

SQL Tabelle ändern

SQL Tabelle ändern ist der Schritt **danach**: die Tabelle steht, es fehlt eine Spalte. Der Dialog kann absichtlich nur den Teil von `ALTER TABLE`, bei dem nichts unabsichtlich verlorengeht.

Möglich	Nicht möglich
Spalte hinzufügen – immer mit NULL erlaubt, ohne Schlüssel, ohne Auto-Wert	Spalte umbenennen
Zeichenspalte verlängern – nur größer	Typ wechseln
Selbst angelegte Spalte löschen	Länge verkleinern, fremde Spalten löschen, Tabelle löschen

Die vorhandenen Spalten stehen deshalb als Text da, mit genau einem Eingabefeld dort, wo die Länge steht. Die Entscheidung darüber fällt im Portalskript, nicht im Formular.

Welche Tabelle geändert wird

Verbindung und Tabelle werden oben gewählt; Views stehen nicht zur Wahl, ihre Spalten kommen aus ihrer Abfrage. Sobald eine Tabelle feststeht, stehen ihre Spalten darunter, so wie die Datenbank sie meldet.

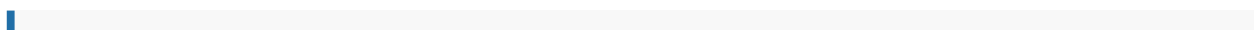
Spalte verlängern

Bei einer Zeichenspalte lässt sich die Länge **nur nach oben** ändern; der Tooltip nennt den aktuellen Wert und die Obergrenze des Typs. Was beim Verkürzen abgeschnitten würde, käme nicht zurück – deshalb gibt es diesen Weg nicht. Reicht die Obergrenze nicht, ist eine neue Spalte vom Typ *Langtext* der Weg.

Spalte löschen

Ein `×` hat nur eine Spalte, die **dieser Editor selbst angelegt** hat. Er merkt sich das in der Custom Property `TS_DdlAddedColumns`. Was aus der Auslieferung kommt oder von Hand entstanden ist, hat keinen Löschknopf und wird auch dann abgelehnt, wenn die Anfrage es verlangt.

Ein Klick markiert die Spalte nur. Gelöscht wird sie erst mit **Ändern** und einer Rückfrage – ein zweiter Klick auf das `×` nimmt die Markierung wieder weg.



Mit der Spalte fällt ihr Inhalt in jeder Zeile. Das lässt sich nicht zurückholen.

ID und **Licence** bleiben in jedem Fall, wie sie sind: eine Tabelle, in der sie anders aussehen, passt nicht mehr zu den anderen.

Änderungen in die Definitionen überführen

Die dritte Karte des Dialogs zieht die Definitionen nach – denn eine Spalte in der Datenbank ist erst zu gebrauchen, wenn eine Definition sie kennt.

Angeboten wird, was auf **genau diese Tabelle** zeigt (dieselbe Verbindung, derselbe Tabellenname), voreingestellt alles davon:

- Angehakte Definitionen bekommen die **neuen** Spalten eingetragen, **gelöschte** fallen dort heraus. Was schon dasteht, bleibt unangetastet.
- Die gerade **geöffnete** Definition wird dabei nicht im Hintergrund gespeichert: sie bekommt die Spalten im Arbeitsspeicher und gilt als geändert. So verliert niemand offene Änderungen dadurch, dass eine Tabelle erweitert wurde.
- Eine Definition, deren JSON nicht lesbar ist, wird nicht angefasst – sie muss erst von Hand geradegezogen werden.

Zeigt keine Definition auf die Tabelle, sagt der Dialog das: die Spalten entstehen dann nur in der Datenbank, und **Neu** macht daraus eine Definition.

SQL anzeigen und Ändern

Wie beim Anlegen zeigt **SQL anzeigen** das Statement, das später läuft, ohne etwas auszuführen. **Ändern** führt es aus; bei markierten Löschungen heißt der Knopf **Ändern und löschen** und fragt vorher nach.

Ist nichts zu tun – keine neue Spalte, keine gewachsene Länge –, sagt der Dialog das, statt ein leeres Statement zu schicken.

Revision #1

Created 2026-09-17 11:03:56 UTC by Helge Czerwinski

Updated 2026-09-17 11:03:56 UTC by Helge Czerwinski