

## 9. Datenabfrage mit mehreren Relationen

Um Redundanz in einer Datenbank zu vermeiden, müssen die Relationen normalisiert werden. Sachlich zusammengehörige Informationen werden dadurch in mehrere Basisrelationen zerlegt. Um diese aber wieder ordnungsgemäß zusammenzusetzen (z.B. welcher Kunde gehört zu welcher Bestellung), sind oft Abfragen nötig, die gleichzeitig mehrere Relationen betreffen.

Bei einem Verbund von Relationen, englisch Join, verbinden wir Tupel verschiedener Relationen zu einem zusammenhängenden Tupel. Die Ergebnisrelation umfasst Attribute aus den beteiligten Basistabellen.

### 9.1. Die Verwendung von Aliasnamen für Relationen

Wenn in einer Abfrage mehrere Relationen auftreten oder eine Relation mehrfach (was auch möglich ist), ist es erforderlich, die benutzten Attribute den entsprechenden Relationen zuzuordnen. Dazu kann grundsätzlich bei der Einführung einer Relation in der FROM-Klausel hinter die Relation ein Aliasname für die Relation geschrieben werden.

Die Syntax ist:

Relation AS Aliasname

#### Beispiele

T\_Kunden AS k

Falls kein Alias angegeben wird, ist standardmäßig der Name der Relation der Alias – das geht allerdings nicht, wenn eine Tabelle in einer Abfrage mehrfach verwendet wird. Ein Bezug auf ein Attribut der entsprechenden Relation erfolgt durch:

Aliasname.Attribut

Der Alias darf nur dann entfallen, wenn der Name des Attributs eindeutig ist, d.h. in genau einer der benutzten Relationen auftritt. Wir empfehlen bei komplexen Abfragen mit mehreren Relationen grundsätzlich alle Attribute mit Aliasnamen zu benutzen, da das auch der Übersichtlichkeit dient – wir können dann sofort erkennen, zu welcher Relation ein entsprechendes Attribut gehört.

Der Verbund ist nicht nur zwischen zwei Relationen, sondern zwischen beliebig vielen Relationen möglich. Nur bei den Varianten des Outer Join spielt die Reihenfolge, in der die Relationen bei der FROM-Klausel aufgezählt werden, eine Rolle. Des Weiteren sind in der Regel Klammern zu setzen, wenn verschiedene Arten des Join in einer Anweisung gemeinsam auftreten.

Durch die Angabe einer Attributliste hinter der SELECT -Klausel wird die Verbundoperation mit einer Projektion kombiniert. In den folgenden Beispielen nutzen wir dies aus, um die Ergebnislisten übersichtlicher zu gestalten.

## 9.2. Der innere Verbund (Inner Join)

### 9.2.1. Cross Join (Kreuzprodukt oder kartesisches Produkt)

Es werden alle Tupel der einen Relation mit allen Tupeln der zweiten Relation verbunden, das folgende Beispiel zeigt das Prinzip, wenn es auch inhaltlich wenig sinnvoll ist.

#### Beispiel

```
SELECT K.Kunden_Nr, K.Vorname, K.Nachname, G.Kontoinhaber, G.BLZ, G.Konto_Nr
FROM T_Kunden AS K, T_Girokonten AS G;
```

### 9.2.2. Condition Join

Bei dieser Variante wird eine Bedingung angegeben, die in einem Wertvergleich zwischen Attributen der einen und Attributen der anderen Tabelle besteht. Die Attribute die verglichen werden sind dabei beliebig, vor allem die Fremd- und Primärschlüssel der Relationen spielen aber eine große Rolle. Der Vergleichsoperator ist im Prinzip auch beliebig. In der Mehrzahl der Fälle wird aber die Gleichheit geprüft.

Das folgende Beispiel zeigt die Kontoinformationen der Kunden an.

#### Beispiel

```
SELECT K.KundenNr, K.Vorname, K.Nachname, G.Kontoinhaber, G.BLZ, G.KontoNr
FROM T_Kunde AS K INNER JOIN T_Girokonto AS G ON K.KundenNr = G.F_KundenNr;
```

Der Condition Join bietet die flexibelste Möglichkeit, Tabellen zu verbinden. Bis zum Standard SQL-89 gab es keine eigenständige JOIN-Klausel, sondern es waren alle zu verknüpfenden Tabellen (in beliebiger Reihenfolge) in der FROM-Klausel anzugeben, gegebenenfalls zusätzlich mit Alias. Die Verknüpfung erfolgte dann mit denselben Ausdrücken wie in der ON-Klausel, aber sie war in der WHERE-Klausel mit unterzubringen, eventuell durch AND mit anderen Join-Verknüpfungen und Selektionskriterien gemischt. Diese Art des Verbunds ist aus Kompatibilitätsgründen noch zulässig und wird als Old Style Join bezeichnet.

Das oben angegebene Beispiel sieht mit dem Old Style Join wie folgt aus:

#### Beispiel

```
SELECT K.KundenNr, K.Vorname, K.Nachname, G.Kontoinhaber, G.BLZ, G.KontoNr
FROM T_Kunde AS K, T_Girokonto AS G
WHERE K.KundenNr = G.F_KundenNr;
```

### Verbund von mehr als zwei Relationen

Den Verbund von mehr als zwei Tabellen demonstrieren wir anhand der folgenden Beispiele.

**Beispiel:** Welcher Kunde hat eine Bestellung abgegeben? (2 Relationen)

```
SELECT K.Nachname, B.BestellNr
FROM T_Kunde AS K INNER JOIN T_Bestellung AS B ON K.KundenNr = B.F_KundenNr;
```

**Beispiel:** Welcher Kunde hat welche Stückzahl bestellt? (3 Relationen)

```
SELECT K.Nachname, B.BestellNr, P.BestellMenge
FROM T_Kunde AS K
INNER JOIN T_Bestellung AS B ON K.KundenNr = B.F_KundenNr
INNER JOIN T_Position AS P ON B.BestellNr = P.F_BestellNr ;
```

---

---

**Beispiel:** Welcher Kunde hat wie viele spezielle Artikel bestellt? (4 Relationen)

```
SELECT K.Nachname, B.BestellNr, P.BestellMenge, A.Bezeichnung
FROM   T_Kunde AS K
INNER JOIN T_Bestellung AS B ON K.KundenNr = B.F_KundenNr
INNER JOIN T_Position   AS P ON B.BestellNr = P.F_BestellNr
INNER JOIN T_Artikel     AS A ON A.ArtikelNr = P.F_ArtikelNr;
```

**A09-1**

Zeigen Sie zu jedem Artikel den Mehrwertsteuersatz an!

(Ausgabe: Bezeichnung, Mwst, Prozent)

**A09-2**

Zeigen Sie mit Hilfe des Mehrwertsteuersatzes den Bruttopreis der Artikel an!

(Ausgabe: Bezeichnung, Bruttopreis)

**A09-3**

Zeigen Sie zu jeder Bestellung den Wohnort des Kunden an.

(Ausgabe: Bestellnummer, Bestelldatum, Ort)

**A09-4**

Zeigen Sie zu jeder Bestellposition die Bestellmenge, die Bestellnummer und den Wohnort des Kunden an!

(Ausgabe: Bestellnummer, Artikelnummer, Bestellmenge, Ort)

**A09-5**

Zeigen Sie zu jedem Ort die Anzahl der bestellten Artikel an!

(Ausgabe: Bestellmenge, Ort)

**A09-6**

Listen Sie die Artikel mit ihrer zugehörigen Bestellmenge auf.

(Ausgabe: Bezeichnung, Artikelnummer, Bestellmenge)

**A09-7**

Listen Sie die Kunden mit ihrer zugehörigen Bankverbindung auf!

(Ausgabe: Vorname, Nachname, BLZ, Kontonummer, Kontoinhaber)

**A09-8**

Welche Artikel wurden wievielmals verkauft?

(Ausgabe: Bezeichnung, Artikelnummer, Liefermenge)

**A09-9**

Welche Artikel wurden per Bankeinzug bezahlt?

(Ausgabe: Bezeichnung, Artikelnummer, Zahlungsart)

**A09-10**

Welche und wie viele Artikel sind zwar bestellt aber stehen noch zur Lieferung aus?

(Ausgabe: Bezeichnung, Artikelnummer, Noch zu liefernde Artikel)

**A09-11**

Welche Bestellung hat welchen Gesamtumsatz (einschließlich MWSt) erbracht?

(Ausgabe: Vorname und Nachname des Kunden, Bestellnummer, Umsatz)