

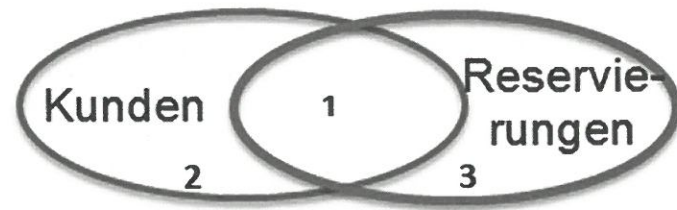
Das Ergebnis ist eine Ausgabe, die alle Kunden mit allen Verleihdaten kombiniert. Aus einer Tabelle mit 50 Einträgen und einer Tabelle mit 400 Einträgen wird somit eine Ergebnismenge mit 20'000 Einträgen erzeugt.

Die Anzahl der Einträge multipliziert sich mit der Anzahl der Datensätze jeder Tabelle, die man hinzufügt.

Arten von Joins

Am Beispiel der beiden Tabellen Kunden und Reservierungen werden die unterschiedlichen Joins erläutert. Es gibt

- Kunden, die Reservierungen in Auftrag gegeben haben (1),
- Kunden, die keine Reservierung beauftragt haben (2) und
- Reservierungen, die noch keinem Kunden zugeordnet sind (3).



Name	Darstellung	Erläuterung	Syntaxbeispiel
INNER JOIN, EQUI JOIN		Ausgabe von Kunden, die reserviert haben, mit den zugehörigen Reservierungen.	SELECT * FROM Kunden INNER JOIN Reservierungen ON Kunden.KdNr = Reservierungen.KdNr;
NATURAL JOIN		Ergebnis wie INNER JOIN	SELECT * FROM Kunden ,Reservierungen WHERE Kunden.KdNr = Reservierungen.KdNr;
LEFT JOIN,		Ausgabe aller Kunden und (falls vorhanden) die dazu gehörenden Reservierungen.	SELECT * FROM Kunden LEFT JOIN Reservierungen ON Kunden.KdNr = Reservierungen.KdNr;
RIGHT JOIN,		Ausgabe aller Reservierungen und – falls gespeichert – der entsprechenden Kunden.	SELECT * FROM Kunden RIGHT JOIN Reservierungen ON Kunden.KdNr = Reservierungen.KdNr;
LEFT OUTER JOIN		Ausgabe nur der Kunden, die nicht reserviert haben.	SELECT * FROM Kunden LEFT JOIN Reservierungen ON Kunden.KdNr = Reservierungen.KdNr WHERE Reservierungen.KdNr IS NULL;

Name	Darstellung	Erläuterung	Syntaxbeispiel
RIGHT OUTER JOIN		Ausgabe der Reservierungen, die keinem Kunden zugeordnet sind.	SELECT * FROM Kunden RIGHT JOIN Reservierungen ON Kunden.KdNr = Reservierungen.KdNr WHERE Reservierungen.KdNr IS NULL;
FULL OUTER JOIN		Ausgabe aller Kunden ohne Reservierung und aller nicht zugeordneten Reservierungen	SELECT * FROM Kunden FULL JOIN Reservierungen ON Kunden.KdNr = Reservierungen.KdNr WHERE Reservierungen.KdNr IS NULL; (Wird von ACCESS-SQL nicht unterstützt)

Inner-Joins, Equi-Joins, Natural Joins

Für Abfragen, die mehrere Tabellen betreffen, werden Verknüpfungen auf Gleichheit, Inner-Joins (auch: Equi-Joins equi = gleich, to join = verknüpfen;) verwendet. Im Beispiel dürfen nur die Kunden berücksichtigt werden, deren Kundennummer auch in der Tabelle Reservierungen aufgeführt ist. Dazu muss eine eindeutige Verknüpfungsvorschrift im SQL-Befehl definiert werden. Dies geschieht durch die Bedingung

WHERE Kunden.KdNr = Reservierungen.KdNr.

Die vollständige Anweisung lautet somit:

```
SELECT Kunden.KdNr, Kunden.Nachname, Kunden.Vorname, Reservierungen.AusleihDatum  
FROM Kunden, Reservierungen  
WHERE Kunden.KdNr=Reservierungen.KdNr;
```

Um gleichnamige Felder in den verschiedenen Tabellen zu unterscheiden, wird der Tabellename, gefolgt von einem Punkt, dem Spaltennamen vorangestellt, z. B. Kunden.KdNr. Die Verknüpfung (Joins) mehrerer Tabellen werden in einer SQL-Abfrage z. B. in Form einer WHERE-Bedingung beschrieben.

Beispiel:

Alle Reservierungen mit den Kundendaten (Kundennummer, Nachname, Vorname) und den Fahrraddaten (Fahrradnummer, Hersteller) werden ausgegeben. Die Anweisung lautet:

```
SELECT K.KdNr, Nachname, Vorname, Ort,  
       LeihDauer AS LDauer, F.Fahrradnummer AS FNr, Hersteller  
FROM Kunden AS K, Fahrraeder AS F, Reservierungen AS R  
WHERE K.KdNr=R.KdNr And F.Fahrradnummer=R.FRadNr;
```

In einem zusätzlichen Feld GesPreis wird der Gesamtleihpreis mit dem Ausdruck P.Preis*LDauer AS GesPreis berechnet und ausgegeben. Das Ergebnis lautet nun:

```
SELECT K.KdNr, Nachname, Vorname,  
       LeihDauer AS LDauer, F.Fahrradnummer AS FNr, Hersteller,  
       P.Preis*LDauer AS GesPreis  
FROM Kunden AS K, Preisgruppen AS P, Fahrraeder AS F,  
       Reservierungen AS R
```