



合肥大學  
HEFEI UNIVERSITY



# Datenbanken

## 7. Fabrik-Datenbank (Teil 1)

Thomas Weise (汤卫思)

[tweise@hfuu.edu.cn](mailto: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. Nutzer und Datenbank Erstellen
3. Zusammenfassung





# Einleitung



# Einleitung



- In vielen Kursen über *Datenbanken* wird das Gebiet in einzelnen Stücken diskutiert.



# Einleitung



- In vielen Kursen über *Datenbanken* wird das Gebiet in einzelnen Stücken diskutiert.
- Themen wie der Datenbank-Entwurfsprozess, Datenmodellierung, Entity-Relationship-Diagramme,  $\sigma$ -Algebra, SQL-Befehle, und Normalisation werden besprochen.

# Einleitung



- In vielen Kursen über *Datenbanken* wird das Gebiet in einzelnen Stücken diskutiert.
- Themen wie der Datenbank-Entwurfsprozess, Datenmodellierung, Entity-Relationship-Diagramme,  $\sigma$ -Algebra, SQL-Befehle, und Normalisation werden besprochen.
- Praktische Beispiele werden die Übung verlagert.

# Einleitung



- In vielen Kursen über *Datenbanken* wird das Gebiet in einzelnen Stücken diskutiert.
- Themen wie der Datenbank-Entwurfsprozess, Datenmodellierung, Entity-Relationship-Diagramme,  $\sigma$ -Algebra, SQL-Befehle, und Normalisation werden besprochen.
- Praktische Beispiele werden die Übung verlagert.
- Das ist OK.



# Einleitung



- In vielen Kursen über *Datenbanken* wird das Gebiet in einzelnen Stücken diskutiert.
- Themen wie der Datenbank-Entwurfsprozess, Datenmodellierung, Entity-Relationship-Diagramme,  $\sigma$ -Algebra, SQL-Befehle, und Normalisation werden besprochen.
- Praktische Beispiele werden die Übung verlagert.
- Das ist OK.
- Als ich Student war, habe ich alles, was ich über Datenbanken wissen musste aber anders gelernt.

# Einleitung



- In vielen Kursen über *Datenbanken* wird das Gebiet in einzelnen Stücken diskutiert.
- Themen wie der Datenbank-Entwurfsprozess, Datenmodellierung, Entity-Relationship-Diagramme,  $\sigma$ -Algebra, SQL-Befehle, und Normalisation werden besprochen.
- Praktische Beispiele werden die Übung verlagert.
- Das ist OK.
- Als ich Student war, habe ich alles, was ich über Datenbanken wissen musste aber anders gelernt.
- Durch Ausprobieren und Herumspielen.

# Einleitung



- In vielen Kursen über *Datenbanken* wird das Gebiet in einzelnen Stücken diskutiert.
- Themen wie der Datenbank-Entwurfsprozess, Datenmodellierung, Entity-Relationship-Diagramme,  $\sigma$ -Algebra, SQL-Befehle, und Normalisation werden besprochen.
- Praktische Beispiele werden die Übung verlagert.
- Das ist OK.
- Als ich Student war, habe ich alles, was ich über Datenbanken wissen musste aber anders gelernt.
- Durch Ausprobieren und Herumspielen.
- Und deshalb machen wir das jetzt auch.

# Was sind Datenbanken und wie benutzt man sie?



- Bisher sind *Datenbanken* für Sie nur irgendwelche abstrakten Dinge zum Speichern von Daten, auf die irgendwie über ein Klienten-Programm zugegriffen wird.



# Was sind Datenbanken und wie benutzt man sie?



- Bisher sind *Datenbanken* für Sie nur irgendwelche abstrakten Dinge zum Speichern von Daten, auf die irgendwie über ein Klienten-Programm zugegriffen wird.
- Alles sehr eigenartig.



# Was sind Datenbanken und wie benutzt man sie?



- Bisher sind *Datenbanken* für Sie nur irgendwelche abstrakten Dinge zum Speichern von Daten, auf die irgendwie über ein Klienten-Programm zugegriffen wird.
- Alles sehr eigenartig.
- Relationale Datenbanken speichern Tabellen haben wir gesagt.

# Was sind Datenbanken und wie benutzt man sie?



- Bisher sind *Datenbanken* für Sie nur irgendwelche abstrakten Dinge zum Speichern von Daten, auf die irgendwie über ein Klienten-Programm zugegriffen wird.
- Alles sehr eigenartig.
- Relationale Datenbanken speichern Tabellen haben wir gesagt.
- Aber irgendwie ganz anders als ein vernünftiges Microsoft Excel-Spreadsheet.

# Was sind Datenbanken und wie benutzt man sie?



- Bisher sind *Datenbanken* für Sie nur irgendwelche abstrakten Dinge zum Speichern von Daten, auf die irgendwie über ein Klienten-Programm zugegriffen wird.
- Alles sehr eigenartig.
- Relationale Datenbanken speichern Tabellen haben wir gesagt.
- Aber irgendwie ganz anders als ein vernünftiges Microsoft Excel-Spreadsheet.
- Wahrscheinlich ist Ihnen immer noch nicht klar, was Datenbanken eigentlich sind.

# Was sind Datenbanken und wie benutzt man sie?



- Bisher sind *Datenbanken* für Sie nur irgendwelche abstrakten Dinge zum Speichern von Daten, auf die irgendwie über ein Klienten-Programm zugegriffen wird.
- Alles sehr eigenartig.
- Relationale Datenbanken speichern Tabellen haben wir gesagt.
- Aber irgendwie ganz anders als ein vernünftiges Microsoft Excel-Spreadsheet.
- Wahrscheinlich ist Ihnen immer noch nicht klar, was Datenbanken eigentlich sind.
- Und wie man die benutzt.



# Was sind Datenbanken und wie benutzt man sie?



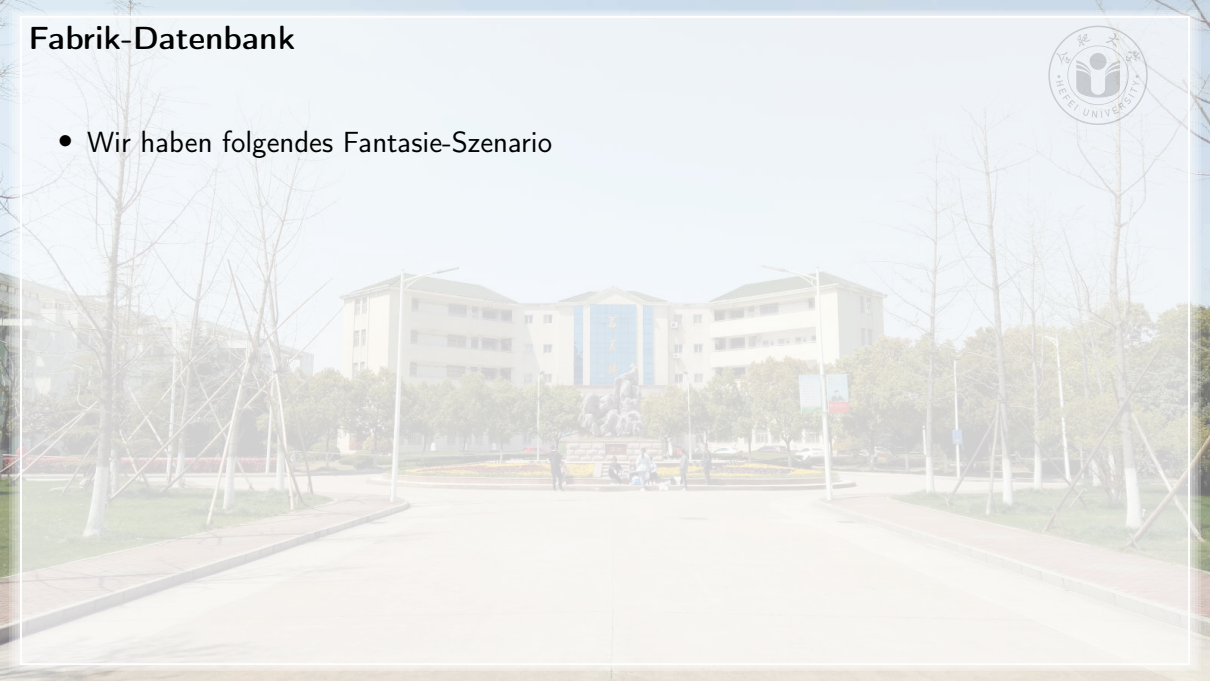
- Bisher sind *Datenbanken* für Sie nur irgendwelche abstrakten Dinge zum Speichern von Daten, auf die irgendwie über ein Klienten-Programm zugegriffen wird.
- Alles sehr eigenartig.
- Relationale Datenbanken speichern Tabellen haben wir gesagt.
- Aber irgendwie ganz anders als ein vernünftiges Microsoft Excel-Spreadsheet.
- Wahrscheinlich ist Ihnen immer noch nicht klar, was Datenbanken eigentlich sind.
- Und wie man die benutzt.
- Lassen Sie uns das ändern.



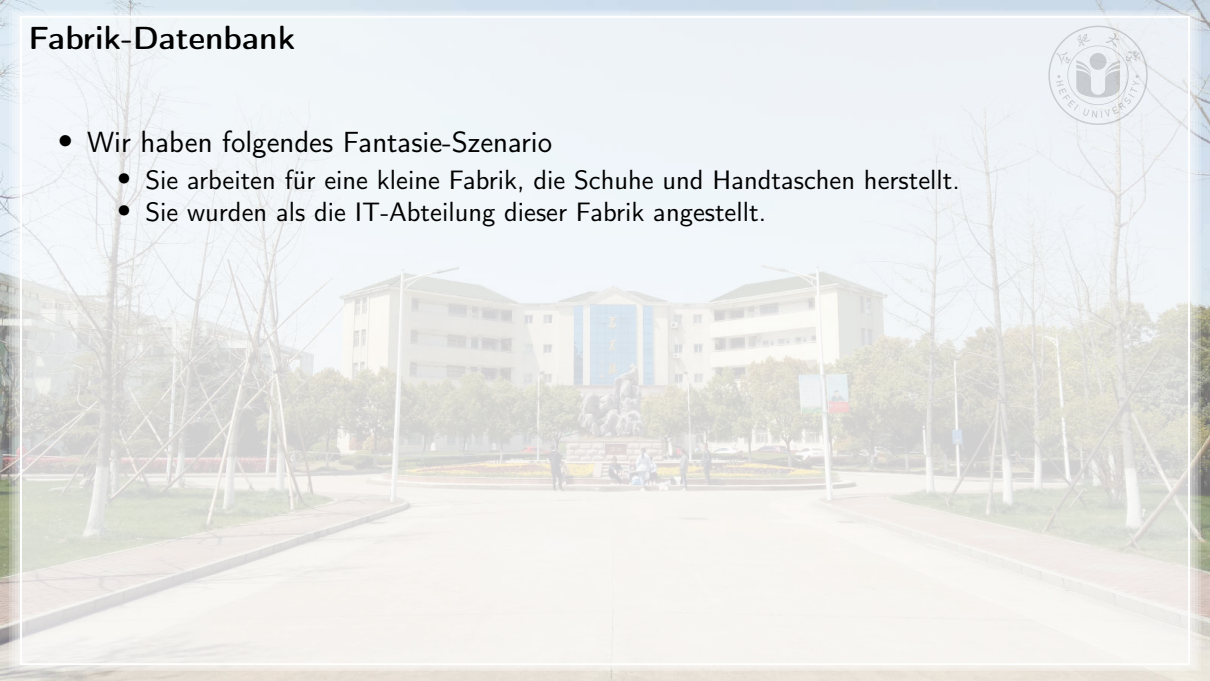
# Fabrik-Datenbank



- Wir haben folgendes Fantasie-Szenario



- Wir haben folgendes Fantasie-Szenario
  - Sie arbeiten für eine kleine Fabrik, die Schuhe und Handtaschen herstellt.
  - Sie wurden als die IT-Abteilung dieser Fabrik angestellt.



# Fabrik-Datenbank



- Wir haben folgendes Fantasie-Szenario
  - Sie arbeiten für eine kleine Fabrik, die Schuhe und Handtaschen herstellt.
  - Sie wurden als die IT-Abteilung dieser Fabrik angestellt.
  - An Ihrem ersten Arbeitstag kommt ihr Chef und sagt: *Mache bitte eine Datenbank, um alle unsere Produktvarianten, Kundendaten, und Bestellungen zu speichern.*

- Wir haben folgendes Fantasie-Szenario
  - Sie arbeiten für eine kleine Fabrik, die Schuhe und Handtaschen herstellt.
  - Sie wurden als die IT-Abteilung dieser Fabrik angestellt.
  - An Ihrem ersten Arbeitstag kommt ihr Chef und sagt: *Mache bitte eine Datenbank, um alle unsere Produktvarianten, Kundendaten, und Bestellungen zu speichern.*
- Wir benutzen dazu PostgreSQL, weil es ein sehr weit verbreitetes DBMS ist.

- Wir haben folgendes Fantasie-Szenario
  - Sie arbeiten für eine kleine Fabrik, die Schuhe und Handtaschen herstellt.
  - Sie wurden als die IT-Abteilung dieser Fabrik angestellt.
  - An Ihrem ersten Arbeitstag kommt ihr Chef und sagt: *Mache bitte eine Datenbank, um alle unsere Produktvarianten, Kundendaten, und Bestellungen zu speichern.*
- Wir benutzen dazu PostgreSQL, weil es ein sehr weit verbreitetes DBMS ist. Im es war auf Rang 1 im "Stack Overflow 2024 Developer Survey"<sup>6</sup> und auf Rang 2 in<sup>3</sup>.





- Wir haben folgendes Fantasie-Szenario
  - Sie arbeiten für eine kleine Fabrik, die Schuhe und Handtaschen herstellt.
  - Sie wurden als die IT-Abteilung dieser Fabrik angestellt.
  - An Ihrem ersten Arbeitstag kommt ihr Chef und sagt: *Mache bitte eine Datenbank, um alle unsere Produktvarianten, Kundendaten, und Bestellungen zu speichern.*
- Wir benutzen dazu PostgreSQL, weil es ein sehr weit verbreitetes DBMS ist. Im es war auf Rang 1 im “Stack Overflow 2024 Developer Survey”<sup>6</sup> und auf Rang 2 in<sup>3</sup>.

## Nützliches Werkzeug

PostgreSQL<sup>1,2,5,7</sup> ist ein fortschrittliches relationales DBMS. Es ist kostenfrei, Open Source, und die Grundlage für alle Beispieldatenbanken in diesem Kurse.



# Nutzer und Datenbank Erstellen



# Nutzer Erstellen



- Der erste Schritt, diese Anforderung zu erstellen, wäre es, eine neue Datenbank anzulegen.

# Nutzer Erstellen



- Der erste Schritt, diese Anforderung zu erstellen, wäre es, eine neue Datenbank anzulegen.
- Wir haben ja PostgreSQL schon installiert, das wird sich also machen lassen.

# Nutzer Erstellen



- Der erste Schritt, diese Anforderung zu erstellen, wäre es, eine neue Datenbank anzulegen.
- Wir haben ja PostgreSQL schon installiert, das wird sich also machen lassen.
- Unser Chef möchte vollen Zugriff auf diese neue Datenbank haben.



# Nutzer Erstellen



- Der erste Schritt, diese Anforderung zu erstellen, wäre es, eine neue Datenbank anzulegen.
- Wir haben ja PostgreSQL schon installiert, das wird sich also machen lassen.
- Unser Chef möchte vollen Zugriff auf diese neue Datenbank haben.
- Er ist aber kein Datenbankadministrator,

# Nutzer Erstellen



- Der erste Schritt, diese Anforderung zu erstellen, wäre es, eine neue Datenbank anzulegen.
- Wir haben ja PostgreSQL schon installiert, das wird sich also machen lassen.
- Unser Chef möchte vollen Zugriff auf diese neue Datenbank haben.
- Er ist aber kein Datenbankadministrator,
- Wir wollen ihm also Zugriff auf die neue Datenbank geben, aber lieber nicht die Kontrolle über das gesamte DBMS.

# Nutzer Erstellen



- Der erste Schritt, diese Anforderung zu erstellen, wäre es, eine neue Datenbank anzulegen.
- Wir haben ja PostgreSQL schon installiert, das wird sich also machen lassen.
- Unser Chef möchte vollen Zugriff auf diese neue Datenbank haben.
- Er ist aber kein Datenbankadministrator,
- Wir wollen ihm also Zugriff auf die neue Datenbank geben, aber lieber nicht die Kontrolle über das gesamte DBMS.
- Deshalb ist der erste Schritt ein anderer

# Nutzer Erstellen



- Der erste Schritt, diese Anforderung zu erstellen, wäre es, eine neue Datenbank anzulegen.
- Wir haben ja PostgreSQL schon installiert, das wird sich also machen lassen.
- Unser Chef möchte vollen Zugriff auf diese neue Datenbank haben.
- Er ist aber kein Datenbankadministrator,
- Wir wollen ihm also Zugriff auf die neue Datenbank geben, aber lieber nicht die Kontrolle über das gesamte DBMS.
- Deshalb ist der erste Schritt ein anderer: Wir erstellen ein neues Benutzerkonto (ein Rolle) speziell für die geplante Datenbank auf unserem DBMS.

# Nutzer Erstellen



- Der erste Schritt, diese Anforderung zu erstellen, wäre es, eine neue Datenbank anzulegen.
- Wir haben ja PostgreSQL schon installiert, das wird sich also machen lassen.
- Unser Chef möchte vollen Zugriff auf diese neue Datenbank haben.
- Er ist aber kein Datenbankadministrator,
- Wir wollen ihm also Zugriff auf die neue Datenbank geben, aber lieber nicht die Kontrolle über das gesamte DBMS.
- Deshalb ist der erste Schritt ein anderer: Wir erstellen ein neues Benutzerkonto (ein Rolle) speziell für die geplante Datenbank auf unserem DBMS.
- Wenn unter diesem Benutzer etwas schief geht, oder ein Angreifer dessen Passwort irgendwie erhält, dann ist der Schaden zumindest auf diese eine Datenbank limitiert.



# Verbinding to PostgreSQL via psql



- Wir sind auf dem Datenbank-Server-Computer engeloggt.



# Verbinding to PostgreSQL via psql



- Wir sind auf dem Datenbank-Server-Computer engeloggt.
- Nehmen Sie an, dass das Passwort des Datenbankadministratorbenutzers textilpostgres auf `XXX` gesetzt ist.

# Verbinding to PostgreSQL via psql



- Wir sind auf dem Datenbank-Server-Computer eingeloggt.
- Nehmen Sie an, dass das Passwort des Datenbankadministratorbenutzers textilpostgres auf `xxx` gesetzt ist. Benutzen Sie niemals etwas wie `xxx` als Passwort.

# Verbinding to PostgreSQL via psql



- Wir sind auf dem Datenbank-Server-Computer angemeldet.
- Nehmen Sie an, dass das Passwort des Datenbankadministratorbenutzers textilpostgres auf `xxx` gesetzt ist. Benutzen Sie niemals etwas wie `xxx` als Passwort.
- Wir öffnen ein Terminal (unter Ubuntu Linux via `Ctrl` + `Alt` + `T`, unter Ubuntu Linux; unter Microsoft Windows durch Druck auf `Windows` + `R`, dann Schreiben von `cmd`, dann Druck auf `↵`).

# Verbindung to PostgreSQL via psql



- Wir sind auf dem Datenbank-Server-Computer angemeldet.
- Nehmen Sie an, dass das Passwort des Datenbankadministratorbenutzers `textilpostgres` auf `xxx` gesetzt ist. Benutzen Sie niemals etwas wie `xxx` als Passwort.
- Wir öffnen ein Terminal (unter Ubuntu Linux via `Ctrl` + `Alt` + `T`, unter Ubuntu Linux; unter Microsoft Windows durch Druck auf `Windows` + `R`, dann Schreiben von `cmd`, dann Druck auf `↵`).
- Wir wollen nun den Klienten `psql` mit der passenden Verbindungs-URI<sup>4</sup> starten, um auf den PostgreSQL-Server zuzugreifen.



# Verbindung to PostgreSQL via psql



- Wir sind auf dem Datenbank-Server-Computer angemeldet.
- Nehmen Sie an, dass das Passwort des Datenbankadministratorbenutzers `textilpostgres` auf `xxx` gesetzt ist. Benutzen Sie niemals etwas wie `xxx` als Passwort.
- Wir öffnen ein Terminal (unter Ubuntu Linux via `Ctrl` + `Alt` + `T`, unter Ubuntu Linux; unter Microsoft Windows durch Druck auf `Windows` + `R`, dann Schreiben von `cmd`, dann Druck auf `↵`).
- Wir wollen nun den Klienten `psql` mit der passenden Verbindungs-URI<sup>4</sup> starten, um auf den PostgreSQL-Server zuzugreifen.

## Nützliches Werkzeug

`psql` ist ein textbasiertes Konsolenprogramm mit dem man sich auf den PostgreSQL-Server verbinden kann. Von der `psql`-Konsole können wir SQL-Befehle an den PostgreSQL-Server schicken und dessen Ausgaben empfangen.

# Verbindung to PostgreSQL via psql



- Wir wollen nun den Klienten psql mit der passenden Verbindungs-URI<sup>4</sup> starten, um auf den PostgreSQL-Server zuzugreifen.

# Verbindung to PostgreSQL via psql



- Wir wollen nun den Klienten psql mit der passenden Verbindungs-URI<sup>4</sup> starten, um auf den PostgreSQL-Server zuzugreifen.
- Die Verbindungs-URI ist `postgres://postgres:XXX@localhost`

# Verbindung to PostgreSQL via psql



- Wir wollen nun den Klienten psql mit der passenden Verbindungs-URI<sup>4</sup> starten, um auf den PostgreSQL-Server zuzugreifen.
- Die Verbindungs-URI ist `postgres://postgres:XXX@localhost`, wobei
  - `postgres://` identifiziert den parameter als Verbindungs-URI.
  - Das zweite “`postgres`” ist der Benutzername.
  - Der Doppelpunkt “`:`” trennt den Benutzername von dem Password `XXX`.



# Verbindung to PostgreSQL via psql



- Wir wollen nun den Klienten psql mit der passenden Verbindungs-URI<sup>4</sup> starten, um auf den PostgreSQL-Server zuzugreifen.
- Die Verbindungs-URI ist `postgres://postgres:XXX@localhost`, wobei
  - `postgres://` identifiziert den parameter als Verbindungs-URI.
  - Das zweite “`postgres`” ist der Benutzername.
  - Der Doppelpunkt “`:`” trennt den Benutzername von dem Passwort `XXX`. Benutzen Sie niemals etwas wie `XXX` als Passwort.



# Verbindung to PostgreSQL via psql



- Wir wollen nun den Klienten psql mit der passenden Verbindungs-URI<sup>4</sup> starten, um auf den PostgreSQL-Server zuzugreifen.
- Die Verbindungs-URI ist `postgres://postgres:XXX@localhost`, wobei
  - `postgres://` identifiziert den parameter als Verbindungs-URI.
  - Das zweite “`postgres`” ist der Benutzername.
  - Der Doppelpunkt “`:`” trennt den Benutzername von dem Passwort `XXX`. Benutzen Sie niemals etwas wie `XXX` als Passwort.
  - Nach dem “`@`” kommt die Netzwerkadresse, auf der der PostgreSQL-Server läuft.

# Verbindung to PostgreSQL via psql



- Wir wollen nun den Klienten psql mit der passenden Verbindungs-URI<sup>4</sup> starten, um auf den PostgreSQL-Server zuzugreifen.
- Die Verbindungs-URI ist `postgres://postgres:XXX@localhost`, wobei
  - `postgres://` identifiziert den parameter als Verbindungs-URI.
  - Das zweite “`postgres`” ist der Benutzername.
  - Der Doppelpunkt “`:`” trennt den Benutzername von dem Passwort `XXX`. Benutzen Sie niemals etwas wie `XXX` als Passwort.
  - Nach dem “`@`” kommt die Netzwerkadresse, auf der der PostgreSQL-Server läuft. `localhost` steht für den aktuellen Computer.

# Erstellen des Benutzers



# Erstellen des Benutzers vis Script







# Zusammenfassung





# Zusammenfassung



- Fertig.



谢谢您们！  
Thank you!  
Vielen Dank!



# References I



- [1] 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. **19–24**).
- [2] 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. **19–24**).
- [3] Camila A. Paiva, Raquel Maximino, Frederico Paiva, Rafael Accetta Vieira, Nicole Espanha, João Felipe Pimentel, Igor Wiese, Marco Aurélio Gerosa, Igor Steinmacher, Leonardo Murta und Vanessa Braganholo. "Analyzing the Adoption of Database Management Systems throughout the History of Open Source Projects". *Empirical Software Engineering: An International Journal* 30(3):71, Feb. 2025. London, England, UK: Springer Nature Limited. ISSN: **1382-3256**. doi:**10.1007/S10664-025-10627-Z**. URL: <https://www.authorea.com/users/677798/articles/674742> (besucht am 2025-06-04) (siehe S. **19–24**).
- [4] *PostgreSQL Documentation*. 17.4. The PostgreSQL Global Development Group (PGDG), Feb. 2025. URL: <https://www.postgresql.org/docs/17/index.html> (besucht am 2025-02-25) (siehe S. **34–45**).
- [5] *PostgreSQL Essentials: Leveling Up Your Data Work*. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2024 (siehe S. **19–24**).
- [6] "Stack Overflow 2024 Developer Survey". In: *Stack Overflow*. New York, NY, USA: Stack Exchange Inc., Mai–Juni 2024. URL: <https://survey.stackoverflow.co/2024> (besucht am 2025-06-01) (siehe S. **19–24**).
- [7] 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. **19–24**).