

	ALTER TABLE Statement		AnPr	V 1.0
	Name	Klasse	Datum	

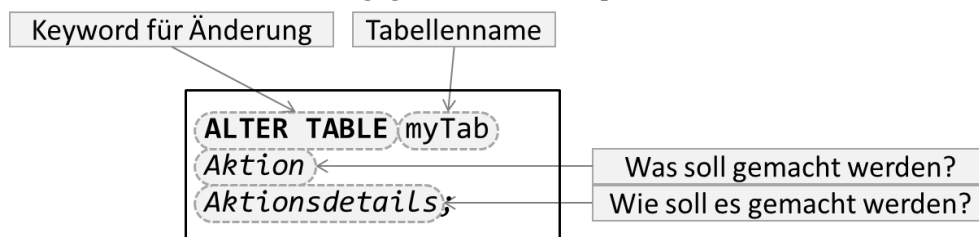
1 Denkbare Änderungen von Tabelleneigenschaften

Wenn wir Änderungen von Tabellen vornehmen möchten, gibt es streng genommen zwei Möglichkeiten. Entweder wir exportieren die Daten in ein CSV-File, löschen die Tabelle, legen sie neu an und laden die Daten wieder hinein – was sehr umständlich ist. Alternativ gibt es das ALTER TABLE Statement, welches die Änderungen direkt in den befüllten Tabellen vornimmt. Im Hintergrund passiert oftmals aber auf Spaltenebene auch ein Neuanlegen der Spalte und das hinüberkopieren der Daten – also ähnlich der erstgenannten Option. Folgende Anpassungen werden benötigt:

- Hinzufügen von Spalten
- Entfernen von Spalten
- Verändern von Spalten
- Umbenennen

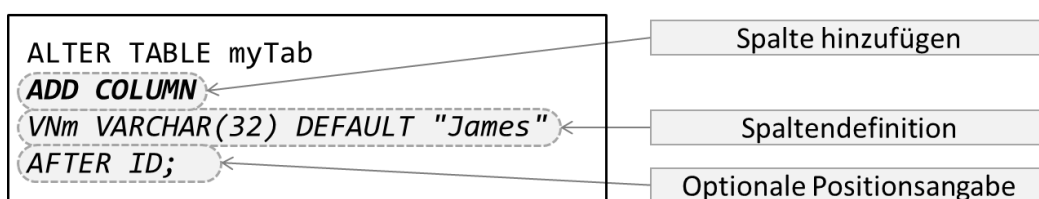
2 Aufbau des ALTER TABLE Statements

Das Statement ist sehr flexibel und kann entsprechend der oben genannten Anforderungen auch unterschiedliche Aktionen umsetzen. Diese müssen angegeben und näher spezifiziert werden.



2.1 Spalte hinzufügen

Um eine Spalte hinzuzufügen, benötigen wir eine Spaltendefinition, wie sie auch beim CREATE TABLE Statement eingesetzt wird. Der Syntax dieser Spaltendefinition ist hier auch identisch mit CREATE TABLE:



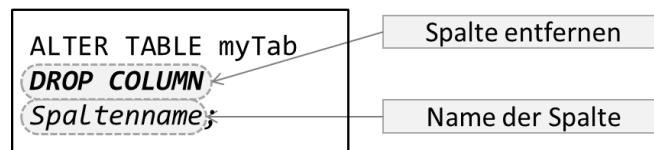
Darüber hinaus können wir noch angeben, an welcher Stelle die Spalte eingefügt werden soll:

- Ohne Angabe – die Spalte wird am Ende angefügt
- FIRST – die Spalte wird ganz vorne eingefügt
- AFTER *Spaltenname* – die Spalte wird nach der angegebenen Spalte eingefügt

Es ist theoretisch auch möglich, einen Primary Key nachträglich zu ergänzen – was in der Praxis jedoch kaum vorkommt, da die Primary Keys in der Regel bereits festgelegt wurden.

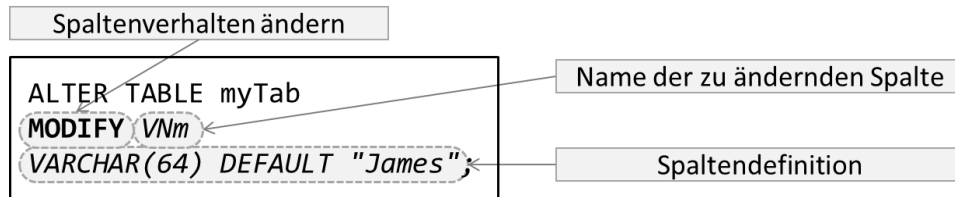
2.2 Entfernen von Spalten

Beim Löschen wird lediglich der Name der zu löschenden Spalte benötigt:



2.3 Ändern von Spalteneigenschaften

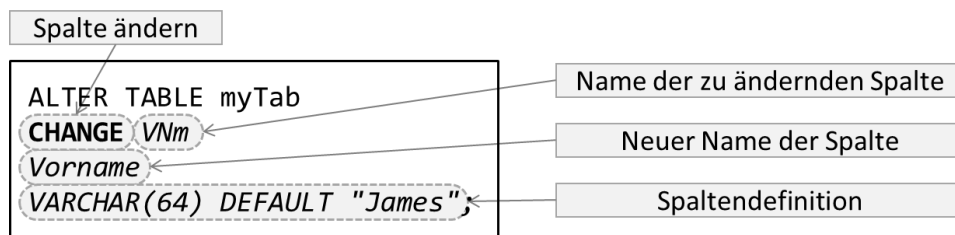
Bei den Eigenschaften von Spalten müssen wir die Verhaltenseigenschaften und den Namen getrennt voneinander betrachten. Beginnen wir mit der reinen Veränderung des Verhaltens, dem „MODIFY“:



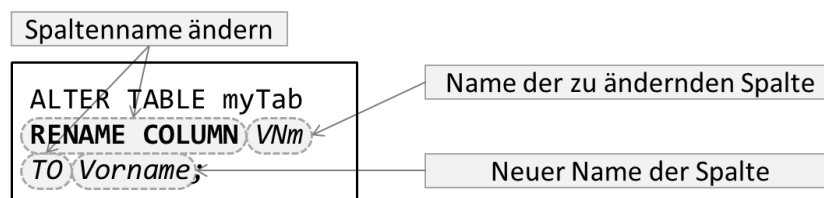
Wichtig ist hier, dass alle Eigenschaften der Spalte komplett angegeben werden müssen. Wenn die Spalte „VM“ vorher DEFAULT "James" war und die Eigenschaft beibehalten werden muss, so muss diese Eigenschaft beim MODIFY wieder erneut angegeben werden. Intern wird eine neue Spalte angelegt, die Daten kopiert und die alte Spalte anschließend gelöscht – es wird also die gesamte Spaltendefinition benötigt!

Bei Veränderungen des Datentyps werden die Daten so angepasst, dass sie in das neue Format eingetragen werden können. So wird bspw. bei Verkleinerung der VARCHAR Breite ggf. der Wert abgeschnitten. Hier ist unbedingt auf die Warnings zu achten. Sollten welche aufgetreten sein, kann man sie mit SHOW WARNINGS; detailliert anzeigen lassen.

Das Statement zum kompletten Ändern der Spalte ist „CHANGE“. Hier wird neben den Verhaltenseigenschaften (wie bei MODIFY) noch der neue Name angegeben:



Bis Exklusive Version 8 von MySQL war dies die einzige Möglichkeit, den Spaltennamen zu ändern. Wenn also nur der Name geändert werden soll, so muss die Spaltendefinition 1:1 der ursprünglichen Spalte entsprechen. Ab Version 8 von MySQL gibt es auch den Syntax „RENAME“:



Ganze Tabellen werden mit **ALTER TABLE oldName RENAME TO newName** umbenannt.

3 Aufgabe

Gegeben ist folgende Tabelle in einem MySQL Version 6 RDBMS:

```
mysql> desc customer;
```

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	
fName	varchar(256)	NO		NULL	
lName	varchar(256)	YES		NULL	
street	varchar(256)	YES		unbekannt	
plz	char(5)	YES		NULL	
city	varchar(128)	NO		NULL	
buLa	varchar(64)	YES		NULL	

- Löschen Sie die Spalte `bula`
- Ändern Sie die Spalte `city`, so dass bis zu 256 Zeichen hineinpassen
- Benennen Sie `street` in `strasse` um
- Ändern Sie das Feld `fName`, so dass NULL Werte erlaubt sind

[illegible]