Wirtschaftsinformatik

V04 — Tabellen verbinden: Schlüssel & Merge

Prof. Dr. Christian Bockermann Prof. Dr. Julian Grote
Prof. Dr. Volker Klingspor

Hochschule Bochum · Fachbereich Wirtschaft

Wintersemester 2026/27

Rückblick: Letzte Woche

- Frage: *Der Vertrieb meldet 1.616 Bestellungen, das Controlling 1.594 — wer hat recht?*
- Antwort: **Beide zählen richtig, aber nicht dasselbe** — der Vertrieb zählte jede Zeile einer ungeprüften Rohtabelle, das Controlling nur validierte, eindeutige Bestellungen.
- In der Lernwerkstatt (Digitalkompetenzwoche) haben Sie die Differenz von 22 Bestellungen selbst aufgeklärt: **8** Doppelerfassungen, **9** Testbestellungen, **1** Storno, **4** verwaiste Kundennummern.
- Eine der vier Qualitätsdimensionen war **Aktualität** — das Beispiel dort: eine Kundin, die längst umgezogen ist. Genau da machen wir heute weiter.

Eine Kundin heiratet und zieht um — was muss geändert werden, und an wie vielen Stellen?

▶ **Aktivierung — Wortmeldung:** Wo überall steht die Adresse einer Kundin in einem Unternehmen? Welche Stellen fallen Ihnen ein?

Zwei Tabellen, ein gemeinsamer Schlüssel

Bestellungen

bestellnr	bestelldatum	**kundennr**	bestellwert
8092	2026-01-10	**1074**	146.05
8093	2026-01-10	1471	42.73
8094	2026-01-10	1491	129.48

Kunden

kundennr	name	ort
1074	Heike Meier	Jena
1471	...	...
1491	...	...

Dieselben drei Bestellzeilen wie letzte Woche — jetzt neben der Tabelle, auf die kundennr verweist.

Drei Begriffe für das, was wir gerade gesehen haben

Primärschlüssel

eindeutig *in seiner Tabelle* — `kundennr` in Kunden

Fremdschlüssel

verweist auf einen Primärschlüssel *woanders* — `kundennr` in Bestellungen

1:n-Beziehung

eine Kundin, viele Bestellungen

Warum *muss* ein Primärschlüssel eindeutig sein? Das sehen wir gleich an einem echten Fall.

Heike Meiers vier echte Bestellungen, alles in einer Tabelle:

bestellnr	bestellwert	name	adresse
8056	123.18	Heike Meier	Am Hang 44, 07743 Jena
8092	146.05	Heike Meier	Am Hang 44, 07743 Jena
8770	199.04	Heike Meier	Am Hang 44, 07743 Jena
8798	149.37	Heike Meier	Am Hang 44, 07743 Jena

Viermal derselbe Name, viermal dieselbe Adresse — Wort für Wort identisch.

Dieselben vier Bestellungen, über **kundennr** verbunden:

bestellnr	bestellwert	kundennr
8056	123.18	1074
8092	146.05	1074
8770	199.04	1074
8798	149.37	1074

kundennr	name	adresse
1074	Heike Meier	Am Hang 44, 07743 Jena

Name und Adresse stehen jetzt **einmal** — die Bestellungen tragen nur noch den Verweis.

▶ **Aktivierung — Wortmeldung:** Heike Meier heiratet und zieht um — an wie vielen Stellen muss die neue Adresse eingetragen werden?

Zwei Zeilen, dieselbe Person?

Eine Adresse an einer Stelle zu halten löst das Redundanz-Problem — aber nur, wenn diese eine Stelle auch wirklich *eindeutig* ist. Was, wenn nicht?

kundennr	name	straße	ort
1046	Monika Müller	Bergstraße 38	Frankfurt am Main
1046	Monika Müller	Bergstr. 38	Frankfurt am Main

Ist das dieselbe Kundin, zweimal erfasst — oder ein Fehler im Schlüssel selbst?

Wie wir die Frage beantworten: der Plan

Erinnerung an die Frage der Stunde: „Eine Kundin heiratet und zieht um — was muss geändert werden, und an wie vielen Stellen?“

1. **Wo steht die Adresse heute?**
   In `kunden` — einmal je Kundin, über `kundennr` identifiziert.
2. **Was hinge daran, wenn sie in jeder Bestellung stünde?**
   Redundanz: bis zu 1.616 Kopien, Inkonsistenzrisiko bei jeder Änderung.
3. **Wie verbindet man die getrennten Tabellen wieder, wenn man beides braucht?**
   Über den gemeinsamen Schlüssel — in der Lernwerkstatt mit `merge` (dem „JOIN“ von `pandas`). Vorher: die Zeilenzahl *erwarten*.

Diesen Dreischritt gehen Sie gleich in der Lernwerkstatt selbst.

Verbinden: aus zwei Tabellen eine

Kunden

kundennr	name	ort
1074	Heike Meier	Jena
1471	Diana Schrader	Kassel
1491	Patrick Baumann	Berlin

Bestellungen

bestellnr	kundennr	bestellwert
8092	1074	146.05
8093	1471	42.73
8094	1491	129.48

⇓ **Verbinden** über `kundennr` (Python: `merge`, Datenbank: **JOIN**)

bestellnr	bestellwert	kundennr	name	ort
8092	146.05	1074	Heike Meier	Jena
8093	42.73	1471	Diana Schrader	Kassel
8094	129.48	1491	Patrick Baumann	Berlin

▶ **Aktivierung — Erwartung bilden:** Hier: 3 Bestellungen, 3 Ergebniszeilen. Wie viele Zeilen erwarten Sie bei allen 1.616 Bestellungen und 550 Kunden? Begründen Sie kurz.

Nicht jede Kundin hat bestellt: `inner` gegen `left`

Kunden

kundennr	name	ort
1057	Anna Weiß	Stuttgart
1074	Heike Meier	Jena
1471	Diana Schrader	Kassel
1491	Patrick Baumann	Berlin

Bestellungen

bestellnr	kundennr	bestellwert
8092	1074	146.05
8093	1471	42.73
8094	1491	129.48

`inner`: nur, wer auf beiden Seiten vorkommt

kundennr	name	bestellwert
1074	Heike Meier	146.05
1471	Diana Schrader	42.73
1491	Patrick Baumann	129.48

`left`: alle Kunden, auch ohne Bestellung

kundennr	name	bestellwert
1057	Anna Weiß	(leer)
1074	Heike Meier	146.05
1471	Diana Schrader	42.73
1491	Patrick Baumann	129.48

Anna Weiß hat noch nie bestellt. Welche Zeilenzahl „richtig“ ist, hängt von der Frage ab: *Bestellungen zählen* (inner genügt) oder *alle Kunden* (left).

Erinnerung an die Frage der Stunde: „Eine Kundin heiratet und zieht um — was muss geändert werden, und an wie vielen Stellen?“

Antwortsatz

An genau **einer** Stelle, wenn die Daten richtig aufgeteilt sind — und an **hunderten**, wenn nicht. Genau dafür gibt es Tabellen mit **Schlüsseln**.

- Heike Meiers Adresse steht *einmal* in der Kundentabelle — ihre vier Bestellungen verweisen nur über `kundennr` darauf.
- Diese eine Stelle muss aber auch *eindeutig* sein — sonst holt einen die Redundanz durch die Hintertür wieder ein (Monika Müller mit zwei Schreibweisen).

In der Lernwerkstatt: Tabellen verbinden

In der Lernwerkstatt verbinden Sie die Bestell- und die Kundentabelle selbst über `kundennr` — mit genau einem Befehl („merge“, der `pandas`-Name für den JOIN, den Sie gerade konzeptionell kennengelernt haben).

Vergleichen Sie danach die tatsächliche Zeilenzahl mit Ihrer Erwartung von eben — stimmt sie?

- Ein **Primärschlüssel** ist eindeutig in seiner Tabelle, ein **Fremdschlüssel** verweist darauf — so bleibt jede Information an genau einer Stelle.
- **Redundanz** ist keine Bequemlichkeit, sondern ein Risiko: Jede Kopie kann bei der nächsten Änderung vergessen werden.
- Und selbst eine einzelne Stelle nützt nichts, wenn sie nicht **eindeutig** ist — Monika Müllers zwei Schreibweisen zeigen, was dann passiert.
- Vor dem **Verbinden** steht die **Erwartung** an die Zeilenzahl — das Frühwarnsystem für die Lernwerkstatt.

Nächste Woche: Wie liest man eine Datenbank? Das Modell von Brick-Flow, und warum nicht jede Bestellung eine Kampagne hat.

- **DAMA International** — *DAMA-DMBOK: Data Management Body of Knowledge* (2. Aufl., 2017): Kap. Data Modeling and Design (Primär-/Fremdschlüssel, 1:n-Beziehungen).
- **Brick-Flow AG** — Fallunternehmen des Kurses, `https://brick-flow.ag/`. Die Beispiele (Heike Meier, Monika Müller, die Zahlen 1.616/550/1.625) sind reale Auszüge aus den Kursdatensätzen (`use-case/data/`), dokumentiert in `use-case/BRICKFLOW.md` §9.6–9.10.