

## 1 Grundbegriffe

### Analoge Daten

Kontinuierliche, stufenlose Werte (z. B. Temperaturkurve, Zeiger).

### Digitale Daten

Diskrete Werte in festen Schritten, meist binär (Bits).

### Strukturierte Daten

Festes Schema, gleiche Spalten für alle Zeilen (z. B. DB-Tabelle).

### Semistrukturierte Daten

Beschriftete Felder, aber kein starres Schema (z. B. XML/JSON, Word/Excel).

### Unstrukturierte Daten

Kein Schema (Fliesstext, Bild, Video).

### Tabelle / Relation

Entitätsmenge; Zeilen = Datensätze, Spalten = Attribute.

### Attribut

Eine Spalte / Eigenschaft einer Tabelle.

### Attributwert

Konkreter Eintrag eines Attributs in 1 Datensatz.

### Datensatz

Eine Zeile = 1 konkretes Objekt / eine Entität.

### Beziehung

Fachlicher Zusammenhang zwischen 2 Entitätsmengen.

### Einfache Beziehung

Verbindet genau 2 Entitätsmengen.

### Komplexe Beziehung

Verbindet mehr als 2 Entitätsmengen gleichzeitig.

### Primärschlüssel (PK)

Identifiziert jeden Datensatz eindeutig. Nie leer, nie doppelt.

### Fremdschlüssel (FK)

Verweist auf den PK einer (anderen) Tabelle; gleicher Datentyp wie der PK.

### Datentyp

Art der Werte: Text, Ganzzahl, Dezimalzahl, Datum, Ja/Nein, ...

### Wertebereich

Technisch gültige Werte für 1 Attribut (z. B. Note 1.0–6.0).

### Domäne

Wiederverwendbarer, fachlich sinnvoller Wertebereich für mehrere Attribute.

### Redundanz

Dieselbe Info ist unnötig mehrfach gespeichert (Änderungsanomalie-Risiko).

### Konsistenz

Daten sind widerspruchsfrei.

### Inkonsistenz

Gespeicherte Daten widersprechen sich.

**Strukturiert ≠ fehlerfrei!** Struktur (Schema) und Datenqualität sind unabhängige Eigenschaften.

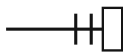
| Sym.      | Bedeutung     | Min | Max  |
|-----------|---------------|-----|------|
| <b>1</b>  | einfach       | 1   | 1    |
| <b>c</b>  | konditionell  | 0   | 1    |
| <b>m</b>  | mehrfach      | 1   | meh. |
| <b>mc</b> | mehrfachkond. | 0   | meh. |

## 2 Beziehungstypen 1/c/m/mc

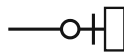
**1** = genau 1 · **c** = 0 oder 1 (konditionell) · **m** = 1 oder mehrere (mehrfach) · **mc** = 0, 1 oder mehrere (mehrfachkonditionell).

### Zeichnen (Krähenfussnotation)

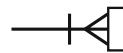
Symbol direkt am Kästchen = Maximum (Strich = 1, Krähenfuss = mehrere). Symbol weiter aussen (zur Linienmitte hin) = Minimum (Strich = 1, Kreis = 0). Kästchen sind rechts, Linie geht nach links.



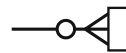
**1** = Strich + Strich



**c** = Strich + Kreis



**m** = Krähenfuss + Strich

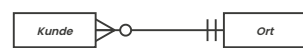


**mc** = Krähenfuss + Kreis

**Methode (2-Sätze-Technik):** Satz A: „Jedes E1 [Beziehung] → wie viele E2?“ ergibt den **2. Wert** (Y, Symbol bei E2).

Satz B: „Jedes E2 [Beziehung] → wie viele E1?“ ergibt den **1. Wert** (X, Symbol bei E1).

Zusammensetzen: **Typ = E1:E2 = X:Y**



Kunde:Ort = mc:1 — Satz A „Jeder Kunde wohnt an wie vielen Orten?“ → genau 1 (Y, bei Ort). Satz B „An jedem Ort wohnen wie viele Kunden?“ → 0, 1, mehrere (mc, bei Kunde).

**Kontrolle:** Klingen Satz A und Satz B sehr ähnlich? Dann wurde vermutlich nur 1 Richtung betrachtet — nochmals prüfen!

## 3 Datenqualität

| Typ            | Merkmal                                                                                                                   |
|----------------|---------------------------------------------------------------------------------------------------------------------------|
| Redundanz      | Exakt derselbe Inhalt mehrfach gespeichert.                                                                               |
| Inkonsistenz   | Mehrfach gespeicherte Duplikate widersprechen sich.                                                                       |
| Falsche Daten  | Falscher Datentyp; mehrere Werte in 1 Feld (nicht atomar); FK verweist auf nicht existierende Zeile; PK doppelt vergeben. |
| Fehlende Daten | Leere Pflichtfelder.                                                                                                      |

**Kein Fehler:** wiederholter FK-Wert (normal bei einer „mehrfach“-Beziehung, z. B. mc:1) · Lücken oder unsortierte IDs (ein PK muss nur eindeutig sein, nicht lückenlos/sortiert) · dieselbe Person/Firma erscheint 2× mit 2 verschiedenen, aber je korrekten Einträgen.

**Fehler benennen (Prüfungsantwort):** Tabelle + Zeilennummer + kurze Begründung. Die Spalte „Zeile“ selbst gehört meist nicht zur Tabelle, sie dient nur der Identifikation.

4 Konzeptionell vs. logisch

|         | Konzeptionell                    | Logisch                               |
|---------|----------------------------------|---------------------------------------|
| Enthält | Tabellen, Attribute, Beziehungen | + PK, FK, wenn nötig Zwischentabellen |
| Typen   | alle 10 (inkl. m:m, m:mc, mc:mc) | nur 7 (ohne m:m, m:mc, mc:mc)         |

**Zwischentabelle nötig**, wenn beide Seiten „mehrere“ sind (m:m, m:mc, mc:mc) — 1 FK-Spalte kann immer nur 1 Wert speichern.

**Aufbau**: eigene neue Tabelle mit neuer eigener PK + 2 FK (je 1 pro Originaltabelle). Name generisch (z. B. Autor\_Buch), nicht auf den Einzelfall bezogen.

**FK-Regel bei nur 1 „mehrfach“-Seite** (z. B. Typ mc:1): FK kommt in die Tabelle der „mehreren“-Seite, benannt nach der Tabelle, auf die verwiesen wird (z. B. Verlag\_ID in Buch).

**Jede Tabelle** im logischen Modell: eigene neue PK + alle ursprünglichen Attribute bleiben + FK zusätzlich, falls nötig.

5 Normalisierung

| Form  | Bedingung                                                            |
|-------|----------------------------------------------------------------------|
| 0. NF | Wiederholungsgruppen / nicht-atomare Werte (mehrere Werte in 1 Feld) |
| 1. NF | Alle Attributwerte atomar (genau 1 Wert pro Feld)                    |
| 2. NF | 1. NF + keine <i>partiellen</i> Abhängigkeiten vom PK                |
| 3. NF | 2. NF + keine <i>transitiven</i> Abhängigkeiten                      |

**Voll funktional abhängig**: Attribut hängt vom **ganzen** PK ab, nicht nur von einem Teil.

**Partielle Abhängigkeit** (verletzt 2. NF): Attribut hängt nur von **einem Teil** eines mehrteiligen Schlüssels ab.

*Bsp.: (KundeNr, ProduktNr, KundenName, Menge) — KundenName hängt nur von KundeNr ab, nicht vom ganzen Schlüssel → partiell.*

**Transitive Abhängigkeit** (verletzt 3. NF): Attribut hängt von einem **Nichtschlüsselattribut** ab statt direkt vom PK.

*Bsp.: (RechnungsNr, KundeNr, KundenOrt) — KundenOrt hängt von KundeNr ab, nicht direkt von RechnungsNr → transitiv.*

**Merksatz**: Besteht der PK nur aus 1 Attribut, ist 2. NF automatisch erfüllt (partielle Abhängigkeit braucht einen mehrteiligen Schlüssel) — dann nur noch auf Transitivität (3. NF) prüfen.

6 Excel & SQLite

| Funktion   | Zweck                                                |
|------------|------------------------------------------------------|
| Filtern    | Zeilen ausblenden                                    |
| Sortieren  | Reihenfolge ändern (zählt nicht als Datenkorrektur!) |
| ZÄHLENWENN | Anzahl mit Bedingung zählen                          |
| SUMMEWENN  | Summe mit Bedingung                                  |
| SVERWEIS   | Wert in anderer Tabelle nachschlagen                 |
| Pivot      | Daten gruppieren / zusammenfassen                    |

**Excel**: einfach, aber erzwingt keine referentielle Integrität (FK-Gültigkeit) — Redundanzrisiko, skaliert schlecht.

**SQLite**: relationale Datenbank in 1 Datei, kein Server nötig — erzwingt referentielle Integrität automatisch (verhindert FK auf nicht existierende Zeile).

zibbi.ch/tools/quiz/module/162 — eigene Zusammenfassung, keine Prüfungsinhalte