



合肥大學
HEFEI UNIVERSITY



Datenbanken

20. Fabrik-Datenbank: ERDs in LibreOffice Base

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. ERDs in LibreOffice Base
3. Zusammenfassung





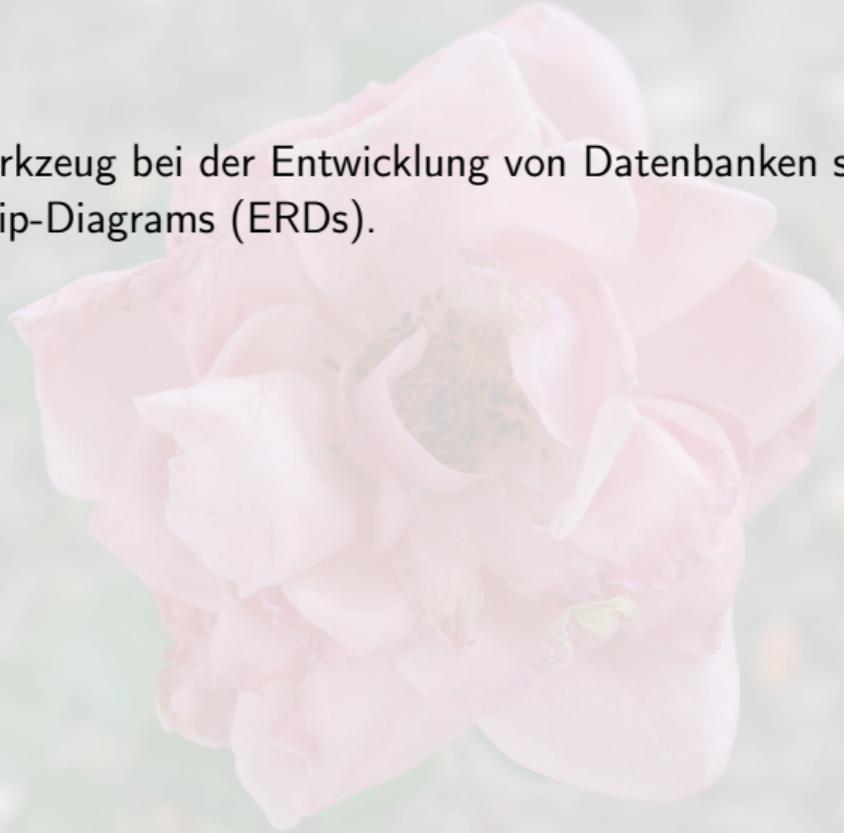
Einleitung



Entity-Relationship-Diagramme



- Ein wichtiges Werkzeug bei der Entwicklung von Datenbanken sind Entity-Relationship-Diagrams (ERDs).



Entity-Relationship-Diagramme



- Ein wichtiges Werkzeug bei der Entwicklung von Datenbanken sind Entity-Relationship-Diagrams (ERDs).
- Sie werden sehr oft verwendet, wenn wir erste, abstrakte, konzeptuelle Modelle einer Datenbank entwickeln.

Entity-Relationship-Diagramme



- Ein wichtiges Werkzeug bei der Entwicklung von Datenbanken sind Entity-Relationship-Diagrams (ERDs).
- Sie werden sehr oft verwendet, wenn wir erste, abstrakte, konzeptuelle Modelle einer Datenbank entwickeln.
- Solche Diagramme stellen Objekte und deren Beziehungen visuell dar.

Entity-Relationship-Diagramme



- Ein wichtiges Werkzeug bei der Entwicklung von Datenbanken sind Entity-Relationship-Diagrams (ERDs).
- Sie werden sehr oft verwendet, wenn wir erste, abstrakte, konzeptuelle Modelle einer Datenbank entwickeln.
- Solche Diagramme stellen Objekte und deren Beziehungen visuell dar.
- Es ist tatsächlich sehr sinnvoll, erstmal ein Modell Datenbank zu entwickeln, bevor wir mit SQL loslegen.

Entity-Relationship-Diagramme

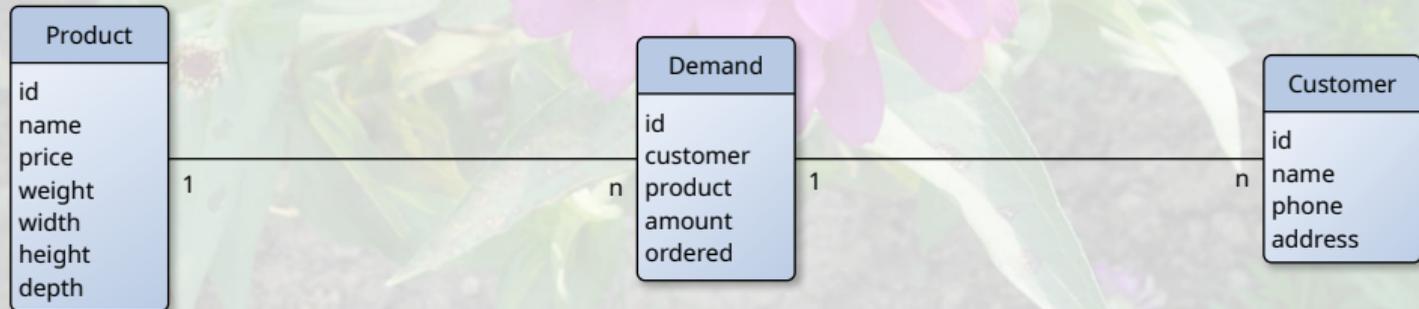


- Ein wichtiges Werkzeug bei der Entwicklung von Datenbanken sind Entity-Relationship-Diagrams (ERDs).
- Sie werden sehr oft verwendet, wenn wir erste, abstrakte, konzeptuelle Modelle einer Datenbank entwickeln.
- Solche Diagramme stellen Objekte und deren Beziehungen visuell dar.
- Es ist tatsächlich sehr sinnvoll, erstmal ein Modell Datenbank zu entwickeln, bevor wir mit SQL loslegen.
- Die modellierten Objekte können dann auf tabellen und die Beziehungen auf Fremdschlüssel abgebildet werden.

Beispiel: Fabrikdatenbank



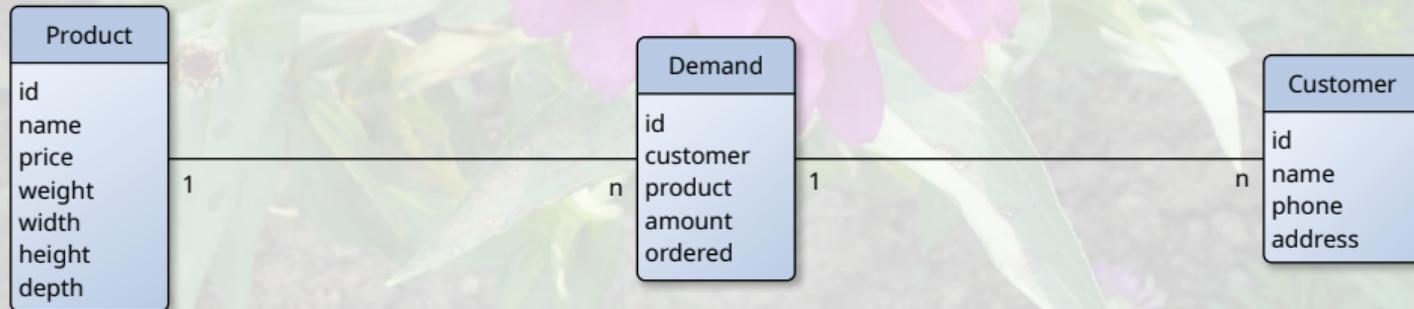
- Hier sehen wir ein ERD für die Objekte in unserer Fabrikdatenbank.



Beispiel: Fabrikdatenbank



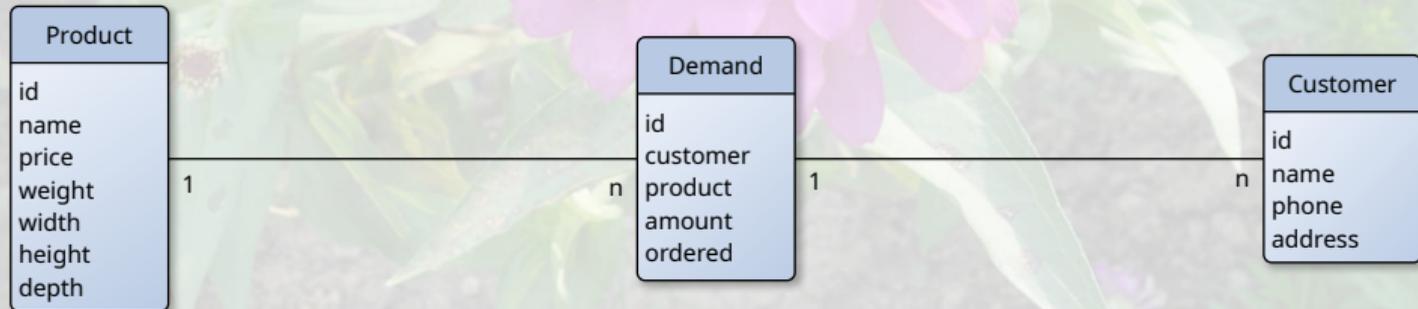
- Hier sehen wir ein ERD für die Objekte in unserer Fabrikdatenbank.
- Alle drei Objekttypen sind dargestellt.



Beispiel: Fabrikdatenbank



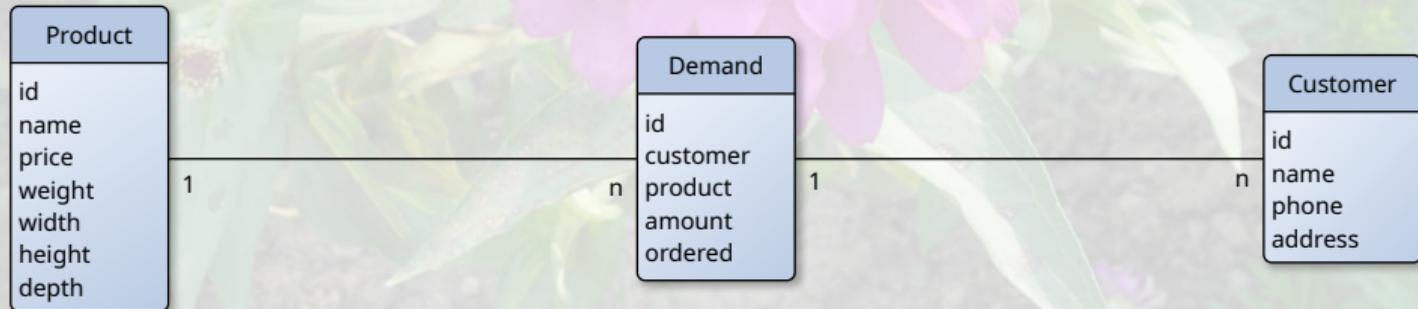
- Hier sehen wir ein ERD für die Objekte in unserer Fabrikdatenbank.
- Alle drei Objekttypen sind dargestellt.
- Die Linien, die sie verbinden, symbolisieren die Beziehungen.



Beispiel: Fabrikdatenbank



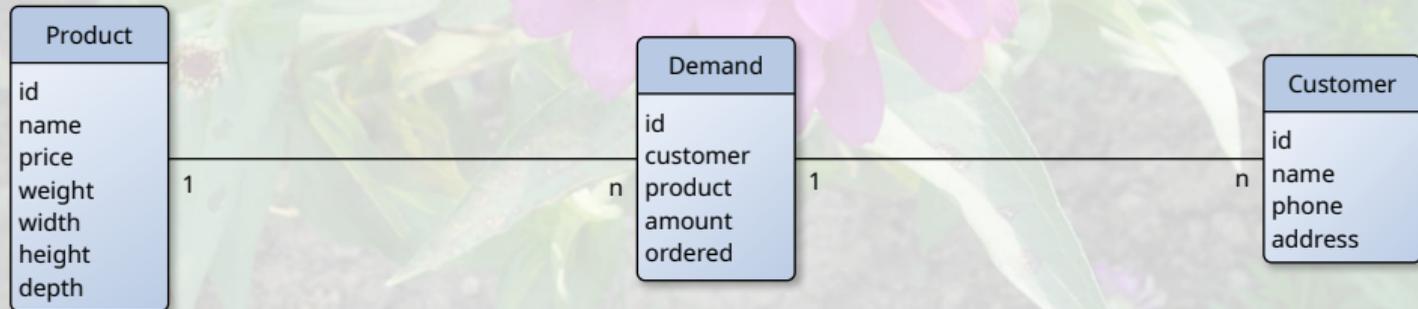
- Hier sehen wir ein ERD für die Objekte in unserer Fabrikdatenbank.
- Alle drei Objekttypen sind dargestellt.
- Die Linien, die sie verbinden, symbolisieren die Beziehungen.
- Hier sehen wir das ein Kunde (und Produkt) mit beliebig vielen Objekten des Typs *Demand* verbunden sein kann.



Beispiel: Fabrikdatenbank



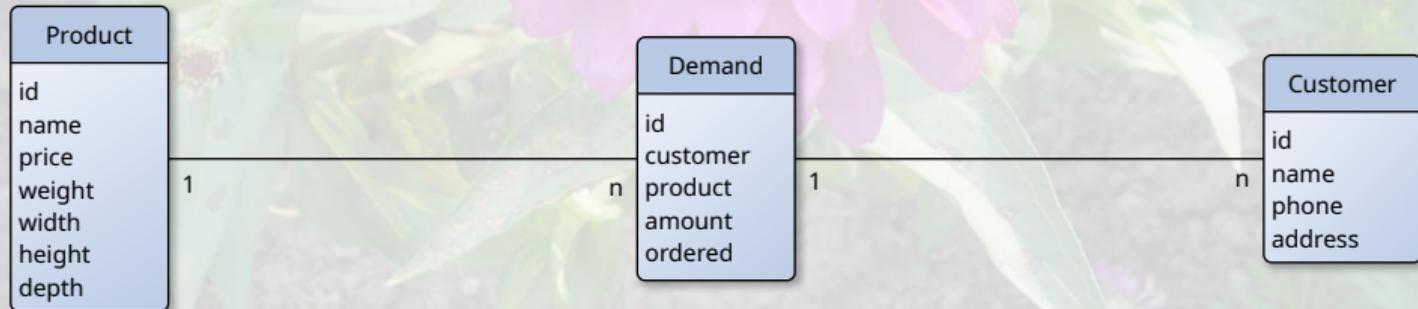
- Hier sehen wir ein ERD für die Objekte in unserer Fabrikdatenbank.
- Alle drei Objekttypen sind dargestellt.
- Die Linien, die sie verbinden, symbolisieren die Beziehungen.
- Hier sehen wir das ein Kunde (und Produkt) mit beliebig vielen Objekten des Typs *Demand* verbunden sein kann.
- Jedes solche Demand-Objekt ist mit genau einem Kunde (und Produkt) verbunden.



Beispiel: Fabrikdatenbank



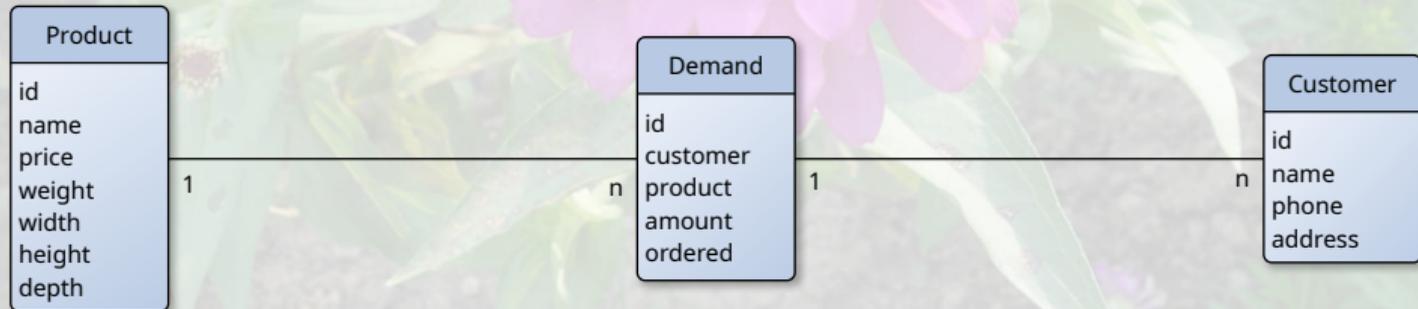
- Hier sehen wir ein ERD für die Objekte in unserer Fabrikdatenbank.
- Alle drei Objekttypen sind dargestellt.
- Die Linien, die sie verbinden, symbolisieren die Beziehungen.
- Hier sehen wir das ein Kunde (und Produkt) mit beliebig vielen Objekten des Typs *Demand* verbunden sein kann.
- Jedes solche Demand-Objekt ist mit genau einem Kunde (und Produkt) verbunden.
- Wir können die Struktur der Datenbank auf Basis so eines Designs erstellen.



Beispiel: Fabrikdatenbank



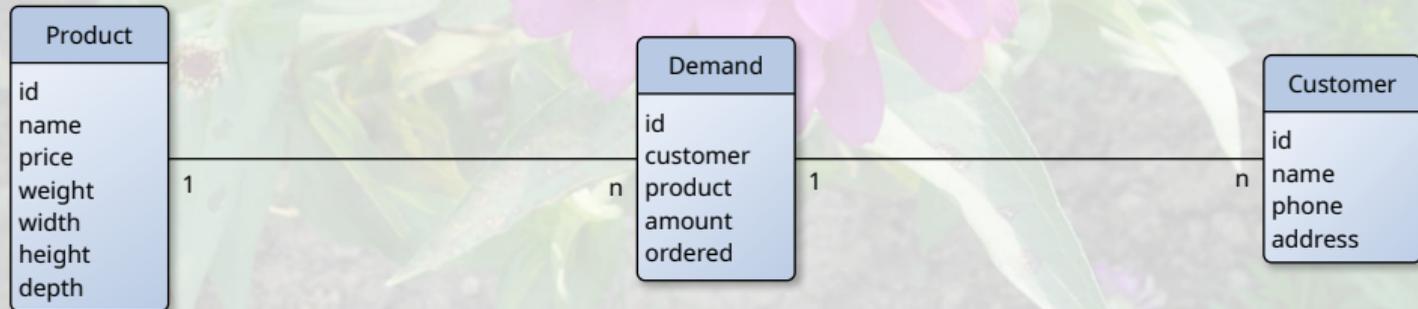
- Alle drei Objekttypen sind dargestellt.
- Die Linien, die sie verbinden, symbolisieren die Beziehungen.
- Hier sehen wir das ein Kunde (und Produkt) mit beliebig vielen Objekten des Typs *Demand* verbunden sein kann.
- Jedes solche Demand-Objekt ist mit genau einem Kunde (und Produkt) verbunden.
- Wir können die Struktur der Datenbank auf Basis so eines Designs erstellen.
- Das lernen wir später alles ganz genau.



Beispiel: Fabrikdatenbank



- Die Linien, die sie verbinden, symbolisieren die Beziehungen.
- Hier sehen wir das ein Kunde (und Produkt) mit beliebig vielen Objekten des Typs *Demand* verbunden sein kann.
- Jedes solche Demand-Objekt ist mit genau einem Kunde (und Produkt) verbunden.
- Wir können die Struktur der Datenbank auf Basis so eines Designs erstellen.
- Das lernen wir später alles ganz genau.
- In unserem Beispiel haben wir das nicht gemacht, weil wir schnell in die Welt der Datenbanken eintauchen wollten.



ERDs Reverse Engineeren



- Sagen wir, dass wir tatsächlich eine Datenbank zuerst als ERD entworfen und dann mit SQL implementiert haben.



ERDs Reverse Engineeren



- Sagen wir, dass wir tatsächlich eine Datenbank zuerst als ERD entworfen und dann mit SQL implementiert haben.
- Die Tabellen und ihre Fremdschlüssel entsprechen dann dem Modell, das als ERD gezeichnet wurde.

ERDs Reverse Engineeren



- Sagen wir, dass wir tatsächlich eine Datenbank zuerst als ERD entworfen und dann mit SQL implementiert haben.
- Die Tabellen und ihre Fremdschlüssel entsprechen dann dem Modell, das als ERD gezeichnet wurde.
- Die Information des ERD ist dann in der Datenbank präsent.

ERDs Reverse Engineeren



- Sagen wir, dass wir tatsächlich eine Datenbank zuerst als ERD entworfen und dann mit SQL implementiert haben.
- Die Tabellen und ihre Fremdschlüssel entsprechen dann dem Modell, das als ERD gezeichnet wurde.
- Die Information des ERD ist dann in der Datenbank präsent.
- Sie steckt in der Struktur der Tabellen und der Fremdschlüssel.

ERDs Reverse Engineeren



- Sagen wir, dass wir tatsächlich eine Datenbank zuerst als ERD entworfen und dann mit SQL implementiert haben.
- Die Tabellen und ihre Fremdschlüssel entsprechen dann dem Modell, das als ERD gezeichnet wurde.
- Die Information des ERD ist dann in der Datenbank präsent.
- Sie steckt in der Struktur der Tabellen und der Fremdschlüssel.
- Wenn das stimmt, dann können wir ein ERD zumindest teilweise auch aus einer Datenbank rekonstruieren.

ERDs Reverse Engineeren



- Sagen wir, dass wir tatsächlich eine Datenbank zuerst als ERD entworfen und dann mit SQL implementiert haben.
- Die Tabellen und ihre Fremdschlüssel entsprechen dann dem Modell, das als ERD gezeichnet wurde.
- Die Information des ERD ist dann in der Datenbank präsent.
- Sie steckt in der Struktur der Tabellen und der Fremdschlüssel.
- Wenn das stimmt, dann können wir ein ERD zumindest teilweise auch aus einer Datenbank rekonstruieren.
- Natürlich können wir die Bedeutung hinter den Objekten und deren Beziehungen nicht wieder „züückberechnen“.

ERDs Reverse Engineeren



- Sagen wir, dass wir tatsächlich eine Datenbank zuerst als ERD entworfen und dann mit SQL implementiert haben.
- Die Tabellen und ihre Fremdschlüssel entsprechen dann dem Modell, das als ERD gezeichnet wurde.
- Die Information des ERD ist dann in der Datenbank präsent.
- Sie steckt in der Struktur der Tabellen und der Fremdschlüssel.
- Wenn das stimmt, dann können wir ein ERD zumindest teilweise auch aus einer Datenbank rekonstruieren.
- Natürlich können wir die Bedeutung hinter den Objekten und deren Beziehungen nicht wieder „züückberechnen“.
- Aber wir können die Objekte und Beziehungen problemlos auf einer syntaktischen Ebene wiederherstellen.

ERDs Reverse Engineeren



- Sagen wir, dass wir tatsächlich eine Datenbank zuerst als ERD entworfen und dann mit SQL implementiert haben.
- Die Tabellen und ihre Fremdschlüssel entsprechen dann dem Modell, das als ERD gezeichnet wurde.
- Die Information des ERD ist dann in der Datenbank präsent.
- Sie steckt in der Struktur der Tabellen und der Fremdschlüssel.
- Wenn das stimmt, dann können wir ein ERD zumindest teilweise auch aus einer Datenbank rekonstruieren.
- Natürlich können wir die Bedeutung hinter den Objekten und deren Beziehungen nicht wieder „zrückberechnen“.
- Aber wir können die Objekte und Beziehungen problemlos auf einer syntaktischen Ebene wiederherstellen.
- Das können wir so schnell und automatisch machen, dass wir eigentlich immer mit einer ERD-basierten Ansicht unserer Datenbank arbeiten können.

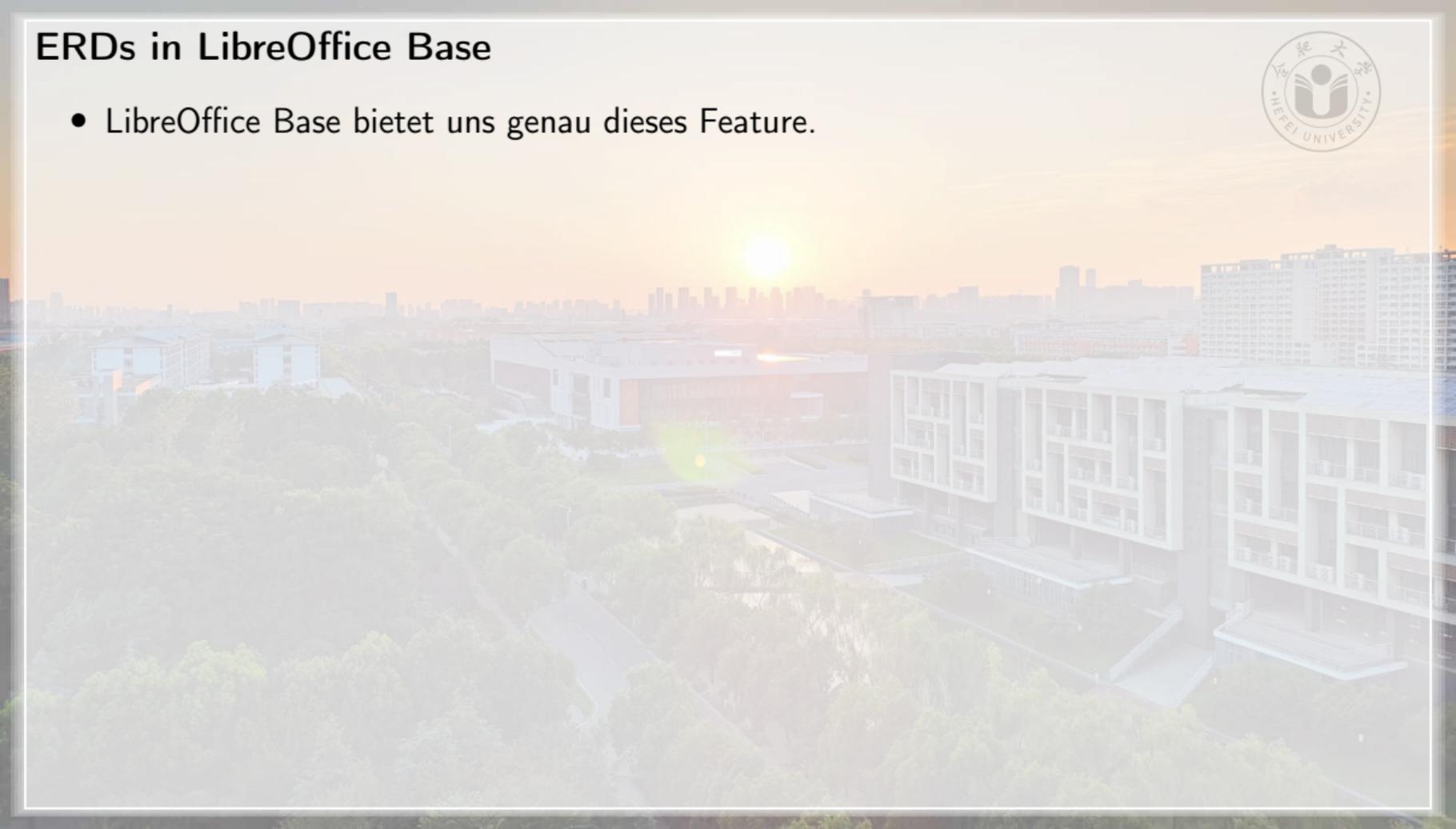


ERDs in LibreOffice Base



ERDs in LibreOffice Base

- LibreOffice Base bietet uns genau dieses Feature.



ERDs in LibreOffice Base

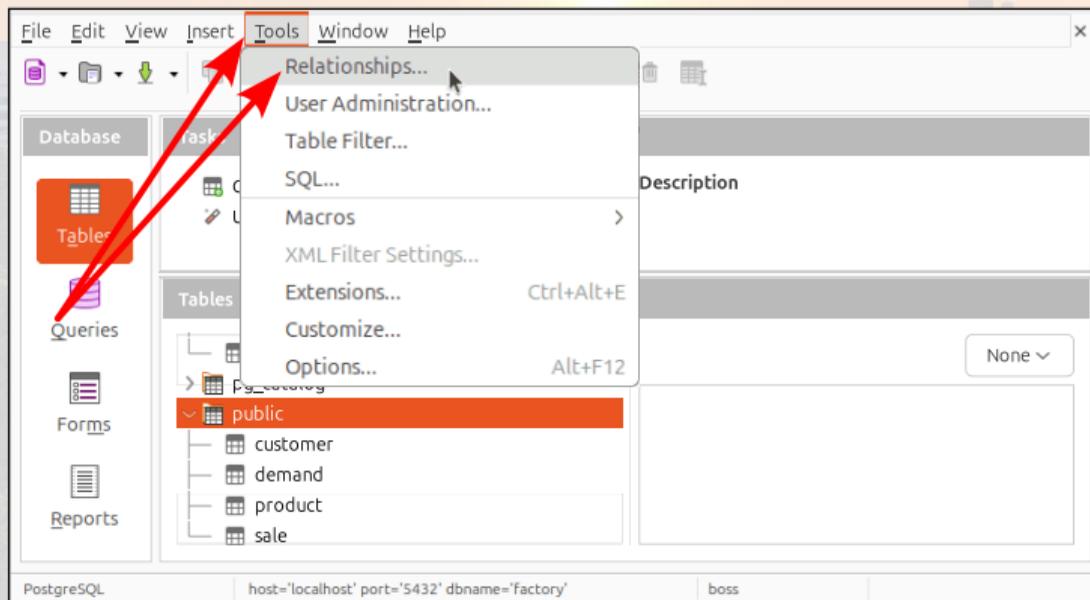


- LibreOffice Base bietet uns genau dieses Feature.
- Wir öffnen also unser Dokument `factory.odt` aus der vorigen Einheit mit LibreOffice Base und verbinden uns auf unsere Datenbank, wobei wir das Passwort `superboss123` eingeben müssen.

ERDs in LibreOffice Base



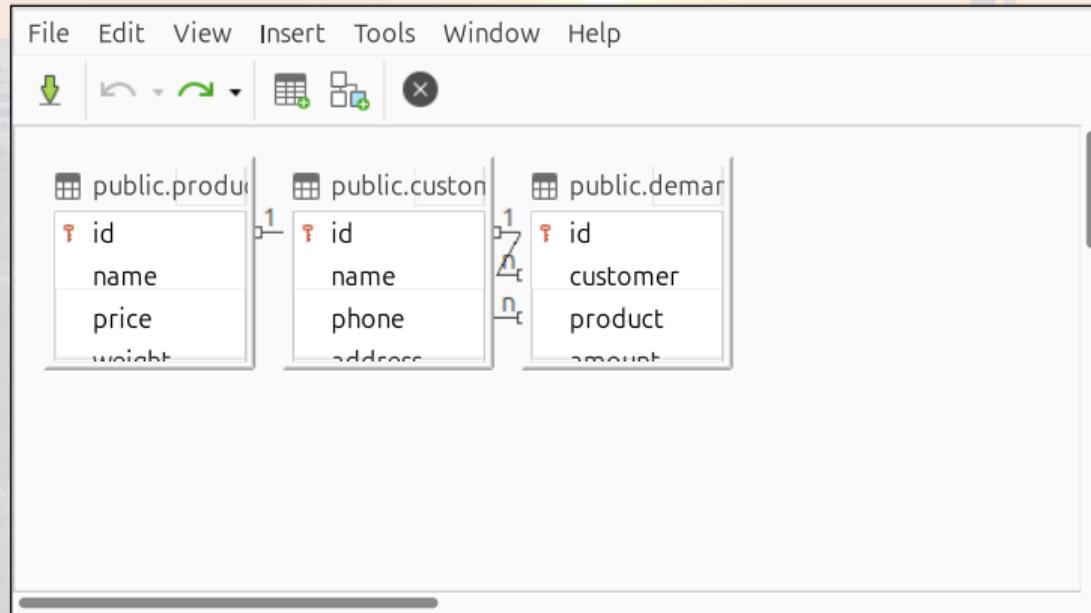
- Wir öffnen das **Tools**-Menü und klicken auf **Relationships**.



ERDs in LibreOffice Base



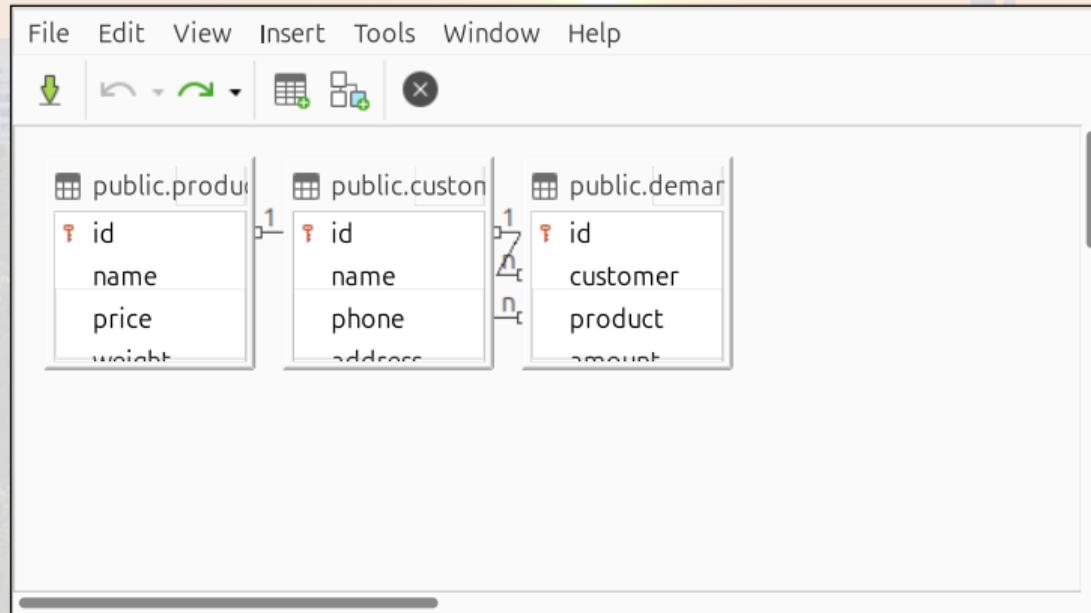
- Ein sehr überfrachtetes Diagramm öffnet sich.



ERDs in LibreOffice Base



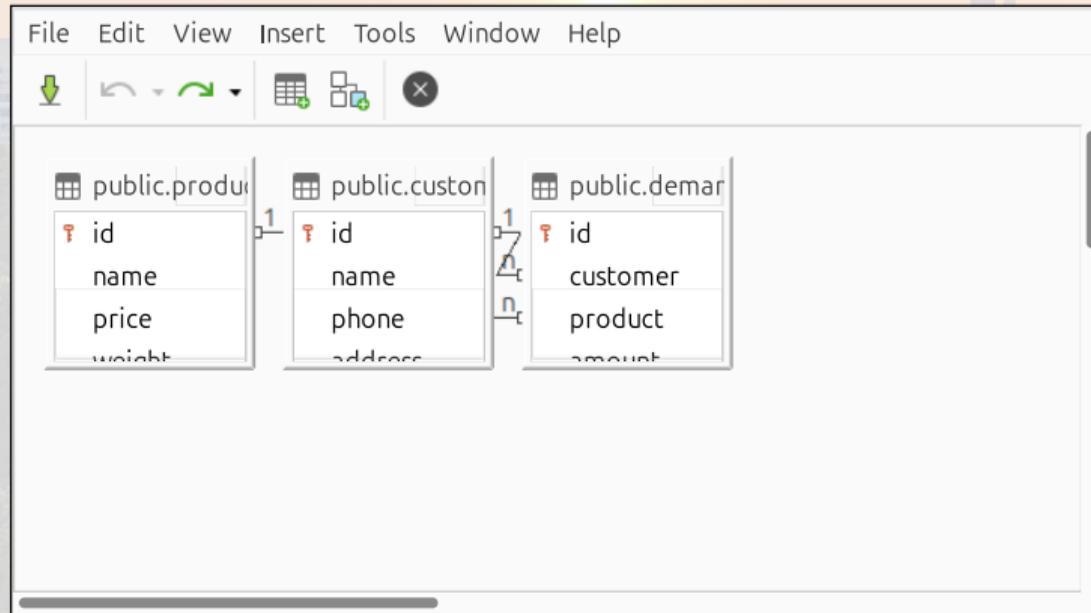
- Ein sehr überfrachtetes Diagramm öffnet sich.
- Wir sehen alle drei Tabellen in unserer Datenbank, aber sie sind nicht schön arrangiert.



ERDs in LibreOffice Base



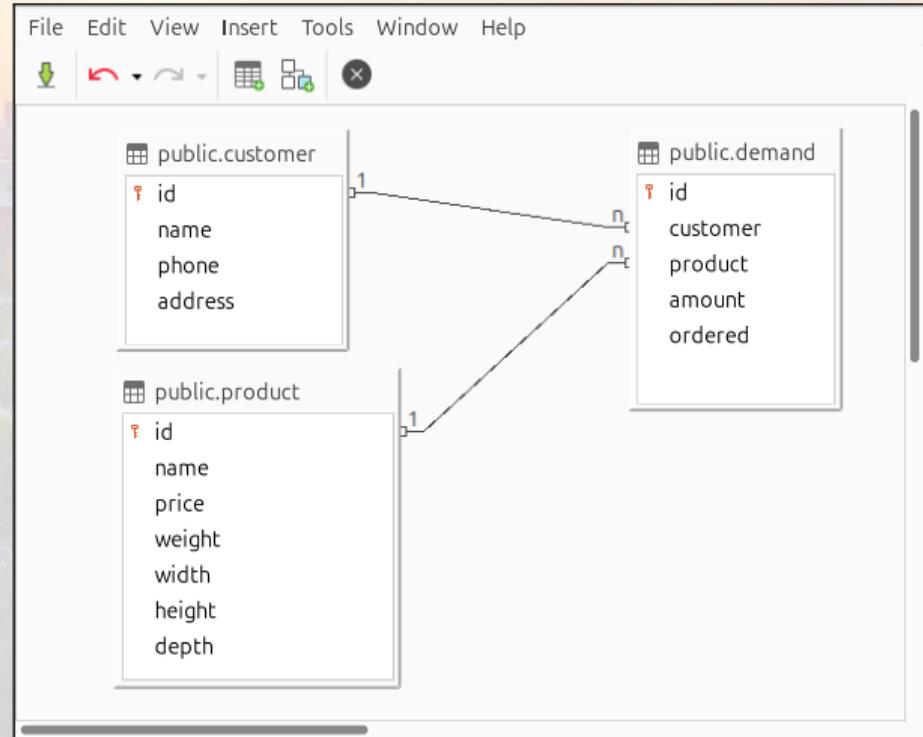
- Ein sehr überfrachtetes Diagramm öffnet sich.
- Wir sehen alle drei Tabellen in unserer Datenbank, aber sie sind nicht schön arrangiert.
- Wir können sie aber anklicken und herumziehen.



ERDs in LibreOffice Base



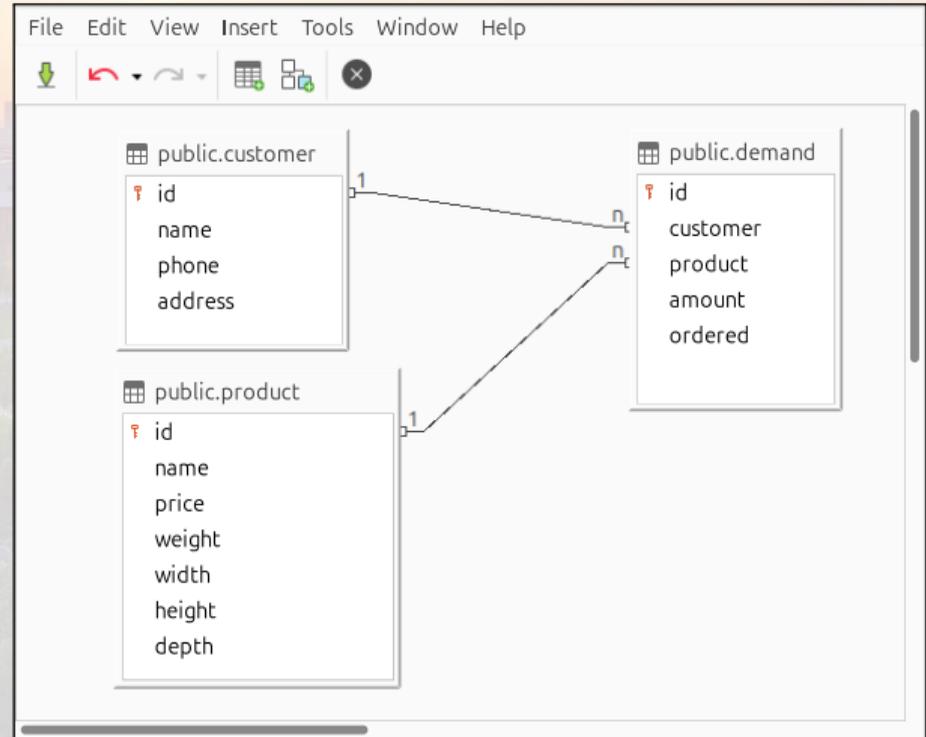
- Nachdem wir sie neu arrangiert haben, bekommen wir eine sehr schöne Übersicht über unsere Datenbank.



ERDs in LibreOffice Base



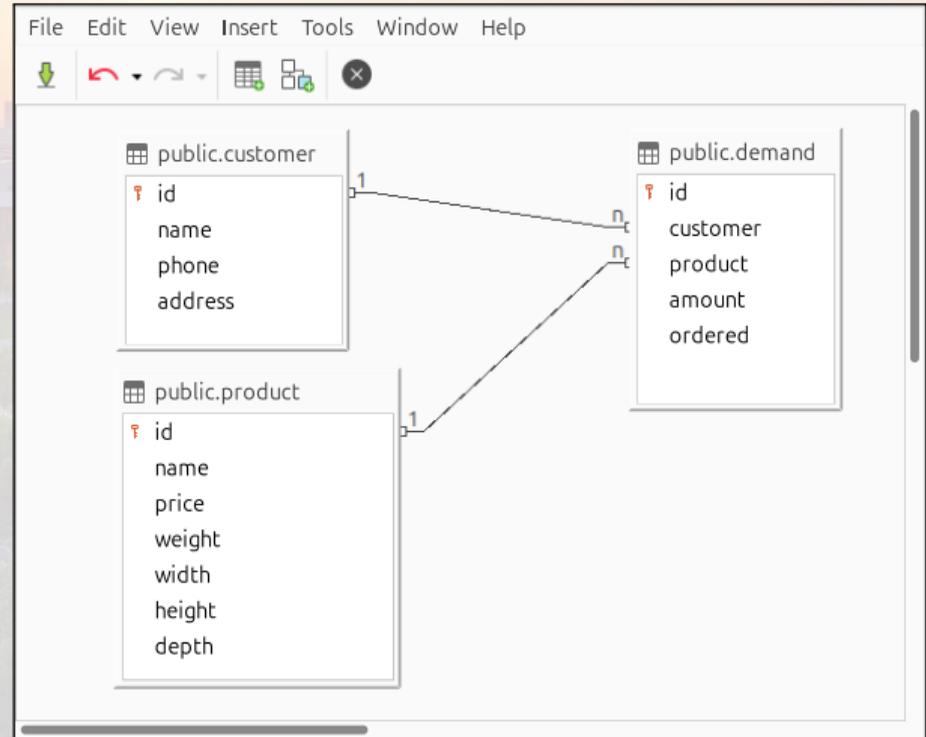
- Wir sehen z. B., dass die `id`-Spalten die Primärschlüssel der Tabellen sind, denn sie sind mit dem Schlüssel-Icon  markiert.



ERDs in LibreOffice Base



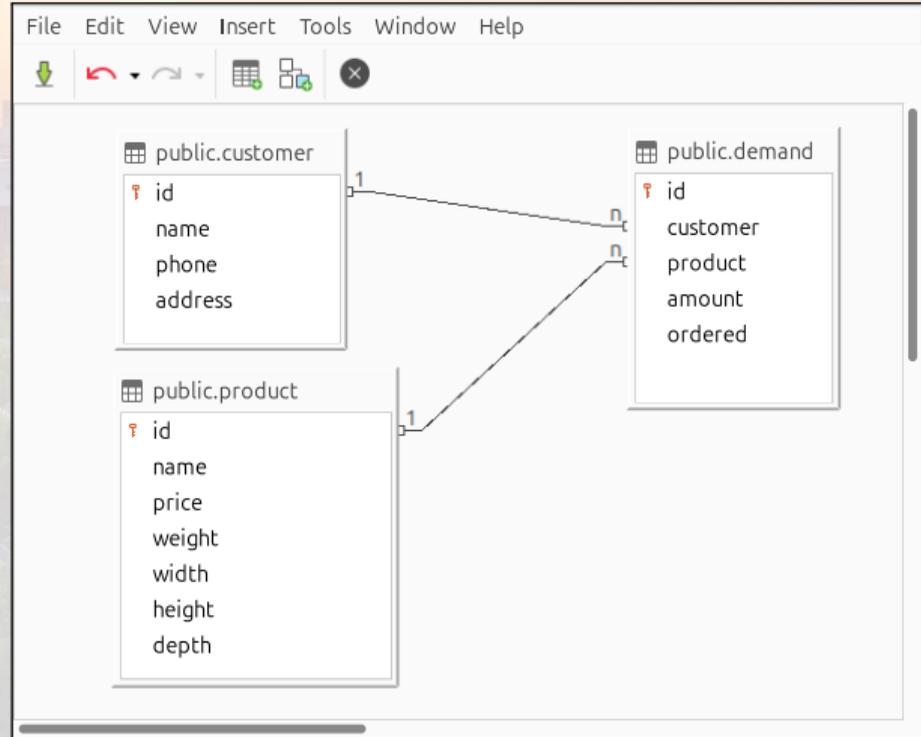
- Wir sehen, dass jeder `id`-Wert in Tabelle `customer` in n Datensätzen als `customer`-Wert in der Tabelle `demand` verwendet werden kann.



ERDs in LibreOffice Base



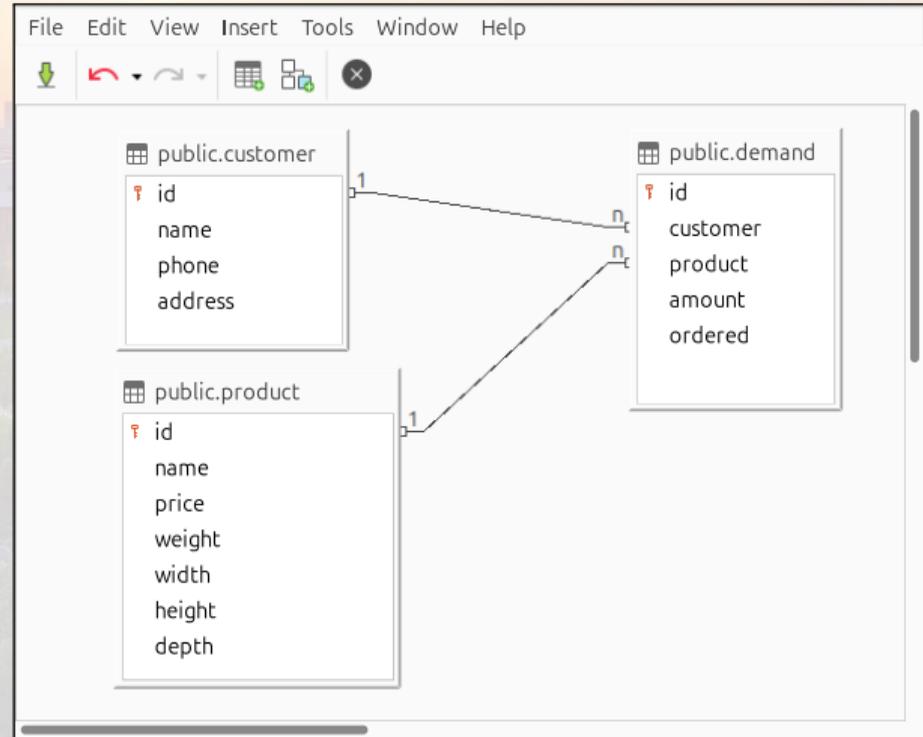
- Das selbe gilt für die `id`-Werte in der Tabelle `product`, die in n Datensätzen der Tabelle `demand` als Wert der Spalte `product` stehen kann.



ERDs in LibreOffice Base



- Dabei steht n für „beliebig oft“.





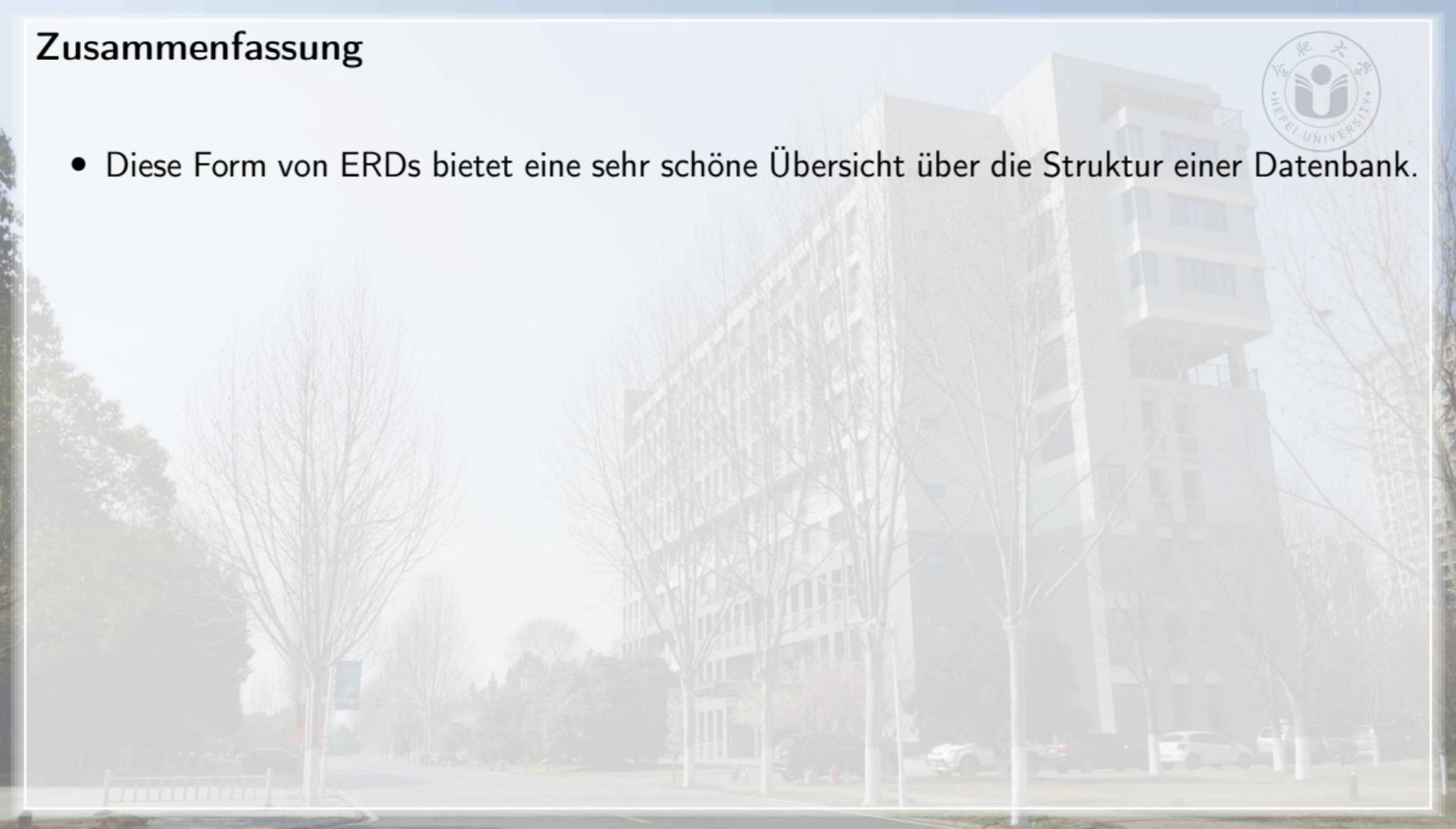
Zusammenfassung



Zusammenfassung



- Diese Form von ERDs bietet eine sehr schöne Übersicht über die Struktur einer Datenbank.



Zusammenfassung



- Diese Form von ERDs bietet eine sehr schöne Übersicht über die Struktur einer Datenbank.
- Solche Diagramme können automatisch von der LibreOffice Base GUI für uns erstellt werden.

Zusammenfassung



- Diese Form von ERDs bietet eine sehr schöne Übersicht über die Struktur einer Datenbank.
- Solche Diagramme können automatisch von der LibreOffice Base GUI für uns erstellt werden.
- Wenn unsere DB komplexer ist, mit vielen Tabellen und Beziehungen, dann kann so eine Illustration schon sehr hilfreich sein.

Zusammenfassung



- Diese Form von ERDs bietet eine sehr schöne Übersicht über die Struktur einer Datenbank.
- Solche Diagramme können automatisch von der LibreOffice Base GUI für uns erstellt werden.
- Wenn unsere DB komplexer ist, mit vielen Tabellen und Beziehungen, dann kann so eine Illustration schon sehr hilfreich sein.
- Stellen Sie sich vor, dass Sie neu in ein Team einer Organisation als DBA eintreten.

Zusammenfassung



- Diese Form von ERDs bietet eine sehr schöne Übersicht über die Struktur einer Datenbank.
- Solche Diagramme können automatisch von der LibreOffice Base GUI für uns erstellt werden.
- Wenn unsere DB komplexer ist, mit vielen Tabellen und Beziehungen, dann kann so eine Illustration schon sehr hilfreich sein.
- Stellen Sie sich vor, dass Sie neu in ein Team einer Organisation als DBA eintreten.
- Stellen Sie sich vor, dass sie an einer existierenden Datenbank arbeiten sollen.

Zusammenfassung



- Diese Form von ERDs bietet eine sehr schöne Übersicht über die Struktur einer Datenbank.
- Solche Diagramme können automatisch von der LibreOffice Base GUI für uns erstellt werden.
- Wenn unsere DB komplexer ist, mit vielen Tabellen und Beziehungen, dann kann so eine Illustration schon sehr hilfreich sein.
- Stellen Sie sich vor, dass Sie neu in ein Team einer Organisation als DBA eintreten.
- Stellen Sie sich vor, dass sie an einer existierenden Datenbank arbeiten sollen.
- Leider gibt es keine Dokumentation.

Zusammenfassung



- Diese Form von ERDs bietet eine sehr schöne Übersicht über die Struktur einer Datenbank.
- Solche Diagramme können automatisch von der LibreOffice Base GUI für uns erstellt werden.
- Wenn unsere DB komplexer ist, mit vielen Tabellen und Beziehungen, dann kann so eine Illustration schon sehr hilfreich sein.
- Stellen Sie sich vor, dass Sie neu in ein Team einer Organisation als DBA eintreten.
- Stellen Sie sich vor, dass sie an einer existierenden Datenbank arbeiten sollen.
- Leider gibt es keine Dokumentation.
- Sie können sich nun auf das DBMS verbinden und versuchen, die Struktur der Datenbank mit SQL herauszufinden.

Zusammenfassung



- Diese Form von ERDs bietet eine sehr schöne Übersicht über die Struktur einer Datenbank.
- Solche Diagramme können automatisch von der LibreOffice Base GUI für uns erstellt werden.
- Wenn unsere DB komplexer ist, mit vielen Tabellen und Beziehungen, dann kann so eine Illustration schon sehr hilfreich sein.
- Stellen Sie sich vor, dass Sie neu in ein Team einer Organisation als DBA eintreten.
- Stellen Sie sich vor, dass sie an einer existierenden Datenbank arbeiten sollen.
- Leider gibt es keine Dokumentation.
- Sie können sich nun auf das DBMS verbinden und versuchen, die Struktur der Datenbank mit SQL herauszufinden.
- Oder Sie verbinden sich mit LibreOffice Base und bekommen eine schnelle und umfangreiche Übersicht über die Datenbank, über die Tabellen und deren Beziehungen...



谢谢您们！
Thank you!
Vielen Dank!



Glossary (in English) I



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*³⁹.

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 database (DB). The DBMS allows the user/application to create, read, write, update, delete, and otherwise manipulate the data in the DB⁴².

ERD Entity relationship diagrams show the relationships between objects, e.g., between the tables in a DB and how they reference each other^{1,4,6-8,22,31,40}

GUI graphical user interface

LibreOffice is an open source office suite^{17,24,30} which is a good and free alternative to Microsoft Office. It offers software such as LibreOffice Writer, LibreOffice Calc, and LibreOffice Base. See³⁹ for more information and installation instructions.

LibreOffice Base is a database management system (DBMS) that can work on stand-alone files but also connect to other popular relational databases^{15,30}. It is part of LibreOffice^{17,24,30} and has functionality that is comparable to Microsoft Access^{2,9,38}.

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^{24,30}, i.e., an alternative to Microsoft Excel. It is part of LibreOffice^{17,24,30}.

LibreOffice Writer is a free and open source text writing program⁴³ and part of LibreOffice^{17,24,30}. It is a good alternative to Microsoft Word.

Microsoft Access is a DBMS that can work on DBs stored in single, stand-alone files but also connect to other popular relational databases^{2,9,25,38}. It is part of Microsoft Office. A free and open source alternative to this commercial software is LibreOffice Base.

Glossary (in English) II



Microsoft Excel is a spreadsheet program that allows users to store, organize, manipulate, and calculate data in tabular structures^{3,18,23}. It is part of Microsoft Office. A free alternative to this commercial software is LibreOffice Calc^{24,30}.

Microsoft Office is a commercial suite of office software, including Microsoft Excel, Microsoft Word, and Microsoft Access²³. LibreOffice is a free and open source alternative.

Microsoft Word is one of the leading text writing programs^{14,27,43} and part of Microsoft Office. A free alternative to this commercial software is the LibreOffice Writer.

PostgreSQL An open source object-relational DBMS^{16,28,29,37}. See <https://postgresql.org> for more information.

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^{10,19,20,32,36,39,41}.

SQL The *Structured Query Language* is basically a programming language for querying and manipulating relational databases^{5,11-13,21,26,33-36}. It is understood by many DBMSes. You find the Structured Query Language (SQL) commands supported by PostgreSQL in the reference³³.