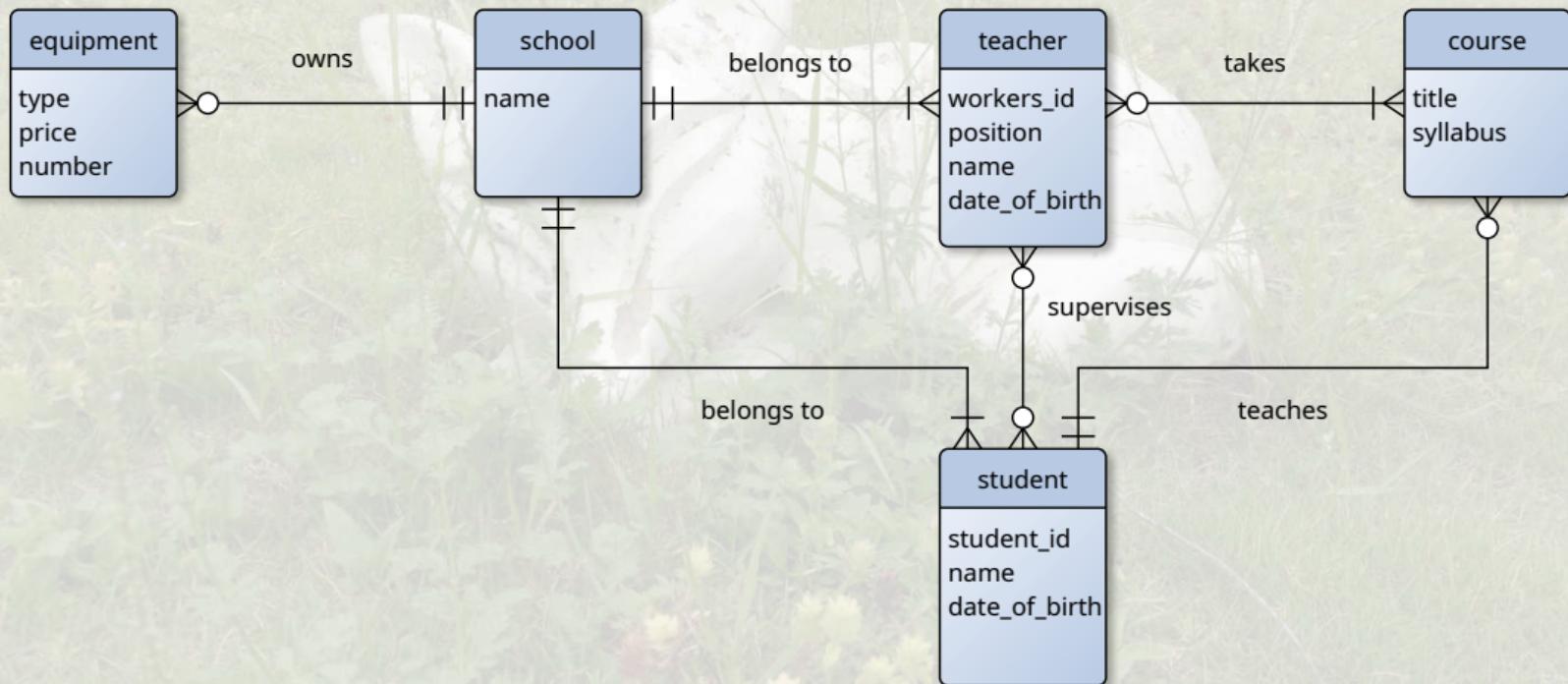
Relationale Daten



- 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*²⁴.



Datenbanken



- Willkommen zum Kurs *Datenbanken*²⁴.
- Hier lernen wir über genau so eine Methode

Datenbanken

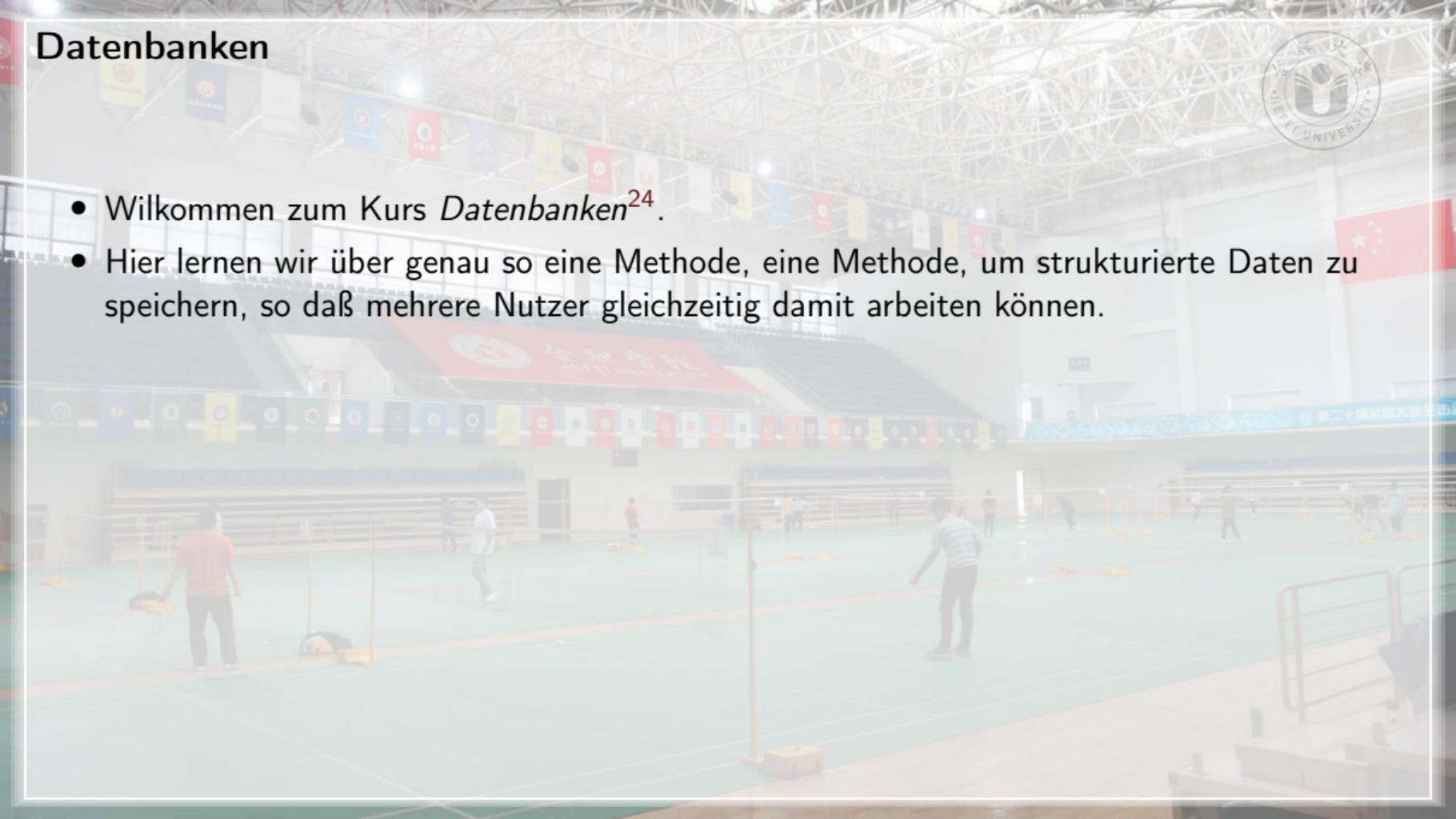


- Willkommen zum Kurs *Datenbanken*²⁴.
- Hier lernen wir über genau so eine Methode, eine Methode, um strukturierte Daten zu speichern

Datenbanken



- Willkommen zum Kurs *Datenbanken*²⁴.
- 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*²⁴.
- 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*²⁴.
- 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 gegenseiter 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 Objecte, usw.





Datenbanken

- Es gibt viele verschiedene Datenbanken.
- Datenbanken für Dokumente, für geographische Daten, für komplexe Objecte, 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 Objecte, 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 (RDB) 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^{5,19,20,26}.



Datenbanken

- Es gibt viele verschiedene Datenbanken.
- Datenbanken für Dokumente, für geographische Daten, für komplexe Objecte, 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 (RDB) 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^{5,19,20,26}.

Definition: Datensatz

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

Relational Data



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

Relational Data



- 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 Data



- 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 Data



- 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.

Relational Data



- 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.

Relational Data



- 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.

Relational Data



- 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.



- 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.



- 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 Datenstructuren, 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 Datenstructuren, die auf der Festplatte gespeichert werden müssen...
- ... und das macht alles schwieriger.

Anforderung: Datenintegrität



- Ein DBMS muss es uns erlauben, Einschränkungen und Beziehungen für Daten zu spezifizieren.

Anforderung: Datenintegrität



- Ein DBMS muss es uns erlauben, Einschränkungen 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 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, dass jeder Universitätsmitarbeiter immer genau einer Fakultät zugeordnet ist.

Anforderung: Datenintegrität



- Ein DBMS muss es uns erlauben, Einschränkungen 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, dass jeder Universitätsmitarbeiter immer genau einer Fakultät zugeordnet ist.
- Das DMBS 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 große, 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 große, 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.

- Dateibanken 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.

- Dateibanken werden in vielen Organisationen und Abteilungen verwendet.
- Daher gibe 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 DBMSes es so gibt.

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 DBMSes es so gibt.
- Und dann legen wir richtig los.



谢谢您门！

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).
- [2] 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).
- [3] 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).
- [4] 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).
- [5] 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).
- [6] 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).
- [7] 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).
- [8] 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).
- [9] Steve Fanning, Vasudev Narayanan, "flywire", Olivier Hallot, Jean Hollis Weber, Jenna Sargent, Pulkit Krishna, Dan Lewis, Peter Schofield, Jochen Schifflers, 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).

References II



- [10] 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).
- [11] 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).
- [12] Katie Kodes. *Intro to XML, JSON, & YAML*. London, England, UK: Payhip, 2019–4. Sep. 2020 (siehe S. 31–35).
- [13] LibreOffice – The Document Foundation. Berlin, Germany: The Document Foundation, 2024. URL: <https://www.libreoffice.org> (besucht am 2024-12-12) (siehe S. 29, 30).
- [14] 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).
- [15] *PostgreSQL Essentials: Leveling Up Your Data Work*. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2024 (siehe S. 9–13).
- [16] 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. *Frontiers in Artificial Intelligence and Applications*. Amsterdam, The Netherlands: IOS Press BV, S. 65–76. ISSN: 0922-6389. ISBN: 978-1-61499-610-1. doi:10.3233/978-1-61499-611-8-65 (siehe S. 9–13).
- [17] 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).
- [18] 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).
- [19] 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).

References III



- [20] 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).
- [21] 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).
- [22] *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).
- [23] 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).
- [24] 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 und 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).
- [25] 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 und 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).
- [26] *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).
- [27] *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).