

### Aufgabe 1 – Abfragen in SQL

Finden Sie bitte für alle Aufgaben zwei strukturell unterschiedliche Lösungen. Beispielsweise nutzt dann eine Lösung mit Verbundoperationen (Joins) und die andere Lösung Unterabfragen (Subqueries).

Geben Sie die zugehörigen SQL-Abfragen an und führen Sie diese auf der Datenbank aus:

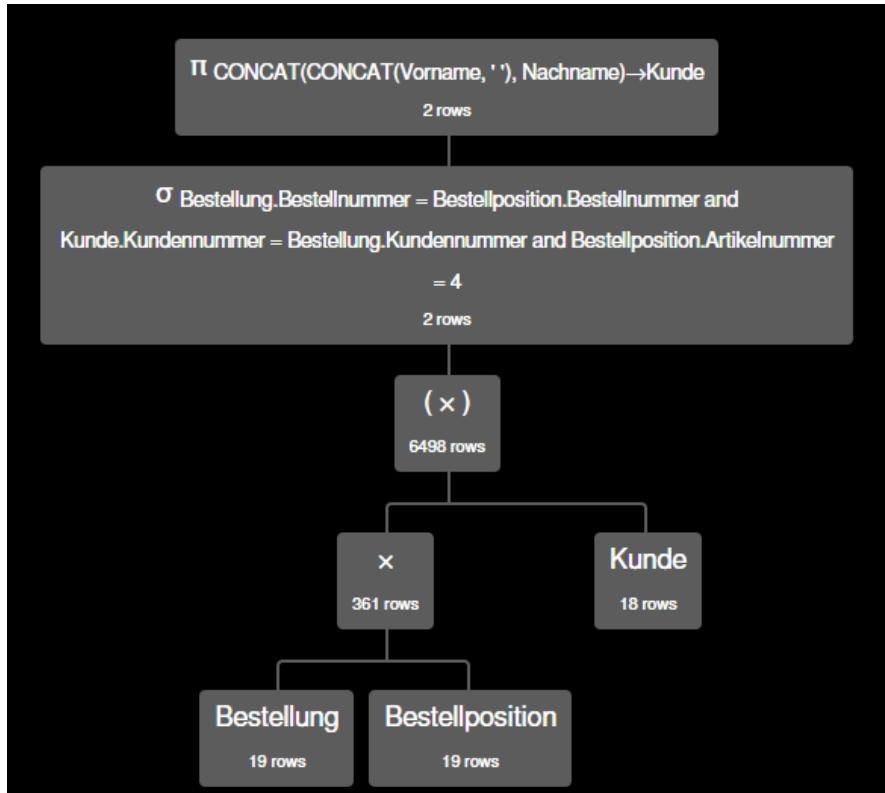
Kunde
A. S. S. Assin
Amber Love

- a) Listen Sie alle Kunden, die Artikel 4 bestellt haben mit den Angaben Vorname und Nachname. Die Ausgabe soll in folgender Form erfolgen:

#### Variante 1 – Verbundoperation (Cross Join):

SQL	RelaX-Trainer
<pre>SELECT Vorname    ' '    Nachname as Kunde FROM Bestellung b, Bestellposition p, Kunde k WHERE b.Bestellnummer=p.Bestellnummer AND k.Kundennummer=b.Kundennummer AND p.Artikelnummer=4;</pre>	$\pi \text{ concat(concat(Vorname, ' '), Nachname)} \rightarrow \text{Kunde}$ $\sigma \text{ Bestellung.Bestellnummer} = \text{Bestellposition.Bestellnummer} \text{ and }$ $\text{Kunde.Kundennummer} = \text{Bestellung.Kundennummer} \text{ and }$ $\text{Bestellposition.Artikelnummer} = 4 \text{ (Bestellung x Bestellposition x Kunde)}$

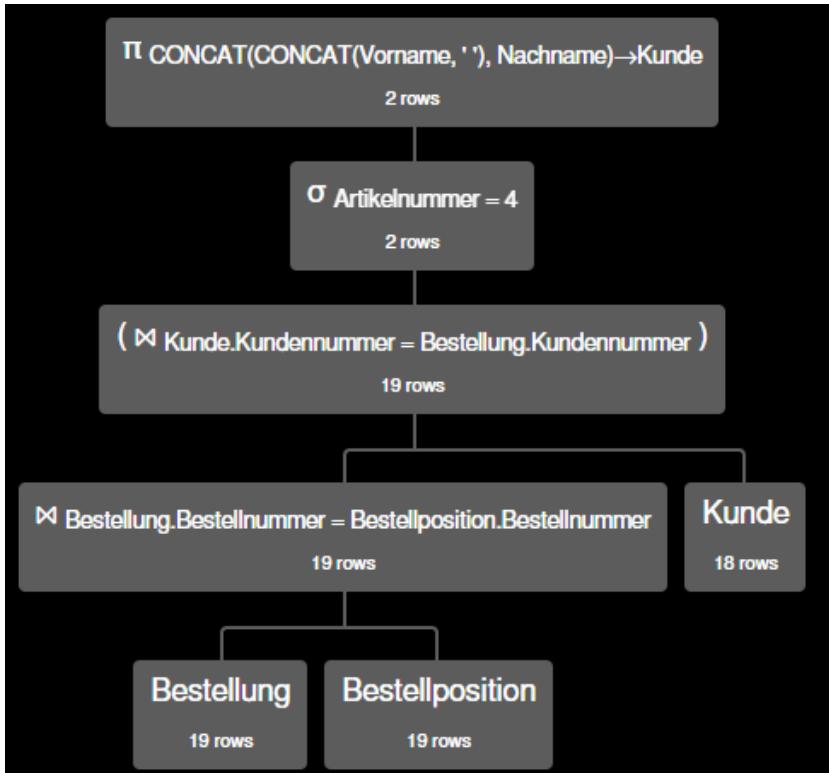
#### Screenshot aus dem RelaX-Trainer:



#### Variante 2 – Verbundoperation:

SQL	RelaX-Trainer
<pre>SELECT Vorname    ' '    Nachname AS Kunde FROM Bestellung b JOIN Bestellposition bp ON b.Bestellnummer=bp.Bestellnummer JOIN Kunde k on k.Kundennummer=b.Kundennummer WHERE Artikelnummer=4;</pre>	$\pi \text{ concat(concat(Vorname, ' '), Nachname) } \rightarrow \text{Kunde}$ $\sigma \text{ Artikelnummer} = 4$ $(\bowtie \text{ Kunde.Kundennummer} = \text{Bestellung.Kundennummer})$ $\bowtie \text{ Bestellung.Bestellnummer} = \text{Bestellposition.Bestellnummer}$ $\text{Kunde}$

Screenshot aus dem RelaX-Trainer:

**Variante 3 – Unterabfrage:**

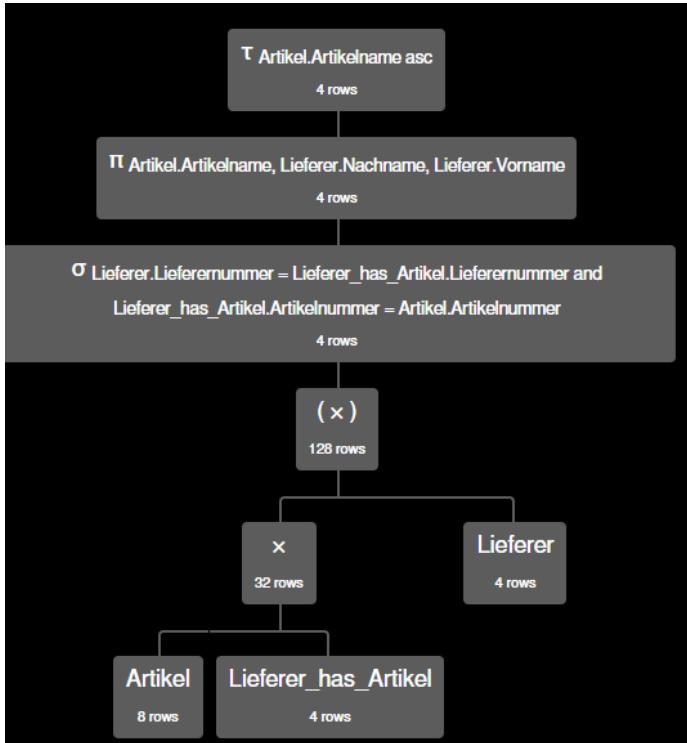
```
Select Vorname || ' ' || Nachname AS Kunde
FROM Kunde
WHERE Kundennummer IN
(SELECT DISTINCT Kundennummer FROM Bestellung NATURAL JOIN Bestellposition
WHERE Artikelnummer=4);
```

- b) Listen Sie alle Lieferer, sortiert nach den Artikelbezeichnungen der Artikel die sie liefern können mit Angaben von Vorname, Nachname und Name des Artikels.

**Variante 1 – Verbundoperation (Cross Join):**

SQL	RelaX-Trainer
<pre>SELECT Vorname, Nachname, Artikelname FROM Lieferer l, Lieferer_has_artikel lha, Artikel a WHERE l.Lieferernummer=lha.Lieferernummer AND a.artikelnummer=lha.artikelnummer ORDER BY a.Artikelname;</pre>	$\tau \text{ Artikel.Artikelname asc } \pi \text{ Artikel.Artikelname, Lieferer.Nachname, Lieferer.Vorname } \sigma \text{ Lieferer.Lieferernummer} = \text{Lieferer_has_Artikel.Lieferernummer and Lieferer_has_Artikel.Artikelnummer} = \text{Artikel.Artikelnummer }$ $(\text{Artikel x Lieferer_has_Artikel x Lieferer})$

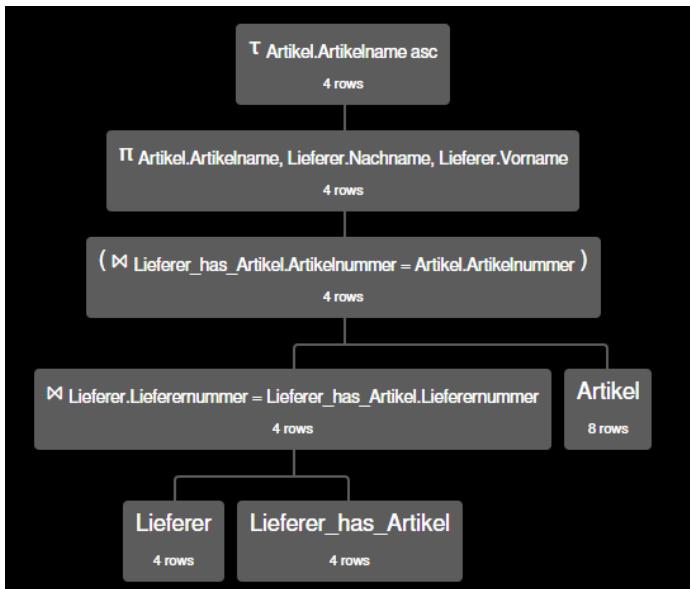
Screenshot aus dem RelaX-Trainer:



## Variante 2 – Verbundoperation:

SQL	RelaX-Trainer
<pre>select vorname, nachname, artikelname from artikel natural join lieferer_has_artikel natural join lieferer order by artikelname;</pre>	$\tau_{\text{Artikel.Artikelname asc}} \pi_{\text{Artikel.Artikelname, Lieferer.Nachname, Lieferer.Vorname}} (\text{Lieferer} \bowtie \text{Lieferer.Lieferernummer} = \text{Lieferer\_has\_Artikel.Lieferernummer} \text{ Lieferer\_has\_Artikel} \bowtie \text{Lieferer\_has\_Artikel.Artikelnummer} = \text{Artikel.Artikelnummer Artikel})$

## Screenshot aus dem RelaX-Trainer:



# Datenbanken 1

# Praktikum 8

## Variante 3 – Unterabfrage:

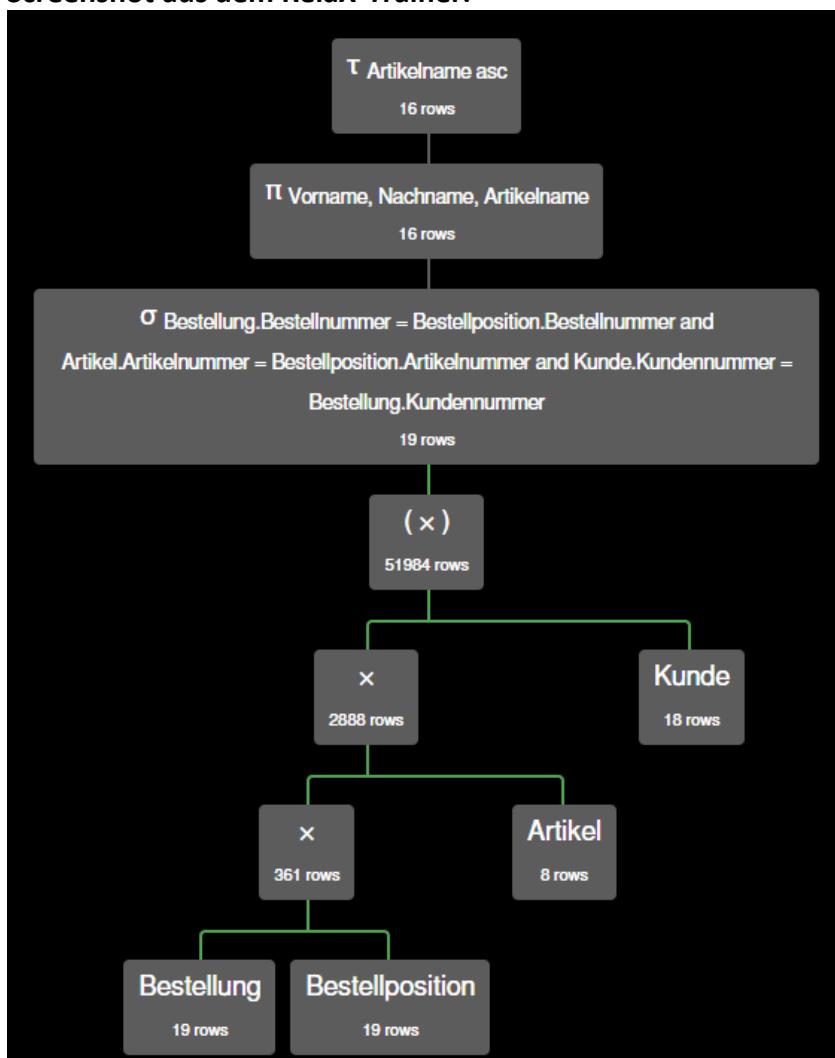
```
SELECT Vorname, Nachname, Artikelname
FROM Lieferer Natural JOIN
(SELECT Artikelname, lieferernummer
FROM lieferer_has_artikel lha NATURAL JOIN Artikel a) la
ORDER BY artikelname;
```

- c) Listen Sie alle Kunden, die Bestellungen haben. Listen Sie neben dem Vornamen und dem Nachnamen auch die Artikelbezeichnung der bestellten Artikel sortiert nach der Artikelbezeichnung.

## Variante 1 – Verbundoperation (Cross Join):

SQL	RelaX-Trainer
<pre>SELECT Vorname, Nachname, Artikelname FROM Bestellung b, Bestellposition p, Artikel a, Kunde k WHERE b.Bestellnummer=p.Bestellnummer AND a.Artikelnummer=p.Artikelnummer AND k.Kundennummer=b.Kundennummer ORDER BY Artikelname;</pre>	$\tau_{\text{Artikelname asc}}$ $\pi_{\text{Vorname, Nachname, Artikelname}}$ $\sigma_{\text{Bestellung.Bestellnummer} = \text{Bestellposition.Bestellnummer} \text{ and } \text{Artikel.Artikelnummer} = \text{Bestellposition.Artikelnummer} \text{ and } \text{Kunde.Kundennummer} = \text{Bestellung.Kundennummer}}$ (Bestellung $\times$ Bestellposition $\times$ Artikel $\times$ Kunde)

## Screenshot aus dem RelaX-Trainer:



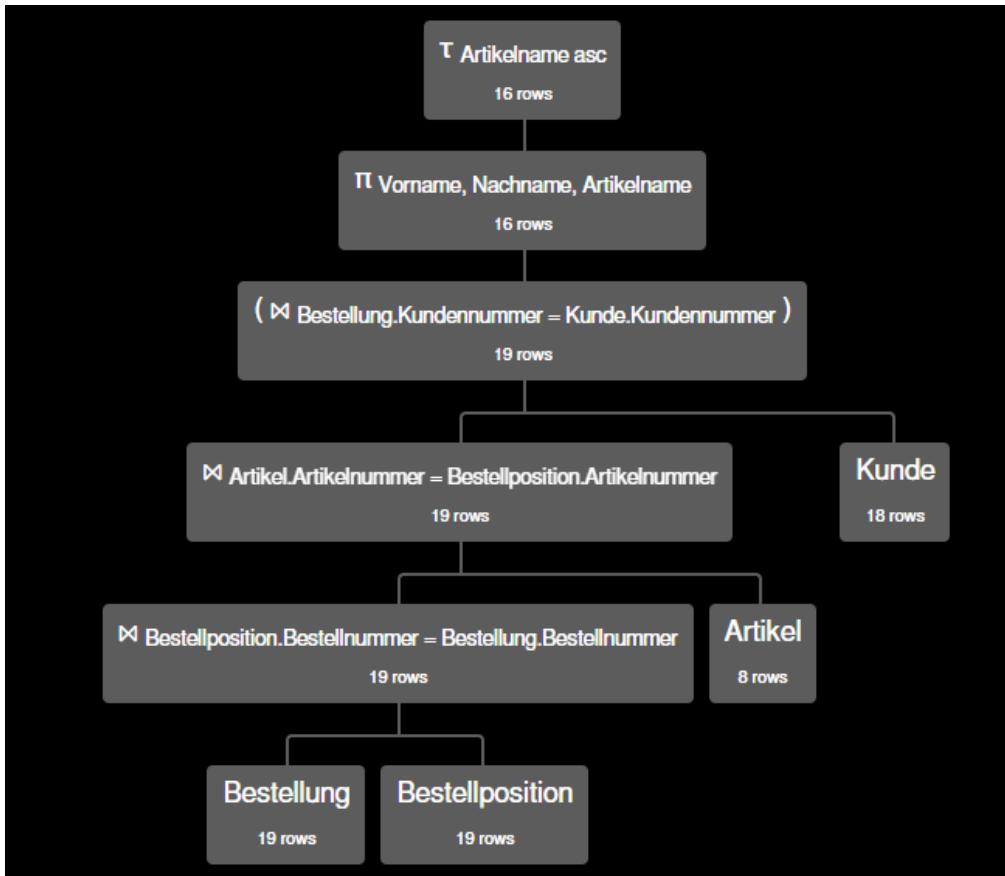
## Datenbanken 1

## Praktikum 8

### Variante 2 – Verbundoperation:

SQL	RelaX-Trainer
<pre>SELECT Vorname, Nachname, Artikelname FROM Bestellung b JOIN Bestellposition bp ON bp.Bestellnummer=b.Bestellnummer JOIN Artikel a on a.Artikelnummer=bp.Artikelnummer JOIN Kunde k on b.Kundennummer=k.Kundennummer ORDER BY Artikelname;</pre>	$\tau_{\text{Artikelname asc}} \pi_{\text{Vorname, Nachname, Artikelname}}$ $(\bowtie_{\text{Bestellung.Kundennummer} = \text{Kunde.Kundennummer}})$ $\bowtie_{\text{Artikel.Artikelnummer} = \text{Bestellposition.Artikelnummer}}$ $\bowtie_{\text{Bestellposition.Bestellnummer} = \text{Bestellung.Bestellnummer}}$ $\text{Bestellung}$ 19 rows $\text{Bestellposition}$ 19 rows $\text{Kunde}$ 18 rows $\text{Artikel}$ 8 rows

### Screenshot aus dem RelaX-Trainer:



### Variante 3 – Unterabfrage:

```

SELECT (SELECT Vorname FROM Kunde WHERE Kundennummer = b.Kundennummer),
(SELECT Nachname FROM Kunde WHERE Kundennummer = b.Kundennummer),
(SELECT Artikelname FROM Artikel WHERE Artikelnummer=bp.Artikelnummer) Artikelname
FROM Bestellposition bp JOIN Bestellung b USING (Bestellnummer)
ORDER BY Artikelname;
  
```