# Modul 06: JOINS: ANGABEN

*Systemvoraussetzungen: eine beliebige Version von SQL-Server und SSMS (Microsoft SQL Server Management Studio)*

*Tools: SQL-Server; SSMS (Microsoft SQL Server Management Studio)*

*Autor: Leonard Hlavin, Letzte Änderung: 19.04.2021*

Daten in der Datenbank sind in mehrere Tabellen aufgeteilt, um nach Möglichkeit Redundanz zu vermeiden. Viele Tabellen sind untereinander über sogenannte Schlüsselfelder miteinander verknüpft. Am Beispiel der Northwind-DB haben wir eine *Customers*-Tabelle und eine *Orders*-Tabelle. In der *Customers*-Tabelle gibt es eine *CustomerID*. Die *CustomerID* in der *Customers*-Tabelle ist der sogenannte *Primärschlüssel* oder *primary key*; von diesem sind alle anderen in der Tabelle abhängig, oder mit anderen Worten, wenn wir die Kundennummer kennen, können wir Kundeninformationen nachschlagen.

Auch in der *Orders*-Tabelle kommt die *CustomerID* vor, denn bei jeder Bestellung muss dabeistehen, welcher Kunde bestellt hat. Hier ist die *CustomerID* ein sogenannter *Fremdschlüssel* oder *foreign key*, also das Feld, über das eine Verknüpfung zur Customers-Tabelle besteht.

Brauchen wir nun die Kontaktinformationen oder den vollen Firmennamen des Kunden einer bestimmten Bestellung, verwenden wir einen sogenannten JOIN, um Zugriff auf Informationen von mehreren Tabellen zu haben.

Syntax:

SELECT t1.Spaltenname1

, t1.Spaltenname2

, t2.Spaltenname1

FROM Tabellenname1 AS t1 INNER JOIN Tabellenname2 AS t2 ON t1.SpalteA = t2.SpalteA

Wir unterscheiden grob in INNER und OUTER JOIN; OUTER JOIN wird noch unterteilt in LEFT JOIN oder RIGHT JOIN. Wenn wir nur Daten aus Tabellen haben wollen, bei denen es Übereinstimmungen gibt, verwenden wir einen INNER JOIN. Wenn wir auch solche Daten haben wollen, wo es in der anderen Tabelle keine Übereinstimmungen dazu gibt, verwenden wir einen OUTER JOIN (also entweder LEFT JOIN oder RIGHT JOIN).

Anders ausgedrückt: Wenn wir nur die Kunden brauchen, die schon Bestellungen getätigt haben, verwenden wir einen INNER JOIN (denn hier gibt es eine Übereinstimmung in der *Orders*-Tabelle, weil bei jeder Bestellung dort auch die *CustomerID* eingetragen wird).

Wollen wir hingegen auch die Kunden ausgeben, die noch nichts bestellt haben, brauchen wir einen OUTER JOIN, denn für diese Kunden gibt es keine Übereinstimmung in der *Orders*-Tabelle, da ihre *CustomerID* dort erst eingetragen wird, wenn sie schon eine Bestellung getätigt haben.

## 1. Übung: Einfache JOINS verwenden

Ziel: Einüben der Verwendung von JOINS

Aufgabenstellung:

* Geben Sie alle Kunden (Firmenname) aus Brasilien und deren Frachtkosten aus.
* Ein Bild, das Text, Tisch enthält.

  Automatisch generierte BeschreibungWählen Sie weitere nützliche Spalten, z.B. Bestellnummer und Kundennummer aus.

Ergebnis: 83 Zeilen

## 2. Übung: JOINS und Wildcards verwenden

Ziel: Einüben der Verwendung von JOINS; Verwendung von Wildcards

Aufgabenstellung:

* Wir wollen bestimmte Produkte nachbestellen:
* Geben Sie die Namen der Anbieter (Supplier), die Sauce verkaufen, aus. (CompanyName, ProductName, Ansprechperson, Telefonnummer).
* Ein Bild, das Text enthält.

  Automatisch generierte BeschreibungAchtung: Wie heißt das Produkt/die Produkte? Verschiedene Möglichkeiten? Verschaffen Sie sich erst einen kleinen Überblick über die Produkte in der Datenbank und decken Sie mehrere Möglichkeiten ab.

Ergebnis: je nach Abfrage 2 oder 3 Zeilen

## 3. Übung: JOINS, WHERE und Serverfunktionen verwenden

Ziel: Einüben der Verwendung von JOINS; Verwendung von WHERE und IN; Verwendung von Serverfunktionen

Aufgabenstellung:

* Angenommen, es gab Beschwerden bei den Bestellungen 10251, 10280, 10990 und 11000.
* Welcher Angestellte hat diese Bestellungen bearbeitet?
* Geben Sie Vor- und Nachname in einem Feld als FullName aus.

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Ergebnis: 4 Zeilen (nur die ausgewählten Bestellungen)

## 4. Übung: JOINS über mehrere Tabellen verwenden

Ziel: Einüben der Verwendung von JOINS über mehrere Tabellen; Verwendung von WHERE und Wildcards

Aufgabenstellung:

* Welche Kunden haben Chai Tee gekauft und wieviel (einzelne Bestellungen)?
* Ein Bild, das Text enthält.

  Automatisch generierte BeschreibungGeben Sie OrderID, CustomerID, CompanyName, ProductName und Quantity an.

Ergebnis: 38 Zeilen

## 5. Übung: JOINS über mehrere Tabellen, WHERE und Wildcards

Ziel: Einüben der Verwendung von JOINS über mehrere Tabellen; Verwendung von WHERE und Wildcards; Verwendung von ORDER BY

Aufgabenstellung:

* Suchen Sie alle Bestellungen, bei denen Bier verkauft wurde. Welcher Kunde? Wieviel? Welches Bier?
* Der Produktname kann „Bier“ oder „Lager“ enthalten oder mit „Ale“ enden.
* Ordnen Sie die Ausgabe nach Menge (absteigend: größte zuerst) und Kundenname (alphabetisch).

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ergebnis: 114 Zeilen

## 6. Übung: SELFJOIN

Ziel: Üben von SELFJOIN; Verknüpfung einer Tabelle mit sich selbst

Aufgabenstellung:

* Wer ist der Chef von wem?
* Ausgabe: Name Angestellter, ID Angestellter, Name Chef, ID vom Chef
* Vergeben Sie neue Spaltenüberschriften
* Geben Sie den Vor- und Nachnamen in einer Spalte an
* Auch Employees, die keinen Vorgesetzten haben, sollen mit ausgegeben werden.

TIPPS:

* Für diese Profiübung müssen wir so tun, als gäbe es zwei Employee-Tabellen: Eine Tabelle für Mitarbeiter, eine für Vorgesetzte.
* In diesem Fall müssen die Tabellen ein ALIAS bekommen („AS name“).
* Ein Bild, das Tisch enthält.

  Automatisch generierte BeschreibungDie Spalte ReportsTo enthält die EmployeeID des Vorgesetzten.

Ergebnis: 9 Zeilen (Anzahl aller Employees)

## 7. Übung: SELFJOIN

Ziel: Üben von SELFJOIN; Verknüpfung einer Tabelle mit sich selbst

Aufgabenstellung:

* Angenommen, wir wollen eine Liste von Mitfahrgelegenheiten erstellen (wer lebt aller in derselben Stadt und könnte daher bei wem mitfahren?)
* Geben Sie alle Kunden aus, die in der gleichen Stadt wohnen, wie ein anderer Kunde (nicht sie selbst).
* Ausgabe: Customer1, City1, Customer2, City2
* Vergeben Sie Spaltenüberschriften
* Ordnen Sie die Ausgabe alphabetisch nach Städten und Kunden

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungTIPP: Die Spalte City wäre nur einmal notwendig, die doppelte Ausgabe dient nur der Kontrolle (und könnte später entfernt werden).

Ergebnis: 88 Zeilen