

## SQL: Grundlegende Anweisungen im Überblick

**SELECT** *Fragt Tabellen oder Sichten ab und stellt Daten dar*

*Grundsyntax* `SELECT <Feldnamen | *> FROM <Tabelle>  
[WHERE <Filterbedingung>]`

*Beispiele* `SELECT KundeNr, Nachname, Vorname FROM Kunden  
WHERE Ort = 'Berlin'`  
`SELECT * FROM Artikel`

SELECT ruft Daten aus Tabellen oder Sichten ab. Dabei können bestimmte Spalten ausgewählt oder Reihen gefiltert werden. Außerdem dient SELECT zur Darstellung verknüpfter Tabellen und zur Berechnung von Werten. Die Feldnamen können in der Anweisung in beliebiger Reihenfolge angegeben werden; mehrere Felder werden durch Komma getrennt. Alle Felder einer Tabelle lassen sich am einfachsten mit dem Sternchen \* ausgeben.

**JOIN** *Verknüpft Tabellen oder Sichten*

*Grundsyntax* `SELECT <Feldnamen | *> FROM <Tabelle1>  
JOIN <Tabelle2>  
ON <Tabelle1.Feld> = <Tabelle2.Feld>`

*Beispiele* `SELECT Artikelname, Kategorienname FROM Artikel  
JOIN Kategorien  
ON Artikel.KatNr = Kategorien.KatNr`

JOIN verknüpft zwei Tabellen anhand je eines Feldes aus beiden Tabellen, das übereinstimmende Werte enthält. Normalerweise ist dieses Feld in einer Tabelle Primärschlüssel und in der anderen Fremdschlüssel. Wird nichts weiter angegeben, handelt es sich um einen „Inner Join“, d. h. nur solche Reihen werden zurückgegeben, in denen in beiden Tabellen Werte existieren.

**INSERT** *Fügt neue Reihen in eine Tabelle oder Sicht ein*

*Grundsyntax* `INSERT [INTO] <Tabelle> VALUES (<wertliste>)`

*Beispiele* `INSERT INTO Kunden VALUES ('Müller', 'Carla',  
'Hauptstr. 11', '12345', 'Nirgendwo')`  
`INSERT Artikel (Nr, Kategorie, Name, Preis)  
VALUES (DEFAULT, 17, 'Bleistift', 1,29)`

Mit INSERT wird eine neue Reihe in eine Tabelle eingefügt. Dabei wird meist mit VALUES eine explizite Wertliste übergeben, die entweder genau so viele Werte in derselben Datentypfolge enthält, wie die Tabelle vorschreibt, oder

deren zu füllende Felder vor der VALUES-Klausel angegeben sind. Alternativ kann anstelle der VALUES-Klausel auch eine SELECT-Anweisung stehen, die die einzufügenden Werte liefert.

**UPDATE** *Verändert Werte in einer Tabelle oder Sicht*

*Grundsyntax* `UPDATE <Tabelle> SET <Feld> = <wert> [...]  
[FROM <Tabellen>]  
WHERE <Filterbedingung>`

*Beispiele* `UPDATE Kunde SET Tel = '040-12345', Fax = NULL  
WHERE KundeNr = 1125`  
`UPDATE Artikel SET Preis = Preis * 1,2  
FROM Artikel JOIN Kategorien  
ON Artikel.Kategorie = Kategorien.Kategorie  
WHERE Kategorienname = 'Getränke'`

Die UPDATE-Anweisung ändert die angegebenen Felder in allen Reihen, die zum angegebenen Filterkriterium passen. Achtung: Da UPDATE immer mengenbezogen arbeitet, werden alle Reihen einer Tabelle geändert, wenn man keinen Filter angibt!

**DELETE** *Löscht Reihen aus einer Tabelle oder Sicht*

*Grundsyntax* `DELETE <Tabelle> [FROM <Tabellen>]  
WHERE <Filterbedingung>`

*Beispiele* `DELETE Kunde WHERE KundeNr > 50`  
`DELETE Artikel  
FROM Artikel JOIN Kategorien  
ON Artikel.Kategorie = Kategorien.Kategorie  
WHERE Kategorienname = 'Getränke'`

DELETE löscht alle Reihen aus der direkt hinter der Anweisung angegebenen Tabelle, auf die das Filterkriterium zutrifft. Dabei kann sich das Filterkriterium auch auf Werte einer verknüpften Tabelle beziehen, wie das zweite Beispiel zeigt (lösche alle Artikel der Kategorie „Getränke“ aus der Artikeltable). Achtung: Auch DELETE arbeitet mengenbezogen und löscht alle Reihen, falls das Filterkriterium weggelassen wird!