



Datenbanken und SQL

(Woche 3 - Tag 2)

Agenda

Weitere JOIN-Varianten

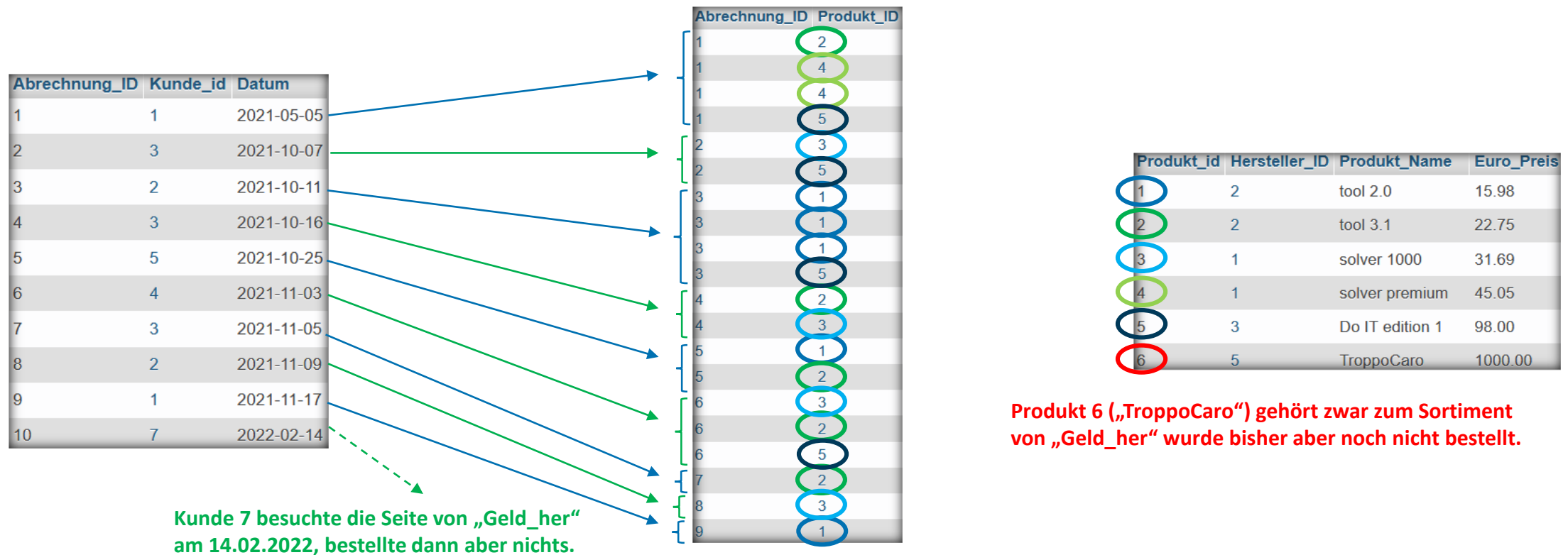
- Der bisherige JOIN
 - Beispielaufgabe + Vorbemerkungen
 - Bisherige und **alternative** Syntax
 - Ausgabetabelle
 - Motivation der ON-Klausel
- LEFT JOIN
 - Definition + (erste) Motivation
 - Beispielaufgabe
 - Syntax + Ausgabetabelle
- RIGHT JOIN
 - Definition + (erste) Motivation
 - Beispielaufgabe
 - Syntax + Ausgabetabelle
- FULL JOIN
 - Definition + Ausblick

JOIN-Varianten

Beispielaufgabe + Vorbemerkungen

Aufgabenstellung:

Für jeden Einkauf eines einzelnen Exemplars soll das **Einkaufsdatum** und der **Produktname** ausgegeben werden.



Produkt 6 („TropoCaro“) gehört zwar zum Sortiment von „Geld_her“ wurde bisher aber noch nicht bestellt.

Bisherige und **alternative** Syntax

Aufgabenstellung:

Für jeden Einkauf eines einzelnen Exemplars soll das **Einkaufsdatum** und der **Produktname** ausgegeben werden.

- Bisherige Lösung:

```
SELECT Datum, Produkt_Name  
FROM Abrechnung, Abrechnung_Produkt, Produkt  
WHERE Abrechnung.Abrechnung_ID = Abrechnung_Produkt.Abrechnung_ID  
AND Abrechnung_Produkt.Produkt_ID = Produkt.Produkt_id;
```

- Alternative Lösung:

```
SELECT Datum, Produkt_Name  
FROM Abrechnung INNER JOIN Abrechnung_Produkt ON Abrechnung.Abrechnung_ID = Abrechnung_Produkt.Abrechnung_ID  
      INNER JOIN Produkt ON Abrechnung_Produkt.Produkt_ID = Produkt.Produkt_id;
```

Ausgabetabelle

```
SELECT Datum, Produkt_Name  
FROM Abrechnung INNER JOIN Abrechnung_Produkt ON Abrechnung.Abrechnung_ID = Abrechnung_Produkt.Abrechnung_ID  
INNER JOIN Produkt ON Abrechnung_Produkt.Produkt_ID = Produkt.Produkt_id;
```

Datum	Produkt_Name
2021-10-11	tool 2.0
2021-10-11	tool 2.0
2021-10-11	tool 2.0
2021-10-25	tool 2.0
2021-11-17	tool 2.0
2021-05-05	tool 3.1
2021-10-16	tool 3.1
2021-10-25	tool 3.1
2021-11-03	tool 3.1
2021-11-05	tool 3.1
2021-10-07	solver 1000
2021-10-16	solver 1000
2021-11-03	solver 1000
2021-11-09	solver 1000
2021-05-05	solver premium
2021-05-05	solver premium
2021-05-05	Do IT edition 1
2021-10-07	Do IT edition 1
2021-10-11	Do IT edition 1
2021-11-03	Do IT edition 1

Es fehlt das Kalenderdatum 14.02.2022,
da an diesem Tag nichts bestellt wurde.

Es fehlt das Produkt „TroppoCaro“,
weil dieses nie bestellt wurde.

Motivation der ON-Klausel

- Ein erstes „weiches“ Argument ist eher **ästhetischer** Natur:
Die „technischen“ Schlüssel-Bedingungen können in der ON-Klausel berücksichtigt werden, während die „inhaltlichen“ Bedingungen (also solche, die seitens der **Aufgabenstellung** gefordert werden) weiterhin mittels der WHERE Klausel abgearbeitet werden (könnten).
- Ein stärkeres Argument ist die (potentiell) **verbesserte Performance** (hängt aber von der internen Nutzung eines Optimizers ab, der den Performance-Unterschied aufheben würde), **da** die Verwendung der ON-Klausel die Anzahl der Kombinationen verringert:

„FROM Abrechnung,Abrechnung_Produkt,Produkt“ erzeugt einen JOIN mit:

$10(\text{Abrechnungen}) \times 20(\text{Einkaufsposten}) \times 6(\text{Produkte}) = 1200 \text{ Kombinationen}$

„Abrechnung INNER JOIN Abrechnung_Produkt ON ...“ erzeugt einen JOIN mit:

$10(\text{Abrechnungen}) \times 20(\text{Einkaufsposten}) = 200 \text{ Kombinationen, ...}$

... die dann aber mittels ON-Klausel sofort auf 20 Datensätze gefiltert werden ...

Das anschließende „... INNER JOIN Produkt ON ...“ erzeugt dann einen JOIN mit:

$20 \text{ Datensätze} \times 6(\text{Produkte}) = 120 \text{ Kombinationen}$

LEFT JOIN

Definition und (erste) Motivation

- Der **LEFT JOIN** erzwingt die Ausgabe von Datensätzen (der im JOIN links notierten Tabelle) **AUCH** wenn diese Datensätze nicht mit Datensätzen der anderen (rechten) Tabelle verknüpft werden können.
- Was damit gemeint ist, werden wir uns mit Hilfe des folgenden **Beispiels** (das eine Variante der zuvor betrachteten Beispielaufgabe darstellt) dann aber noch genauer anschauen können.
- Auch die **Motivation** für eine solche Vorgehensweise werden wir uns mit Hilfe dieses Beispiels verdeutlichen können.
- Wenn wir dabei von einer „**ersten**“ **Motivation** sprechen, dann deswegen, weil wir uns in den kommenden Tagen mit mathematischen Funktionen beschäftigen wollen, die uns weitere Gründe für den Einsatz eines LEFT JOINS liefern werden.

Beispielaufgabe

Erneut gilt: Für jeden Einkauf eines einzelnen Exemplars soll das **Einkaufsdatum** und der **Produktname** ausgegeben werden. Nun aber sollen **auch Kalenderdaten** ausgegeben werden, an denen die Seite von „Geld_her“ zwar besucht, aber **nichts bestellt** wurde.

(Produktname sollen jedoch weiterhin nur (zusammen mit dem Einkaufsdatum) ausgegeben werden, falls diese auch tatsächlich gekauft wurden.)

Abrechnung_ID	Kunde_id	Datum
1	1	2021-05-05
2	3	2021-10-07
3	2	2021-10-11
4	3	2021-10-16
5	5	2021-10-25
6	4	2021-11-03
7	3	2021-11-05
8	2	2021-11-09
9	1	2021-11-17
10	7	2022-02-14

Das Kalenderdatum „14.02.2022“ soll ausgegeben werden, obwohl diesem kein Produktname zugeordnet werden kann.

Abrechnung_ID	Produkt_ID
1	2
1	4
1	4
1	5
2	3
2	5
3	1
3	1
3	1
3	5
4	2
4	3
5	1
5	2
6	3
6	2
6	5
7	2
8	3
9	1

Produkt_id	Hersteller_ID	Produkt_Name	Euro_Preis
1	2	tool 2.0	15.98
2	2	tool 3.1	22.75
3	1	solver 1000	31.69
4	1	solver premium	45.05
5	3	Do IT edition 1	98.00
6	5	TropoCaro	1000.00

Das Produkt „TropoCaro“ soll nicht ausgegeben werden, da dieses bisher noch nicht bestellt wurde.

Syntax und Ausgabetabelle

```
SELECT Datum, Produkt_Name
FROM Abrechnung LEFT JOIN Abrechnung_Produkt
    ON Abrechnung.Abrechnung_ID = Abrechnung_Produkt.Abrechnung_ID
LEFT JOIN Produkt
    ON Abrechnung_Produkt.Produkt_ID = Produkt.Produkt_id;
```

Datum	Produkt_Name
2021-05-05	tool 3.1
2021-05-05	solver premium
2021-05-05	solver premium
2021-05-05	Do IT edition 1
2021-10-07	solver 1000
2021-10-07	Do IT edition 1
2021-10-11	tool 2.0
2021-10-11	tool 2.0
2021-10-11	tool 2.0
2021-10-11	Do IT edition 1
2021-10-16	tool 3.1
2021-10-16	solver 1000
2021-10-25	tool 2.0
2021-10-25	tool 3.1
2021-11-03	solver 1000
2021-11-03	tool 3.1
2021-11-03	Do IT edition 1
2021-11-05	tool 3.1
2021-11-09	solver 1000
2021-11-17	tool 2.0
2022-02-14	NULL

RIGHT JOIN

Definition und (erste) Motivation

- Der **RIGHT JOIN** erzwingt die Ausgabe von Datensätzen (der im JOIN rechts notierten Tabelle) **AUCH** wenn diese Datensätze nicht mit Datensätzen der anderen (linken) Tabelle verknüpft werden können.
- Damit wird deutlich, dass es sich beim RIGHT JOIN quasi um die „**gespiegelte**“ Version des LEFT JOIN handelt, was erneut am kommenden Beispiel deutlich werden wird.
- Für die **Motivation** gilt entsprechendes.

Beispielaufgabe

Erneut gilt: Für jeden Einkauf eines einzelnen Exemplars soll das **Einkaufsdatum** und der **Produktname** ausgegeben werden.
Nun aber sollen **auch Produktname** ausgegeben werden, die **nie bestellt** wurden.

(Kalenderdaten sollen hingegen nur (zusammen mit dem Produktname) ausgegeben werden, falls an diesem Tag dieses Produkt tatsächlich gekauft wurde.)

Abrechnung_ID	Kunde_id	Datum
1	1	2021-05-05
2	3	2021-10-07
3	2	2021-10-11
4	3	2021-10-16
5	5	2021-10-25
6	4	2021-11-03
7	3	2021-11-05
8	2	2021-11-09
9	1	2021-11-17
10	7	2022-02-14

Das Kalenderdatum „14.02.2022“ soll nicht ausgegeben werden, da diesem Datum kein Produktname zugeordnet werden kann.

Abrechnung_ID	Produkt_ID
1	2
1	4
1	4
1	5
2	3
2	5
3	1
3	1
3	1
3	5
4	2
4	3
5	1
5	2
6	3
6	2
6	5
7	2
8	3
9	1

Produkt_id	Hersteller_ID	Produkt_Name	Euro_Preis
1	2	tool 2.0	15.98
2	2	tool 3.1	22.75
3	1	solver 1000	31.69
4	1	solver premium	45.05
5	3	Do IT edition 1	98.00
6	5	TropoCaro	1000.00

Das Produkt „TropoCaro“ soll ausgegeben werden, obwohl es bisher noch nicht bestellt wurde.

Syntax und Ausgabetabelle

```
SELECT Datum, Produkt_Name
FROM Abrechnung RIGHT JOIN Abrechnung_Produkt
    ON Abrechnung.Abrechnung_ID = Abrechnung_Produkt.Abrechnung_ID
    RIGHT JOIN Produkt
    ON Abrechnung_Produkt.Produkt_ID = Produkt.Produkt_id;
```


Datum	Produkt_Name
2021-10-11	tool 2.0
2021-10-11	tool 2.0
2021-10-11	tool 2.0
2021-10-25	tool 2.0
2021-11-17	tool 2.0
2021-05-05	tool 3.1
2021-10-16	tool 3.1
2021-10-25	tool 3.1
2021-11-03	tool 3.1
2021-11-05	tool 3.1
2021-10-07	solver 1000
2021-10-16	solver 1000
2021-11-03	solver 1000
2021-11-09	solver 1000
2021-05-05	solver premium
2021-05-05	solver premium
2021-05-05	Do IT edition 1
2021-10-07	Do IT edition 1
2021-10-11	Do IT edition 1
2021-11-03	Do IT edition 1
NULL	TropoCaro

FULL JOIN

Definition und Ausblick

- Der **FULL JOIN** würde - bezogen auf unsere Beispielaufgabe – neben jedem Einkauf (Ausgabe: Datum und Produktname) ebenfalls auch jene Kalenderdaten ausgeben, an denen nichts bestellt wurde UND ebenso auch die Namen jener Produkte, die bisher noch nicht bestellt wurden.
- Der FULL JOIN kann also als eine „**Vereinigung**“ von LEFT und RIGHT JOIN betrachtet werden. Genau dies wird in einigen Tagen dann auch unser Lösungsansatz sein.
- Aktuell werden wir den FULL JOIN allerdings noch nicht darstellen können, da MySQL diesen **nicht explizit** anbietet, sondern ihn nur über einen Umweg ermöglicht.

Gemeinsame Übung („Live-Coding“) -> A_03_02_01



Aufgabe_03_02_01

Implementieren Sie bitte zunächst den Code aus dem Dokument **A_03_02_01.sql**. Studieren Sie bitte anschließend die dort notierten Bemerkungen, insbesondere, um jene Datensätze der Tabellen Mitarbeiter, Auto und Parkplatz kennenzulernen, die keinem Datensatz einer anderen Tabelle zugeordnet werden können.

Führen Sie bitte anschließend alle folgenden Varianten durch:

```
SELECT
  mitarbeiter.mitarbeiter_bezeichnung,
  fahrzeug.fahrzeug_bezeichnung,
  parkplatz.parkplatz_bezeichnung
Mitarbeiter (INNER, LEFT, RIGHT) JOIN Fahrzeug ON ...
(INNER, LEFT, RIGHT) JOIN Parkplatz ON ...;
```

Betrachten Sie hierfür bitte zunächst die Zwischenlösung von:

```
SELECT
  mitarbeiter.mitarbeiter_bezeichnung,
  fahrzeug.fahrzeug_bezeichnung,
  parkplatz.parkplatz_bezeichnung
Mitarbeiter (INNER, LEFT, RIGHT) JOIN Fahrzeug ON ...;
```

und ergänzend Sie bitte erst anschließend durch:


```
... (INNER, LEFT, RIGHT) JOIN Parkplatz
```

Versuchen Sie bitte zunächst in allen Fällen theoretisch zu durchdenken, welche Ausgabe erscheinen wird, bevor Sie den jeweiligen Befehl tatsächlich ausführen lassen und erläutern Sie bitte das tatsächliche Ergebnis schriftlich.

WBS TRAINING AG
Lorenzweg 5
D-12099 Berlin
Amtsgericht Berlin HRB 68531
Sitz der Gesellschaft: Berlin

Vorstand:
Heinrich Kronbichler,
Joachim Giese
Aufsichtsrat (Vorsitz): Dr. Daniel Stadler
USt-IdNr.: DE 209 768 248

GLS Gemeinschaftsbank eG
IBAN: DE18 4305 0967 1146 1814 00
BIC: GENODEM33GLS


GLS zertifiziert nach
DIN EN ISO 9001:2015 und ISO 14001:2015
Zertifizierung nach ISO 26001:2017
nach ISO 26001:2017

Vielen Dank für Ihre Aufmerksamkeit!

