

Sommersemester 2026

Datenmanagement & -analyse

Prof. Dr. Christoph M. Flath

Data Driven Decisions Group, Universität Würzburg

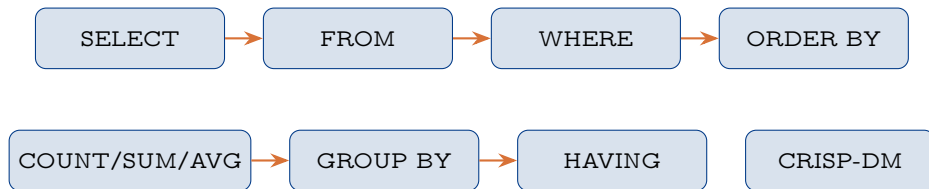
- 1 Rückblick & Motivation
- 2 Die “Mega-Tabelle” – Ein Beispiel
- 3 Anomalien: Was schief gehen kann
- 4 Anomalien selbst erleben
- 5 Die Lösung: Daten aufteilen
- 6 Erste Schritte zur Modellierung
- 7 Zusammenfassung & Ausblick

Lernziele

Probleme redundanter Datenhaltung erkennen und die Notwendigkeit von Datenmodellierung verstehen.

► 1 Rückblick & Motivation

- 2 Die “Mega-Tabelle” – Ein Beispiel
- 3 Anomalien: Was schief gehen kann
- 4 Anomalien selbst erleben
- 5 Die Lösung: Daten aufteilen
- 6 Erste Schritte zur Modellierung
- 7 Zusammenfassung & Ausblick



Bisher gearbeitet mit:

- Bundesliga-Tabelle (eine Tabelle)
- Spieler-Daten (eine Tabelle)
- Shipman/Benford-Daten (eine Tabelle pro Analyse)

Heute: Warum reicht eine Tabelle oft nicht aus?

Szenario

Sie möchten eine **Musiksammlung** verwalten:

- Welche Alben haben Sie?
- Welcher Künstler hat welches Album?
- In welchem Jahr erschien es?
- Welches Genre?

Naheliegender Ansatz: Alles in eine Tabelle!

Was kann da schon schiefgehen?

Album	Künstler	Künstler_Land	Jahr	Genre
Abbey Road	The Beatles	UK	1969	Rock
Let It Be	The Beatles	UK	1970	Rock
Thriller	Michael Jackson	USA	1982	Pop
Bad	Michael Jackson	USA	1987	Pop
The Dark Side of the Moon	Pink Floyd	UK	1973	Rock
Wish You Were Here	Pink Floyd	UK	1975	Rock

Sieht doch gut aus, oder?

Album	Künstler	Künstler_Land	Jahr	Genre
Abbey Road	The Beatles	UK	1969	Rock
Let It Be	The Beatles	UK	1970	Rock
Thriller	Michael Jackson	USA	1982	Pop
Bad	Michael Jackson	USA	1987	Pop
The Dark Side of the Moon	Pink Floyd	UK	1973	Rock
Wish You Were Here	Pink Floyd	UK	1975	Rock

Sieht doch gut aus, oder?

Schau genauer hin!

- “The Beatles” und “UK” stehen **zweimal**
- “Michael Jackson” und “USA” stehen **zweimal**
- “Pink Floyd” und “UK” stehen **zweimal**

⇒ **Redundanz!**

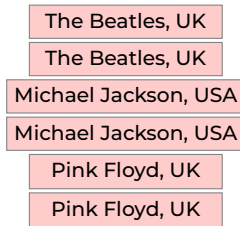
- 1 Rückblick & Motivation
- ▶ **2 Die “Mega-Tabelle” – Ein Beispiel**
- 3 Anomalien: Was schief gehen kann
- 4 Anomalien selbst erleben
- 5 Die Lösung: Daten aufteilen
- 6 Erste Schritte zur Modellierung
- 7 Zusammenfassung & Ausblick

Definition:

Redundanz bedeutet, dass dieselbe Information **mehrfach** gespeichert wird.

Probleme:

- Speicherplatz verschwendet
- Inkonsistenzen möglich
- Änderungen müssen mehrfach erfolgen
- Fehleranfällig



6x gespeichert

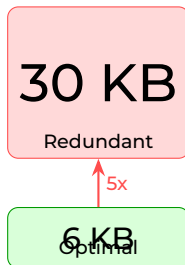
aber nur 3 Künstler!

Speicherkosten:

- Bei 1000 Alben von 200 Künstlern:
- Jeder Künstlername ~ 30 Bytes
- Redundant: $1000 \times 30 = 30.000$ Bytes
- Optimal: $200 \times 30 = 6.000$ Bytes
- **5x mehr Speicher!**

Wartungskosten:

- Jede Änderung an vielen Stellen
- Zeitaufwand multipliziert sich
- Fehlerwahrscheinlichkeit steigt



Konsistenzkosten

Eine vergessene Änderung führt zu widersprüchlichen Daten – das Schlimmste!

Eine Bibliothek speichert Ausleihen:

AusleihNr	Nutzer	Adresse	Buch	Autor	Datum
A001	Müller	Hauptstr. 1	Der Prozess	Kafka	2026-01-10
A002	Müller	Hauptstr. 1	Das Schloss	Kafka	2026-01-12
A003	Müller	Hauptstr. 1	1984	Orwell	2026-01-15
A004	Schmidt	Bachweg 7	Der Prozess	Kafka	2026-01-16
A005	Weber	Ringstr. 3	1984	Orwell	2026-01-18

Redundanzen:

- “Müller, Hauptstr. 1” erscheint **3x**
- “Der Prozess, Kafka” erscheint **2x**
- “1984, Orwell” erscheint **2x**

Eine Schule speichert Kursbelegungen:

Schüler	Klasse	Lehrer_Kl	Kurs	Kurslehrer	Raum	Note
Anna	10a	Fr. Berg	Mathe	Hr. Klein	R201	2
Ben	10a	Fr. Berg	Mathe	Hr. Klein	R201	3
Clara	10a	Fr. Berg	Mathe	Hr. Klein	R201	1
Anna	10a	Fr. Berg	Deutsch	Fr. Groß	R105	2
Ben	10a	Fr. Berg	Deutsch	Fr. Groß	R105	2

Mehrfach gespeichert:

- Klassenlehrer “Fr. Berg” – bei jedem Schüler der 10a
- Kursinfos “Mathe, Hr. Klein, R201” – bei jedem Teilnehmer

Ein Krankenhaus speichert Behandlungen:

Patient	Geb.Datum	Arzt	Abteilung	Behandlung	Datum
Huber	15.03.1980	Dr. Meier	Chirurgie	OP Knie	10.01.26
Huber	15.03.1980	Dr. Meier	Chirurgie	Kontrolle	20.01.26
Fischer	22.07.1965	Dr. Schulz	Innere	Bluttest	12.01.26
Fischer	22.07.1965	Dr. Schulz	Innere	MRT	15.01.26
Fischer	22.07.1965	Dr. Schulz	Innere	Besprechung	18.01.26

Kritische Redundanzen:

- Patientendaten bei jeder Behandlung wiederholt
- Arzt-Abteilungs-Zuordnung mehrfach gespeichert
- **Bei Tippfehler:** Falsches Geburtsdatum kann zu Verwechslungen führen!

BestellNr	Kunde	Adresse	Artikel	Preis	Datum
1001	Müller	Hauptstr. 1	Laptop	999	2026-01-15
1001	Müller	Hauptstr. 1	Maus	29	2026-01-15
1001	Müller	Hauptstr. 1	Tastatur	79	2026-01-15
1002	Schmidt	Bachweg 7	Monitor	349	2026-01-16
1003	Müller	Hauptstr. 1	Webcam	89	2026-01-20

Redundanzen identifizieren:

- “Müller, Hauptstr. 1” erscheint **4x**
- “BestellNr 1001, 2026-01-15” erscheint **3x**

BestellNr	Kunde	Adresse	Artikel	Preis	Datum
1001	Müller	Hauptstr. 1	Laptop	999	2026-01-15
1001	Müller	Hauptstr. 1	Maus	29	2026-01-15
1001	Müller	Hauptstr. 1	Tastatur	79	2026-01-15
1002	Schmidt	Bachweg 7	Monitor	349	2026-01-16
1003	Müller	Hauptstr. 1	Webcam	89	2026-01-20

Redundanzen identifizieren:

- “Müller, Hauptstr. 1” erscheint **4x**
- “BestellNr 1001, 2026-01-15” erscheint **3x**

Frage

Was passiert, wenn Herr Müller umzieht?

Stellen Sie sich vor, wir erweitern unsere Spieler-Tabelle:

Spieler	Position	Verein	Vereinsort	Stadion	Tore
Müller	Sturm	Bayern München	München	Allianz Arena	8
Neuer	Tor	Bayern München	München	Allianz Arena	0
Kimmich	Mittelfeld	Bayern München	München	Allianz Arena	3
Wirtz	Mittelfeld	Bayer Leverkusen	Leverkusen	BayArena	11
Tah	Abwehr	Bayer Leverkusen	Leverkusen	BayArena	0

Redundanz:

- Vereinsinformationen werden pro Spieler wiederholt
- Bei 30 Bayern-Spielern: 30x “München, Allianz Arena”

Welche Daten sind in dieser Tabelle redundant gespeichert?

Kurs	Dozent	Raum	Student	Matrikelnr	Note
DMA	Flath	S103	Müller	12345	1.7
DMA	Flath	S103	Schmidt	12346	2.0
DMA	Flath	S103	Weber	12347	1.3
BWL	Meier	S201	Müller	12345	2.3
BWL	Meier	S201	Weber	12347	1.7

Welche Daten sind in dieser Tabelle redundant gespeichert?

Kurs	Dozent	Raum	Student	Matrikelnr	Note
DMA	Flath	S103	Müller	12345	1.7
DMA	Flath	S103	Schmidt	12346	2.0
DMA	Flath	S103	Weber	12347	1.3
BWL	Meier	S201	Müller	12345	2.3
BWL	Meier	S201	Weber	12347	1.7

Redundant:

- Kursinformationen (Dozent, Raum) pro Teilnehmer wiederholt
- Studentinformationen bei mehreren Kursen wiederholt

- 1 Rückblick & Motivation
- 2 Die “Mega-Tabelle” – Ein Beispiel
- ▶ **3 Anomalien: Was schief gehen kann**
- 4 Anomalien selbst erleben
- 5 Die Lösung: Daten aufteilen
- 6 Erste Schritte zur Modellierung
- 7 Zusammenfassung & Ausblick

Problem: Bei Änderung muss an **vielen Stellen** geändert werden.

Spieler	Verein	Vereinsort	Stadion
Müller	Bayern München	München	Allianz Arena
Neuer	Bayern München	München	Allianz Arena
Kimmich	Bayern München	München	Allianz Arena

Szenario: Bayern zieht in ein neues Stadion um.

```
UPDATE spieler  
SET Stadion = 'FC Bayern Arena'  
WHERE Verein = 'Bayern München';
```

Gefahr

Vergisst man **eine** Zeile, sind die Daten **inkonsistent!**

Szenario: Die Bibliothek zieht um – neue Adresse.

Buch	Bibliothek	Adresse	Telefon
Der Prozess	Stadtbibliothek	Marktplatz 1	0931-123
1984	Stadtbibliothek	Marktplatz 1	0931-123
Faust	Stadtbibliothek	Marktplatz 1	0931-123
Harry Potter	Unibibliothek	Campus 5	0931-456

Szenario: Die Bibliothek zieht um – neue Adresse.

Buch	Bibliothek	Adresse	Telefon
Der Prozess	Stadtbibliothek	Marktplatz 1	0931-123
1984	Stadtbibliothek	Marktplatz 1	0931-123
Faust	Stadtbibliothek	Marktplatz 1	0931-123
Harry Potter	Unibibliothek	Campus 5	0931-456

Nach dem “Update” (ein Fehler passiert):

Buch	Bibliothek	Adresse	Telefon
Der Prozess	Stadtbibliothek	Rathausstr. 10	0931-123
1984	Stadtbibliothek	Rathausstr. 10	0931-123
Faust	Stadtbibliothek	Marktplatz 1	0931-123
Harry Potter	Unibibliothek	Campus 5	0931-456

Inkonsistenz!

Wo ist die Stadtbibliothek jetzt? Marktplatz 1 oder Rathausstr. 10?

Problem: Informationen können nicht gespeichert werden, weil **andere Daten fehlen**.

Szenario: Ein neuer Verein (RB Leipzig II) steigt auf, hat aber noch keine Spieler verpflichtet.

Spieler	Verein	Stadion
Müller	Bayern	Allianz Arena
???	RB Leipzig II	Bruno-Plache

Problem

Ohne Spieler kann der Verein nicht in die Tabelle!

Oder wir brauchen NULL-Werte / Dummy-Daten.

Real-World: Sie möchten einen Künstler speichern, der noch kein Album hat.

Bibliothek: Neues Buch anschaffen, das noch niemand ausgeliehen hat.

AusleihNr	Nutzer	Buch	ISBN	Verlag
A001	Müller	Der Prozess	978-3-123	Fischer
A002	Schmidt	1984	978-3-456	Ullstein
???	???	Neues Buch	978-3-789	Suhrkamp

Bibliothek: Neues Buch anschaffen, das noch niemand ausgeliehen hat.

AusleihNr	Nutzer	Buch	ISBN	Verlag
A001	Müller	Der Prozess	978-3-123	Fischer
A002	Schmidt	1984	978-3-456	Ullstein
???	???	Neues Buch	978-3-789	Suhrkamp

Krankenhaus: Neuen Arzt einstellen, der noch keinen Patienten hat.

Patient	Arzt	Abteilung	Behandlung
Huber	Dr. Meier	Chirurgie	OP Knie
Fischer	Dr. Schulz	Innere	Bluttest
???	Dr. Neu	Neurologie	???

Konsequenz

Wichtige Informationen können nicht gespeichert werden!

Problem: Beim Löschen gehen **ungewollt andere Informationen** verloren.

Spieler	Verein	Vereinsort	Stadion
Wirtz	Bayer Leverkusen	Leverkusen	BayArena
Tah	Bayer Leverkusen	Leverkusen	BayArena

Szenario: Wirtz und Tah wechseln beide ins Ausland.

```
DELETE FROM spieler  
WHERE Spieler IN ( 'Wirtz', 'Tah' );
```

Gefahr

Plötzlich wissen wir nicht mehr, dass Bayer Leverkusen in der BayArena spielt!
⇒ Vereinsinformation komplett verloren

Szenario Bibliothek: Das einzige Exemplar eines Buches wird zurückgegeben.

	AusleihNr	Nutzer	Buch	Autor	Verlag
Vorher:	A001	Müller	Der Prozess	Kafka	Fischer
	A002	Schmidt	1984	Orwell	Ullstein
	A003	Weber	Das Schloss	Kafka	Fischer

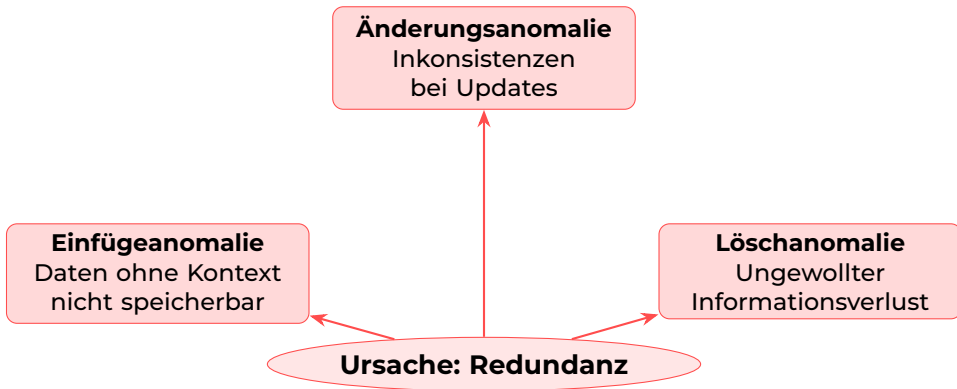
Szenario Bibliothek: Das einzige Exemplar eines Buches wird zurückgegeben.

	AusleihNr	Nutzer	Buch	Autor	Verlag
Vorher:	A001	Müller	Der Prozess	Kafka	Fischer
	A002	Schmidt	1984	Orwell	Ullstein
	A003	Weber	Das Schloss	Kafka	Fischer

	AusleihNr	Nutzer	Buch	Autor	Verlag
Nachher (A003 gelöscht):	A001	Müller	Der Prozess	Kafka	Fischer
	A002	Schmidt	1984	Orwell	Ullstein

Informationsverlust!

Das Buch "Das Schloss" existiert nicht mehr im System – obwohl wir es noch haben!



Merke: Alle drei Anomalien haben dieselbe Wurzel – **Redundanz durch schlechtes Tabellendesign.**

Ordnen Sie die Szenarien den Anomalien zu:

Szenario	Anomalie?
Die Telefonnummer eines Kunden ändert sich, ist aber in 50 Bestellungen gespeichert	???
Ein neuer Lieferant soll erfasst werden, hat aber noch nie geliefert	???
Der letzte Mitarbeiter einer Abteilung kündigt, und wir vergessen den Abteilungsnamen	???

Ordnen Sie die Szenarien den Anomalien zu:

Szenario	Anomalie?
Die Telefonnummer eines Kunden ändert sich, ist aber in 50 Bestellungen gespeichert	???
Ein neuer Lieferant soll erfasst werden, hat aber noch nie geliefert	???
Der letzte Mitarbeiter einer Abteilung kündigt, und wir vergessen den Abteilungsnamen	???

Lösungen: Änderungsanomalie, Einfügeanomalie, Löschanomalie

Weitere Szenarien – welche Anomalie liegt vor?

- 1 Ein Professor geht in Rente. Damit verschwinden alle seine Kurse aus dem System.

Weitere Szenarien – welche Anomalie liegt vor?

- 1 Ein Professor geht in Rente. Damit verschwinden alle seine Kurse aus dem System.
⇒ **Löschanomalie**
- 2 Die Postleitzahl einer Stadt ändert sich. Sie müssen 500 Kundendatensätze durchsuchen.

Weitere Szenarien – welche Anomalie liegt vor?

- ① Ein Professor geht in Rente. Damit verschwinden alle seine Kurse aus dem System.
⇒ **Löschanomalie**
- ② Die Postleitzahl einer Stadt ändert sich. Sie müssen 500 Kundendatensätze durchsuchen.
⇒ **Änderungsanomalie**
- ③ Sie möchten ein neues Produkt ins Sortiment aufnehmen, aber es wurde noch nie bestellt.

Weitere Szenarien – welche Anomalie liegt vor?

- ① Ein Professor geht in Rente. Damit verschwinden alle seine Kurse aus dem System.
⇒ **Löschanomalie**
- ② Die Postleitzahl einer Stadt ändert sich. Sie müssen 500 Kundendatensätze durchsuchen.
⇒ **Änderungsanomalie**
- ③ Sie möchten ein neues Produkt ins Sortiment aufnehmen, aber es wurde noch nie bestellt.
⇒ **Einfügeanomalie**
- ④ Ein Hotelzimmer wird renoviert. Die neue Zimmerkategorie muss in 200 Buchungsdatensätzen geändert werden.

Weitere Szenarien – welche Anomalie liegt vor?

- ① Ein Professor geht in Rente. Damit verschwinden alle seine Kurse aus dem System.
⇒ **Löschanomalie**
- ② Die Postleitzahl einer Stadt ändert sich. Sie müssen 500 Kundendatensätze durchsuchen.
⇒ **Änderungsanomalie**
- ③ Sie möchten ein neues Produkt ins Sortiment aufnehmen, aber es wurde noch nie bestellt.
⇒ **Einfügeanomalie**
- ④ Ein Hotelzimmer wird renoviert. Die neue Zimmerkategorie muss in 200 Buchungsdatensätzen geändert werden.
⇒ **Änderungsanomalie**

Schlechtes Design:

Kurs	Dozent	Student
DMA	Flath	Müller
DMA	Flath	Schmidt
DMA	Flath	Weber
BWL	Meier	Müller

- ✗ Dozent 3x wiederholt
- ✗ Müller 2x wiederholt
- ✗ Alle Anomalien möglich

Gutes Design:

Kurs	Dozent
DMA	Flath
BWL	Meier

Kurs	Student
DMA	Müller
DMA	Schmidt
DMA	Weber
BWL	Müller

- ✓ Keine Redundanz
- ✓ Keine Anomalien

- 1 Rückblick & Motivation
- 2 Die “Mega-Tabelle” – Ein Beispiel
- 3 Anomalien: Was schief gehen kann
- ▶ **4 Anomalien selbst erleben**
- 5 Die Lösung: Daten aufteilen
- 6 Erste Schritte zur Modellierung
- 7 Zusammenfassung & Ausblick

Hands-on

Anomalien selbst erleben

marimo: 05-redundanz-probleme.py

Aufgaben 5.1 – 5.3

Im Notebook werden Sie:

① Redundanzen identifizieren

- Eine “schlechte” Tabelle analysieren
- Doppelte Informationen zählen

② Änderungsanomalie provozieren

- Ein UPDATE durchführen
- Inkonsistenz erzeugen

③ Löschanomalie erleben

- Daten löschen
- Informationsverlust beobachten

Ziel

Die Probleme am eigenen Leib erfahren – nicht nur theoretisch verstehen!

15 Minuten Pause

Nach der Pause:
Die Lösung – Daten intelligent aufteilen

- 1 Rückblick & Motivation
- 2 Die “Mega-Tabelle” – Ein Beispiel
- 3 Anomalien: Was schief gehen kann
- 4 Anomalien selbst erleben
- ▶ **5 Die Lösung: Daten aufteilen**
- 6 Erste Schritte zur Modellierung
- 7 Zusammenfassung & Ausblick

Problem: Alles in einer Tabelle \Rightarrow Redundanz

Lösung: Jedes “Ding” bekommt seine eigene Tabelle!

Vorher: Eine Tabelle

Spieler	Verein	Stadion
Müller	Bayern	Allianz
Neuer	Bayern	Allianz
Wirtz	Leverkusen	BayArena

Nachher: Zwei Tabellen

ID	Verein	Stadion
1	Bayern	Allianz
2	Leverkusen	BayArena

Spieler	Verein_ID
Müller	1
Neuer	1
Wirtz	2

\Rightarrow Vereinsinformation nur **einmal** gespeichert!

Änderungen:

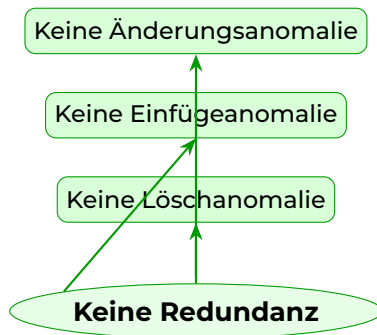
- Stadion ändern? \Rightarrow Eine Zeile!
- Keine Inkonsistenzen möglich

Einfügen:

- Neuer Verein ohne Spieler? \Rightarrow Kein Problem!
- Informationen unabhängig speicherbar

Löschen:

- Letzter Spieler weg? \Rightarrow Verein bleibt erhalten!
- Kein Informationsverlust



Vorher (eine Tabelle):

Nutzer	Buch	Autor
Müller	Prozess	Kafka
Müller	Schloss	Kafka
Schmidt	Prozess	Kafka

“Kafka” 3x gespeichert

Nachher (drei Tabellen):

B_ID	Titel
1	Prozess
2	Schloss

A_ID	Name
1	Kafka

Nutzer	B_ID	A_ID
Müller	1	1
Müller	2	1
Schmidt	1	1

Ergebnis

“Kafka” nur **einmal** gespeichert – verknüpft über ID!

Tabelle: Künstler

ID	Name	Land
1	The Beatles	UK
2	Michael Jackson	USA
3	Pink Floyd	UK

Tabelle: Alben

ID	Titel	Jahr	K_ID
1	Abbey Road	1969	1
2	Let It Be	1970	1
3	Thriller	1982	2
4	Bad	1987	2
5	Dark Side...	1973	3

Vorteile:

- “The Beatles, UK” nur **einmal**
- Künstler ohne Album möglich
- Land ändern = eine Zeile

Schlüsselprinzip

Die Spalte K_ID in Alben **verweist** auf die Künstler-Tabelle.

⇒ **Fremdschlüssel**

Definition:

Ein **Primärschlüssel** (Primary Key, PK) ist ein Attribut (oder eine Attributkombination), das jede Zeile einer Tabelle **eindeutig identifiziert**.

Eigenschaften:

- **Eindeutig:** Keine zwei Zeilen haben denselben Wert
- **Nicht NULL:** Muss immer einen Wert haben
- **Stabil:** Sollte sich nicht ändern

ID	Name	Land
1	Beatles	UK
2	Jackson	USA
3	Pink Floyd	UK



Beispiele für PKs:

- Matrikelnummer
- ISBN
- Ausweisnummer

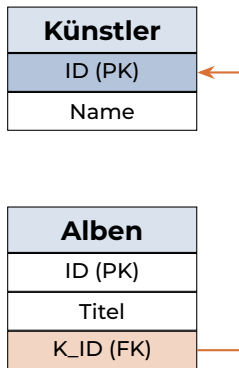
Definition:

Ein **Fremdschlüssel** (Foreign Key, FK) ist ein Attribut, das auf den Primärschlüssel einer **anderen** Tabelle verweist.

Zweck:

- Stellt **Beziehungen** zwischen Tabellen her
- Vermeidet Redundanz
- Gewährleistet **referenzielle Integrität**

Regel: Ein FK-Wert muss als PK in der referenzierten Tabelle existieren!





Primärschlüssel (PK):

- Identifiziert jede Zeile **eindeutig**
- Keine Duplikate erlaubt
- Darf nicht NULL sein

Fremdschlüssel (FK):

- Verweist auf PK einer anderen Tabelle
- Stellt **Beziehung** her
- Kann sich wiederholen

```
CREATE TABLE kuenstler (  
    id INTEGER PRIMARY KEY,  
    name TEXT NOT NULL,  
    land TEXT  
);  
  
CREATE TABLE alben (  
    id INTEGER PRIMARY KEY,  
    titel TEXT NOT NULL,  
    jahr INTEGER,  
    kuenstler_id INTEGER,  
    FOREIGN KEY (kuenstler_id) REFERENCES kuenstler(id)  
);
```

Wichtige Konzepte:

- PRIMARY KEY: Eindeutige Identifikation
- FOREIGN KEY: Referenz auf andere Tabelle
- REFERENCES: Gibt an, auf welche Tabelle/Spalte verwiesen wird

Definition: Jeder Fremdschlüsselwert muss auf einen existierenden Primärschlüssel verweisen.

Gültig:

ID	Name
1	Beatles
2	Jackson

Titel	K_ID
Abbey Road	1
Thriller	2

✓ K_ID 1 und 2 existieren

Ungültig:

ID	Name
1	Beatles
2	Jackson

Titel	K_ID
Abbey Road	1
Neues Album	99

✗ K_ID 99 existiert nicht!

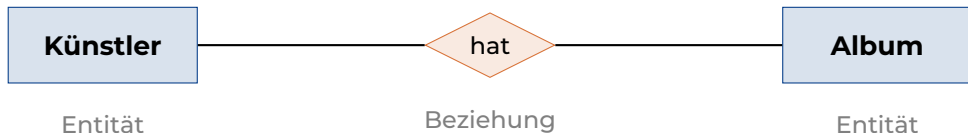
Schutz durch das DBMS

Die Datenbank **verhindert** ungültige Fremdschlüsselwerte automatisch!

- 1 Rückblick & Motivation
- 2 Die “Mega-Tabelle” – Ein Beispiel
- 3 Anomalien: Was schief gehen kann
- 4 Anomalien selbst erleben
- 5 Die Lösung: Daten aufteilen
- ▶ **6 Erste Schritte zur Modellierung**
- 7 Zusammenfassung & Ausblick

Kernfrage der Datenmodellierung:

- 1 Welche “Dinge” (Entitäten) gibt es?
- 2 Wie hängen sie zusammen (Beziehungen)?



Entität: Ein “Ding” der realen Welt (Künstler, Album, Spieler, Verein)

Beziehung: Verbindung zwischen Entitäten (hat, gehört_zu, spielt_für)

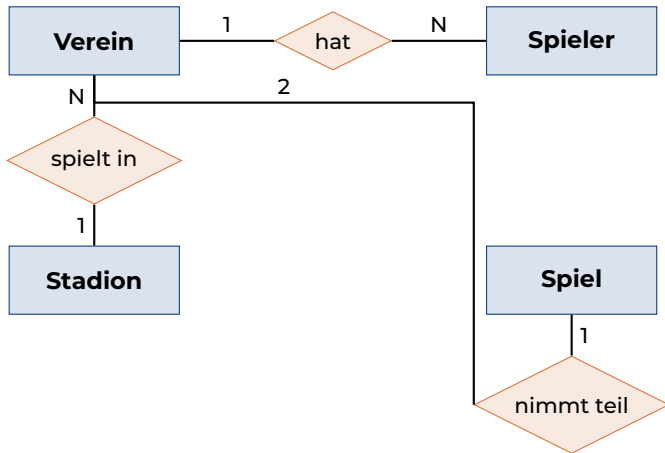
Wichtige Frage: Wie viele Entitäten stehen in Beziehung?



“Ein Künstler hat **mehrere** Alben”
“Ein Album gehört zu **einem** Künstler”

Typische Kardinalitäten:

- **1:1** – Ein Bürger hat einen Personalausweis
- **1:N** – Ein Verein hat mehrere Spieler
- **M:N** – Studenten besuchen Kurse (mehrere zu mehreren)



Entitäten: Verein, Spieler, Stadion, Spiel

Beziehungen: hat (1:N), spielt_in (N:1), nimmt_teil (2:1)

Welche Kardinalität haben diese Beziehungen?

Beziehung	Kardinalität
Ein Buch hat einen Verlag	???
Ein Student belegt Kurse	???
Eine Person hat eine Sozialversicherungsnnummer	???
Ein Dozent hält Vorlesungen	???

Welche Kardinalität haben diese Beziehungen?

Beziehung	Kardinalität
Ein Buch hat einen Verlag	???
Ein Student belegt Kurse	???
Eine Person hat eine Sozialversicherungsnnummer	???
Ein Dozent hält Vorlesungen	???

Lösungen:

- Buch – Verlag: N:1 (viele Bücher, ein Verlag)
- Student – Kurse: M:N (viele zu vielen)
- Person – SVN: 1:1 (eindeutig)
- Dozent – Vorlesungen: 1:N (ein Dozent, viele Vorlesungen)

Hands-on

Tabellen aufteilen

marimo: 05-redundanz-probleme.py

Aufgaben 5.4 – 5.6

Im zweiten Teil des Notebooks:

① Tabellen aufteilen

- Die “schlechte” Tabelle in zwei Tabellen zerlegen
- Fremdschlüssel einführen

② Änderung testen

- Dieselbe Änderung wie vorher durchführen
- Sehen: Nur eine Zeile betroffen!

③ Daten abfragen

- Vorschau auf JOINS (nächste Vorlesung)
- Wie kombiniert man die Tabellen wieder?

- 1 Rückblick & Motivation
- 2 Die “Mega-Tabelle” – Ein Beispiel
- 3 Anomalien: Was schief gehen kann
- 4 Anomalien selbst erleben
- 5 Die Lösung: Daten aufteilen
- 6 Erste Schritte zur Modellierung
- ▶ **7 Zusammenfassung & Ausblick**

Probleme der Mega-Tabelle:

- Redundanz (gleiche Daten mehrfach)
- Änderungsanomalie
- Einfügeanomalie
- Löschanomalie

Lösung:

- Daten in mehrere Tabellen aufteilen
- Jede Entität eigene Tabelle
- Verbindung durch Schlüssel

Neue Konzepte:

- **Entität:** "Ding" der realen Welt
- **Beziehung:** Verbindung zwischen Entitäten
- **Primärschlüssel:** Eindeutige ID
- **Fremdschlüssel:** Verweis auf andere Tabelle
- **Kardinalität:** 1:1, 1:N, M:N

Begriff	Bedeutung
Redundanz	Mehrfache Speicherung derselben Information
Änderungsanomalie	Inkonsistenzen bei Updates
Einfügeanomalie	Daten ohne Kontext nicht speicherbar
Löschanomalie	Ungewollter Informationsverlust beim Löschen
Entität	“Ding” der realen Welt (wird zu Tabelle)
Primärschlüssel (PK)	Eindeutige Identifikation einer Zeile
Fremdschlüssel (FK)	Verweis auf PK einer anderen Tabelle
Kardinalität	Anzahl der Beziehungen (1:1, 1:N, M:N)
Referenzielle Integrität	FK-Werte müssen als PK existieren

Vorlesung 6: Entity-Relationship-Modellierung

- Formale ER-Notation (Chen / Krähenfuß)
- Komplexe Modelle erstellen
- Übung: Eigene Modelle zeichnen

Vorlesung 7: Relationales Modell

- ER-Modelle in Tabellen umwandeln
- CREATE TABLE Statements
- Integritätsbedingungen

Vorschau

Vorlesung 9: JOINS – Wie kombiniert man mehrere Tabellen in Abfragen?

1. Welche Anomalie tritt auf, wenn man einen Wert ändert und er an mehreren Stellen steht?

- A) Einfügeanomalie
- B) Änderungsanomalie
- C) Löschanomalie

1. Welche Anomalie tritt auf, wenn man einen Wert ändert und er an mehreren Stellen steht?

- A) Einfügeanomalie
- B) Änderungsanomalie
- C) Löschanomalie

Antwort: B

2. Welcher Schlüsseltyp verweist auf einen Primärschlüssel in einer anderen Tabelle?

- A) Superkey
- B) Primärschlüssel
- C) Fremdschlüssel

1. Welche Anomalie tritt auf, wenn man einen Wert ändert und er an mehreren Stellen steht?

- A) Einfügeanomalie
- B) Änderungsanomalie
- C) Löschanomalie

Antwort: B

2. Welcher Schlüsseltyp verweist auf einen Primärschlüssel in einer anderen Tabelle?

- A) Superkey
- B) Primärschlüssel
- C) Fremdschlüssel

Antwort: C

3. Was ist die Hauptursache aller drei Anomalien?

- A) Zu viele Tabellen
- B) Redundanz
- C) Fehlende Primärschlüssel

3. Was ist die Hauptursache aller drei Anomalien?

- A) Zu viele Tabellen
- B) Redundanz
- C) Fehlende Primärschlüssel

Antwort: B

4. Ein Künstler kann mehrere Alben haben, aber jedes Album gehört zu genau einem Künstler. Welche Kardinalität liegt vor?

- A) 1:1
- B) 1:N
- C) M:N

3. Was ist die Hauptursache aller drei Anomalien?

- A) Zu viele Tabellen
- B) Redundanz
- C) Fehlende Primärschlüssel

Antwort: B

4. Ein Künstler kann mehrere Alben haben, aber jedes Album gehört zu genau einem Künstler. Welche Kardinalität liegt vor?

- A) 1:1
- B) 1:N
- C) M:N

Antwort: B

Fragen?

Nächste Woche: Entity-Relationship-Modellierung

*“Gutes Datenbankdesign ist wie gute Architektur –
man sieht es erst, wenn etwas schief geht.”*