

Datenbanken Grundlagen

01 Datenbank – Definition

Eine Datenbank ist eine Sammlung von Daten, die in einem strukturierten Format organisiert und gespeichert sind, so dass sie leicht zugänglich, verwaltbar und aktualisierbar sind.

Eine Datenbank kann auf einem Computer oder einem Server gehostet werden und kann aus verschiedenen Tabellen bestehen, die miteinander in Beziehung stehen. Datenbanken können verwendet werden, um große Datenmengen zu speichern und zu verwalten, so dass sie schnell und effizient abgerufen werden können.

Datenbanken werden in vielen Bereichen eingesetzt, darunter Unternehmen, Regierungen, Gesundheitswesen, Bildung, Forschung und vielen anderen.

02 Datenbank – Einfluss auf unser Leben

Wo werden überall Daten gesammelt und genutzt?

- Social Media: persönliche Daten, Interaktionen, Vorlieben und Verhaltensweisen
- Transport: Routen, Geschwindigkeit und Abfahrtszeiten
- Banken: Finanzen und unser Kaufverhalten
- ...

Wie beeinflusst dies unser tägliches Leben?

- personalisierte Werbung und Produktempfehlungen
- leichter online einkaufen und Informationen finden
- Angreifbar für Datenmissbrauch (Identitätsdiebstahl, Überwachung, Phishing, ...)
- Verbesserung der Taktung von öffentlichen Verkehrsmitteln
- ...

Orte, wo Daten gesammelt werden

- Beispiele: Google, Amazon, Youtube, TikTok, ..., Kauf von Fahrkarte
- Die gesammelten Daten werden auch BIG DATA genannt

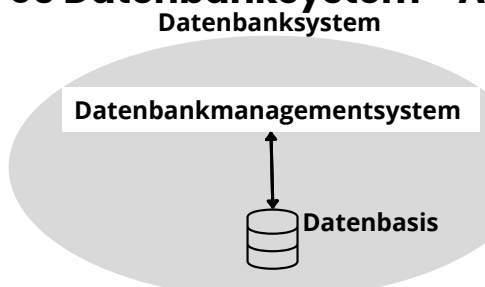
Big Data - Die 3-V's

Variety - eine hohe Vielfalt an Daten

Velocity - eine hohe Geschwindigkeit, in der Daten generiert werden

Volume - ein hohes Volumen an Daten, das aufkommt

03 Datenbanksystem – Aufbau, Bestandteile



Ein **Datenbanksystem (DBS)** besteht aus einer systematischen und strukturierten Sammlung von Daten (**Datenbasis=DB**) sowie einer Software zur Verwaltung dieser Daten (**Datenbankmanagementsystem = DBMS**)

**Datenbanksystem =
Datenbasis + Datenbankmanagementsystem**

Datenbankmanagementsystem

- Zentrale Speicherung und einheitliche Verwaltung von Daten eines Problemereichs
- Dienstleistung des Datenzugriffs unter Verwendung von Datenbanksprachen (z.B. SQL)
- Mechanismen der Datensicherheit (Kontrolle der Legalität des Zugriffs auf die Datenbasis, Schutz vor Bedienfehlern, Organisation des Zugriffs durch mehrere Nutzer, ...)

Zusammenfassend:

Ein DBMS ist eine Schnittstelle zwischen Datenbasis und Benutzer

04 Datenbankmodelle

Datenbanken können unterschiedlich aufgebaut sein, je nach Anforderungen und Anwendungsbereiche. In der folgenden Tabelle sind einige bekannte Datenbankmodelle, deren Aufbau und Anwendungsgebiete

Datenbankmodell	Aufbau	Anwendungsgebiete
Hierarchisches Datenbankmodell	Besteht aus verschiedenen Ebenen, in denen Daten aufeinander aufbauen und übergeordnete Ebenen von untergeordneten Ebenen getrennt sind.	Wird häufig in der Industrie eingesetzt, insbesondere in der Fertigung, um Materialflüsse und Produktionsprozesse zu steuern.
Graphen Datenbankmodell	Besteht aus Knoten und Kanten, die miteinander verbunden sind, um komplexe Beziehungen darzustellen.	Wird häufig in sozialen Netzwerken, wissenschaftlichen Forschungseinrichtungen und in der Logistik eingesetzt, um komplexe Beziehungen zwischen verschiedenen Elementen zu modellieren.
Objektorientiertes Datenbankmodell	Daten werden in Form von Objekten gespeichert, die Eigenschaften und Methoden haben und miteinander verknüpft werden können.	Wird häufig in der Medizin und der Geologie eingesetzt, um komplexe Datenstrukturen wie den menschlichen Körper oder geologische Formationen zu modellieren.
Relationales Datenbankmodell	Daten werden in Tabellen organisiert, die miteinander verbunden sind, um komplexe Datenbeziehungen darzustellen.	Wird in fast allen Branchen eingesetzt, insbesondere in der Finanzbranche, im Einzelhandel und in der Verwaltung, um große Mengen strukturierter Daten effizient zu verwalten und abzurufen.

05 Relationale Datenbank

Relationale Datenbanken sind eine besonders häufige Datenbankart. Eine relationale Datenbank ist eine Art von Datenbank, die auf dem relationalen Datenbankmodell basiert. In einer relationalen Datenbank werden Daten in Tabellen organisiert und in Beziehung zueinander gesetzt. Jede Tabelle besteht aus einer Reihe von Spalten, die bestimmte Datentypen repräsentieren, und einer Menge von Zeilen, die konkrete Daten enthalten. Beziehungen zwischen Tabellen werden durch Verknüpfung von Primärschlüsseln und Fremdschlüsseln hergestellt.

Jede Tabelle wird durch ein Schema charakterisiert. Ein **Datenbankschema** enthält:

- Name der Tabelle
- Liste der Attribute (Spaltenname, Name der Felder)
- Typen der Attributwerte (Felddatentypen, z.B. Text, Zahl, ...)

Formale Darstellung: Tabellename(Attribut_1: Typ_1, ..., Attribut_n: Typ_n)

Jede Tabelle hat ein Attribut, das einen Datensatz eindeutig identifizieren lässt. Dieses Attribut wird **PRIMÄRSCHLÜSSEL** genannt.

Tabellename (USER)

Attribute mit Attributnamen (id, username, name, gender)

Datensatz / (Daten-) Objekt (1, 2, 101, ...)

Attributwerte (niclas258, rafael54, celine96, ...)

Dient als Primärschlüssel zur eindeutigen Identifizierung eines Datensatzes (id)

id	username	name	gender
1	niclas258	Niclas Schweizer	male
2	rafael54	Rafael Probst	male
101	celine96	Celine Fisher	female
...

Häufige Datentypen für Attributwerte:

- Ganzzahlige Werte (z.B. Alter, Anzahl, ID): INTEGER
- Fließkommazahlen (z.B. Preise, Bewertungen): DECIMAL oder FLOAT
- Zeichenketten (z.B. Name, Adresse, Titel): VARCHAR
- Datumswerte (z.B. Geburtsdatum, Erstellungsdatum): DATE oder DATETIME
- Wahrheitswerte (z.B. aktiv/inaktiv, ja/nein): BOOLEAN

06 Relationale Datenbank – Primärschlüssel

Primärschlüssel erlauben die eindeutige Identifikation einzelner Datensätze.

Es werden natürliche und künstliche Primärschlüssel unterschieden.

Natürlicher Primärschlüssel:

ein Attribut in einer Tabelle, das von Natur aus eindeutig ist und sich gut als Primärschlüssel eignet. Ein Beispiel dafür ist die Sozialversicherungsnummer oder die E-Mail-Adresse.

Künstlicher Primärschlüssel:

wird vom Datenbankmanagementsystem automatisch generiert und hat keine Bedeutung außerhalb der Datenbank. Ein Beispiel hierfür ist eine fortlaufende Nummer oder ein zufällig generierter Code.

06 Relationale Datenbank – Entwicklungsschritte

Anforderungsanalyse

In dieser Phase werden die Anforderungen an die Datenbank ermittelt, um den Anwendungsbereich und den Zweck der Datenbank zu verstehen.

Konzeptuelle Modellierung

In dieser Phase wird ein Entity-Relationship-Modell (ER-Modell) erstellt, um die Beziehungen zwischen den verschiedenen Entitäten zu erfassen.

Logische Modellierung

In dieser Phase wird das konzeptuelle Modell in ein logisches Modell überführt, welches die Struktur der Datenbank auf der Ebene der Tabellen und Beziehungen widerspiegelt. Dabei wird das Modell auf Normalformen überprüft und gegebenenfalls optimiert. - **Datenbankschema**

Implementierung

In dieser Phase wird die Datenbank auf der Grundlage des logischen Modells tatsächlich erstellt.

Testen und Feinabstimmung

In dieser Phase wird die Datenbank getestet, um sicherzustellen, dass sie den Anforderungen entspricht. Außerdem können in dieser Phase Feinabstimmungen vorgenommen werden, um die Performance zu optimieren.

Betrieb und Wartung

In dieser Phase wird die Datenbank in den produktiven Einsatz überführt und muss regelmäßig gewartet werden, um sicherzustellen, dass sie zuverlässig und sicher arbeitet.

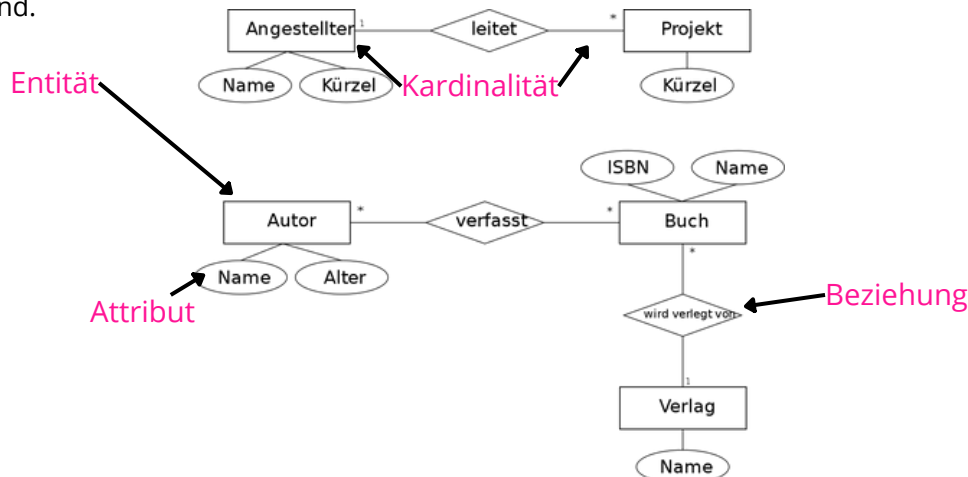
ANMERKUNG: Ablauf kann stark abweichen und variieren. Häufig sind mehrere Iterationen (Schleifen/Wiederholungen) notwendig

07 Entity Relationship Modell – Definition

- eine grafische Darstellung, um die Beziehungen zwischen Entitäten (Objekten, Personen, Orten, Konzepten usw.) in einer Datenbank zu modellieren.
- genutzt, um die Struktur der Datenbank zu planen und zu definieren, indem es Entitäten, Attribute und Beziehungen zwischen Entitäten darstellt.
- Das ER-Modell bietet eine visuelle Darstellung der Datenbank und
- ER-Modelle erleichtern die Kommunikation zwischen Datenbankentwicklern und anderen Interessengruppen, um sicherzustellen, dass alle Anforderungen berücksichtigt werden.
- Wird für die konzeptuelle Modellierung verwendet

08 Entity Relationship Modell – Bestandteile

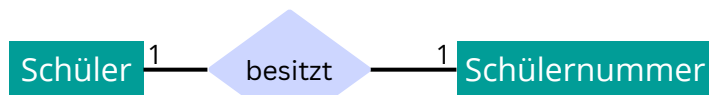
- **Entitäten** (Objekte, Dinge) werden als Rechtecke mit dem Entitätsnamen im Inneren dargestellt.
- **Attribute** (Eigenschaften von Entitäten) werden als Ovale oder Ellipsen dargestellt, die mit den Entitäten verbunden sind.
- **Beziehungen** zwischen Entitäten werden als Linien mit einer Bezeichnung (z.B. "arbeitet bei") dargestellt.
- **Kardinalitäten** (Anzahl der Beziehungen zwischen Entitäten) werden als Striche oder Pfeile an den Beziehungslinien dargestellt, die angeben, wie viele Entitäten auf jeder Seite der Beziehung beteiligt sind.



09 Entity Relationship Modell – Kardinalitäten

1:1: Jedes Element auf der einen Seite der Beziehung ist mit höchstens einem Element auf der anderen Seite der Beziehung verknüpft.

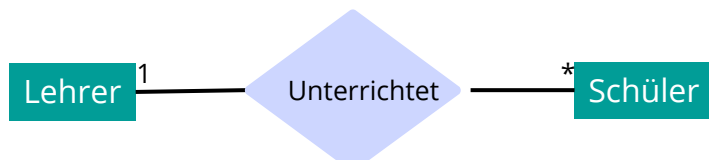
Beispiel: Jede:r einzelne:r SchülerIn besitzt einer SchülerInnen-Nummer



1:N: Ein Element auf der einen Seite der Beziehung ist mit beliebig vielen Elementen auf der anderen Seite der Beziehung verknüpft, aber jedes Element auf der anderen Seite kann nur mit einem Element auf der einen Seite verknüpft sein.

Beispiel:

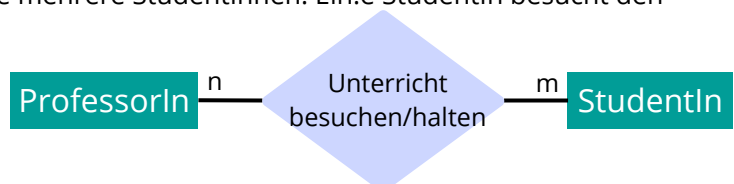
ein Lehrer unterrichtet mehrere Schüler, aber jeder Schüler nur von einem Lehrer unterrichtet wird (für ein Fach)



N:M: Ein Element auf der einen Seite der Beziehung kann mit beliebig vielen Elementen auf der anderen Seite verknüpft sein, und umgekehrt. In diesem Fall ist eine Zwischentabelle erforderlich, um die Beziehung zu modellieren.

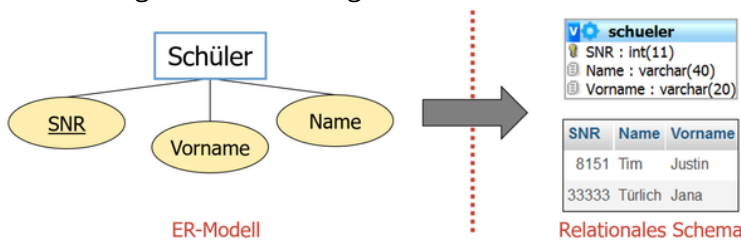
Beispiel:

Ein:e ProfessorIn unterrichtet üblicherweise mehrere StudentInnen. Ein:e StudentIn besucht den Unterricht von mehreren ProfessorInnen.



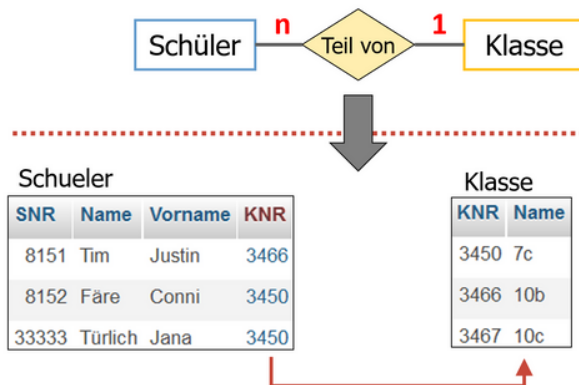
10 Logische Modellierung – ER Modell zum relationalen Modell

Die Überführung eines Entity-Relationship-Modells in das Relationen-Modell basiert im Wesentlichen auf den folgenden Abbildungen:



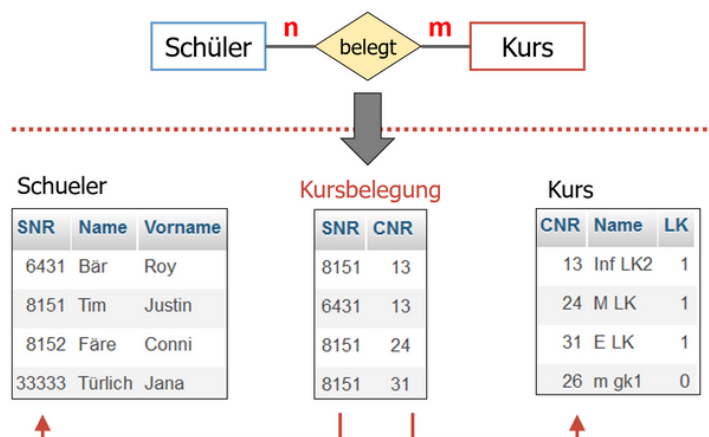
Entitätstyp

für jeden Entitätstyp entsteht eine Tabelle/Relation. Dabei muss (sofern noch nicht vorhanden) ein Primärschlüssel und der Datentyp aller Attribute festgelegt werden.



1:n Beziehungstypen

Für einen 1:n-Beziehungstyp wird ein Fremdschlüssel auf Seite des "abhängigen" (man sagt auch "Kind")-Entitätstyps hinzugefügt (dort, wo im Diagramm das "n" steht). Als Fremdschlüssel wird in der Regel der Primärschlüssel des anderen Entitätstyps verwendet.



n:m Beziehungstypen

Für jede n:m-Beziehung wird eine zusätzliche Tabelle angelegt, die die Fremdschlüssel auf die beteiligten Tabellen enthält (sowie evtl. vorhandene Attribute des Beziehungstyps selbst).

11 Normalformen – Definition

Normalformen sind Regeln, die angewendet werden, um sicherzustellen, dass eine relationale Datenbank gut strukturiert und effizient ist. Sie dienen dazu, Redundanz und Inkonsistenzen in den Daten zu vermeiden, um die Datenkonsistenz zu gewährleisten. Es gibt verschiedene Normalformen, wie die erste Normalform (1NF), die zweite Normalform (2NF), die dritte Normalform (3NF) und höhere Normalformen wie die Boyce-Codd Normalform (BCNF) oder die vierte Normalform (4NF). Jede Normalform hat ihre eigenen spezifischen Anforderungen, die erfüllt werden müssen, um sicherzustellen, dass die Datenbank in dieser Normalform vorliegt.

Die Normalformen werden versucht bei der logischen Modellierung zu erreichen. Im Idealfall sollte die dritte Normalform erreicht werden, da sie Redundanzen weitestgehend eliminiert und dadurch eine effiziente und konsistente Datenhaltung gewährleistet.

12 Normalformen – Übersicht

Normalform	Kennzeichen
1. Normalform 1NF	Ein Attribut darf nur atomare (unzerlegbare) Werte enthalten. Das bedeutet, dass jedes Attribut nur einen einzelnen Wert pro Datensatz enthalten darf und nicht in mehrere Bestandteile aufgeteilt werden kann. Zum Beispiel sollte eine Telefonnummer als ein einziges Attribut gespeichert werden, statt als separate Attribute für Vorwahl, Rufnummer, etc.
2. Normalform 2.NF	Ein Datensatz darf nur von einem Primärschlüssel abhängig sein. Das bedeutet, dass jedes Nichtschlüsselattribut in einem Datensatz nur vom gesamten Primärschlüssel abhängig sein darf und nicht nur von einem Teil des Primärschlüssels. Zum Beispiel sollten bei einer Bestelldatenbank das Lieferdatum und der Lieferstatus nicht nur vom Bestelldatum abhängig sein, sondern auch vom Kunden, der die Bestellung aufgegeben hat.
3. Normalform 3NF	Ein Nichtschlüsselattribut darf nicht von einem anderen Nichtschlüsselattribut abhängig sein. Das bedeutet, dass es keine transitive Abhängigkeiten zwischen Nichtschlüsselattributen geben darf. Zum Beispiel sollte in einer Mitarbeiterdatenbank das Gehalt nicht vom Abteilungsleiter abhängig sein, sondern nur von der Abteilung, in der der Mitarbeiter arbeitet.

12 Normalformen – Beispiel

Ausgangstabelle

Datum	Name	Straße	Ort	Artikel	Anzahl
01.01.2012	Max Mustermann	Musterstr. 1	12345 Musterort	Bleistift	5

1.NF

R.-Nr.	Datum	Name	Vorname	Straße	Hnr.	PLZ	Ort	Artikel	Anzahl	Preis	Währung
187	01.01.2012	Mustermann	Max	Musterstr.	1	12345	Musterort	Bleistift	5	1,00	Euro

2.NF

Rechnung		
R.-Nr.	Datum	Knr.
187	01.01.2012	007

Kunde						
Knr.	Name	Vorname	Straße	Hnr.	PLZ	Ort
007	Mustermann	Max	Musterstr.	1	12345	Musterort

Rechnungsposition			
R.-P.-Nr.	R.-Nr.	Art.-Nr.	Anzahl
1	187	69	5

Artikel		
Art.-Nr.	Artikel	Preis
69	Bleistift	1,00

↓ 3.NF

Kunde					
Knr.	Name	Vorname	Straße	Hnr.	PLZ
007	Mustermann	Max	Musterstr.	1	12345

Postleitzahl	
PLZ	Ort
12345	Musterort