**ALTER - Statement**

Mit Hilfe des Befehl *ALTER* kann die Struktur einer bestehenden Tabelle verÃ¤ndert werden. Dazu wird dem Statement je nach Bedarf eine *drop, add, change, modify* - Klausel hinzugefÃ¼gt.

Der grundlegende Aufbau sieht wie folgt aus.

ALTER [IGNORE] TABLE tbl\_name alter\_specification [, alter\_specification ...]

alter\_specification:

ADD [COLUMN] create\_definition [FIRST | AFTER column\_name ]

| ADD [COLUMN] (create\_definition, create\_definition,...)

| ADD INDEX [index\_name] (index\_col\_name,...)

| ADD PRIMARY KEY (index\_col\_name,...)

| ADD UNIQUE [index\_name] (index\_col\_name,...)

| ADD FULLTEXT [index\_name] (index\_col\_name,...)

| ADD [CONSTRAINT symbol] FOREIGN KEY [index\_name] (index\_col\_name,...)

[reference\_definition]

| ALTER [COLUMN] col\_name {SET DEFAULT literal | DROP DEFAULT}

| CHANGE [COLUMN] old\_col\_name create\_definition

[FIRST | AFTER column\_name]

| MODIFY [COLUMN] create\_definition [FIRST | AFTER column\_name]

| DROP [COLUMN] col\_name

| DROP PRIMARY KEY

| DROP INDEX index\_name

| DISABLE KEYS

| ENABLE KEYS

| RENAME [TO] new\_tbl\_name

| ORDER BY col

| table\_options

Das folgende Beispiel zeigt den Umgang mit dem ALTER TABLE-Statement.

| **Befehl** | **Ziel** |
| --- | --- |
|  | Wir beginnen mit dem Erzeugen einer Demodatenbank |
|  | Wir erzeugen eine Tabelle t1 |
|  | Wir ändern den Tabellennamen von t1 in t2 |
|  | Wir Ã¤ndern den Spaltentyp von *a* in TINYINT NOT NULL und Ã¤ndern den Spaltentyp von *b* nach CHAR(20) und geben der Spalte b den Namen *c* . |
|  | Wir fÃ¼gen eine neue Spalte *d* mit Datentyp TIMESTAMP hinzu. |
|  | Wir fÃ¼gen einen Index auf die Spalte *d* hinzu und machen aus der Spalte *a* einen PrimÃ¤rschlÃ¼ssel. |
|  | Wir lÃ¶schen die Spalte *c* |
|  | Wir fÃ¼gen eine neue Spalte *c* mit dem Datentyp INTEGER hinzu. Der Wert soll sich automatisch hochzÃ¤hlen. |

**Tabelle 2.14. Gebrauch des ALTER-Statements**