

MySQL – Kurzreferenz Teil 1: Administration

Datenbank

Datenbank erstellen

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name
[create_spezifikation [, ...] ...]
create_spezifikation:
  [DEFAULT] CHARACTER SET standardzeichensatz
  | [DEFAULT] COLLATE standardsortierfolge
```

Datenbank ändern und löschen

```
ALTER {DATABASE | SCHEMA} [db_name]
alter_spezifikation [, ...] ...
alter_spezifikation:
  [DEFAULT] CHARACTER SET standardzeichensatz
  | [DEFAULT] COLLATE standardsortierfolge
DROP {DATABASE | SCHEMA} [IF EXISTS] db_name
```

Syntax

Allgemeine Syntax

Tabellen können dauerhaft oder temporär erstellt werden. Sie können eine individuelle Spaltenliste haben oder aus einer Abfrage Strukturen und/oder Daten übernehmen.

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tabelle
[(create_definition,...)]
[tabelle_optionen] [select_anweisung]
```

Tabellen können strukturell als Kopie einer anderen Tabelle erstellt werden.

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tabelle
[() LIKE alte_tabelle ()];
```

```
create_definition:
  spalte_definition
  | [CONSTRAINT [name]] PRIMARY KEY [index_typ]
    (index_spalte,...)
  | KEY [index_name] [index_typ] (index_spalte,...)
  | INDEX [index_name] [index_typ]
    (index_spalte,...)
  | [CONSTRAINT [name]] UNIQUE [INDEX]
    [index_name] [index_typ] (index_spalte,...)
  | [CONSTRAINT [name]] FOREIGN KEY
    [index_name] (index_spalte,...)
    [verweis_definition]
  | CHECK (ausdruck)
```

```
spalte_definition:
  spalte typ [NOT NULL | NULL] [DEFAULT standardwert]
  [AUTO_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]
  [COMMENT 'string'] [verweis_definition]
```

MySQL-Datentypen

Zahlen

Alle mit Optionen: [(length)]
[UNSIGNED] [ZEROFILL]
TINYINT | SMALLINT | MEDIUMINT
| INT | INTEGER | BIGINT

Alle mit Optionen:
[(length,decimals)] [UNSIGNED]
[ZEROFILL]
REAL | DOUBLE | FLOAT

Alle mit Optionen: (length,decimals)
[UNSIGNED] [ZEROFILL]
DECIMAL | NUMERIC

Zeit

DATE | TIME | TIMESTAMP |
DATETIME | YEAR

Zeichenkette

CHAR(length) [BINARY | ASCII |
UNICODE]
VARCHAR(length) [BINARY]

Mit Option (length)
BINARY | VARBINARY

Ohne weitere Option

TINYBLOB | BLOB | MEDIUMBLOB |
LONGBLOB
spatial_type

Mit Option [BINARY]

TINYTEXT | TEXT | MEDIUMTEXT |
LONGTEXT

Mit Option (wert1,wert2,wert3,...)
ENUM | SET

Eigenschaften von Tabellen und Spalten

Tabellen haben Eigenschaften, von denen die meisten optional sind und viele durch Standardwerte gesetzt werden. Spalten können Einschränkungen für den zulässigen Wertebereich haben.

Spalten

- [NOT] NULL-Einschränkung für NULL-Werte (leeres Feld)
- CHECK-Bedingung für dynamische Werteüberprüfung bspw. mit Vergleichen oder Funktionen
- Angabe eines Wertebereichs mit SET

Tabellen

tabelle_optionen: tabelle_option [tabelle_option] ...

```
tabelle_option:
  (ENGINE|TYPE) [=] engine_name
  | AUTO_INCREMENT [=] wert
  | AVG_ROW_LENGTH [=] wert
  | [DEFAULT] CHARACTER SET zeichensatz
  | [COLLATE sortierung]
  | CHECKSUM [=] {0 | 1}
  | COMMENT [=] 'string'
  | CONNECTION [=] 'string'
  | MAX_ROWS [=] wert
  | MIN_ROWS [=] wert
  | PACK_KEYS [=] {0 | 1 | DEFAULT}
  | PASSWORD [=] 'string'
  | DELAY_KEY_WRITE [=] {0 | 1}
  | ROW_FORMAT [=] {DEFAULT|DYNAMIC|FIXED|COMPRESSED
  | REDUNDANT|COMPACT}
  | UNION [=] (tabelle[,tabelle]...)
  | INSERT_METHOD [=] { NO | FIRST | LAST }
  | DATA DIRECTORY [=] 'absoluter pfad'
  | INDEX DIRECTORY [=] 'absoluter pfad'
```

Schlüssel

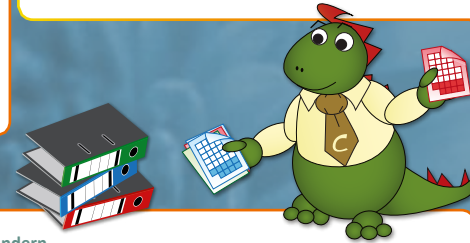
Allgemeine Syntax für eine Spaltendefinition:

```
spalte_definition:
  spalte typ [NOT NULL | NULL]
    [AUTO_INCREMENT]
    [UNIQUE [KEY] | [PRIMARY] KEY]
    [verweis_definition]
```

Primärschlüssel

Ein Primärschlüssel kann mit und ohne den Namen der Einschränkung angelegt werden.

```
[CONSTRAINT [symbol]] PRIMARY KEY
[index_typ] (index_spalte,...)
```



Tabellen ändern

Änderungen für Spalten, Tabellen-Einstellungen, Primär- und Fremdschlüssel sowie Änderungen für Indizes und Partitionen (siehe jeweils dort) werden über die ALTER TABLE-Anweisung durchgeführt.

```
ALTER [ONLINE | OFFLINE] TABLE name
  ADD [COLUMN] spaltenname definition
    [FIRST | AFTER spaltenname ]
  | ADD [COLUMN] (spaltenname definition,...)
  | ADD [CONSTRAINT [name]] PRIMARY KEY
    [index_type] (index_spaltenname,...)
    [index_option] ...
  | ADD [CONSTRAINT [name]]
    FOREIGN KEY [index_name]
      (index_spaltenname,...)
    Fremdschlüssel_referenz
  | ALTER [COLUMN] spaltenname {SET DEFAULT literal
  | DROP DEFAULT}
  | CHANGE [COLUMN] alter_spaltenname
    neuer_spaltenname definition
    [FIRST|AFTER spaltenname]
  | MODIFY [COLUMN] spaltenname definition
    [FIRST | AFTER spaltenname]
  | DROP [COLUMN] spaltenname
  | DROP PRIMARY KEY
  | DROP FOREIGN KEY fk_name
  | RENAME [TO] neuer_name
  | CONVERT TO CHARACTER SET zeichensatz_name
    [COLLATE collation_name]
  | [DEFAULT] CHARACTER SET [=]zeichensatz_name
    [COLLATE [=] collation_name]
```

Die Umbenennung gelingt auch über RENAME.

```
RENAME TABLE tbl_name TO new_tbl_name
[, tbl_name2 TO new_tbl_name2] ...
```

Fremdschlüssel

Ein Fremdschlüssel kann neben der einfachen Verknüpfung auch Angaben enthalten, was geschehen soll, wenn der Elterndatensatz gelöscht oder geändert wird: verhindern (RESTRICT), ebenfalls löschen oder aktualisieren (CASCADE), auf NULL setzen (SET NULL) oder nichts (NO ACTION).

```
verweis_definition:
  REFERENCES tabelle
    [(index_spalte,...)]
  [MATCH FULL | MATCH PARTIAL
  | MATCH SIMPLE]
  [ON DELETE verweis_option]
  [ON UPDATE verweis_option]
```

```
verweis_option:
  RESTRICT | CASCADE | SET NULL
  | NO ACTION
```

Primär- und Fremdschlüssel können definiert werden:

- Direkt innerhalb der Spaltendefinition
- Innerhalb der Tabellendefinition nach allen Spalten
- Außerhalb der Tabellendefinition in einer Änderungsanweisung

Information Schema

Im Information Schema (auch Data Dictionary oder Systemkatalog) befinden sich Sichten, mit denen man Informationen über das System, die Datenbanken und ihre Strukturen abfragen kann.

Die Tabellen des Systemkatalogs gehören alle zu INFORMATION_SCHEMA (abgekürzt durch ~). Hier eine Liste der wichtigsten:

Tabellen und Spalten

```
~.COLUMNS Spalten
~.TABLES Tabellen
~.TABLE_CONSTRAINTS Einschränkungen
~.REFERENTIAL_CONSTRAINTS Fremdschlüssel
~.TABLE_CONSTRAINTS Tabellen-Einschr
```

Speicher-Optionen

```
~.PARTITIONS Partitionen
```

Sicherheit

```
~.USER_PRIVILEGES Globale Rechte
~.TABLE_PRIVILEGES Rechte an Tabellen
~.SCHEMA_PRIVILEGES Rechte an Schemata
~.COLUMN_PRIVILEGES Rechte an Spalten
```

Sonstige Objekte

```
~.VIEWS Sichten
~.TRIGGERS Trigger
~.ROUTINES Prozeduren und Funktionen
```

Partitionen

Partitionierung dient der Aufteilung von großen Datenmengen in einzelnen Teilen (Partitionen). Bei DML-Operationen wird dann nur die entsprechende Partitionierung angesprochen.

Allgemeine Syntax

Die Definition von Partitionen folgt den Spalten- und Tabellenoptionen.

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] name
[(create_definition,...)]
[tabellenoptionen]
[partitionsoptionen]
```

Die Partitionsoptionen legen fest, wie die Partitionen grundsätzlich strukturiert sind. Bei HASH und bei KEY dürfen VALUES LESS THAN oder VALUES IN nicht verwendet werden.

- **HASH (ausdruck):** Erzeugt einen Schlüssel aus dem angegebenen Ausdruck und verteilt Daten gleichmäßig und automatisch. Der Ausdruck kann eine oder mehrere Spalten sowie Funktionen umfassen.
 - **KEY (spalten):** Ähnlich wie HASH, allerdings darf nur eine Spaltenliste angegeben werden und der Schlüssel wird automatisch ermittelt.
 - **RANGE (ausdruck):** Der Ausdruck enthält eine Spalte oder eine Funktion, deren Werte mit VALUES LESS THAN oder VALUES IN für die Partitionierung verwendet werden.
 - **LIST (ausdruck):** Der Ausdruck ist wie bei RANGE, doch werden die Werte für die Partitionen abschließend aufgezählt.
- ```
partitionsoptionen
PARTITION BY
{ [LINEAR] HASH(ausdruck)
| [LINEAR] KEY(spalten)
| RANGE(ausdruck)
| LIST(ausdruck) }
[PARTITIONS anzahl]
[(partitionsdefinition [,
partitionsdefinition] ...)]
```

In der Definition einer einzigen Partition gibt man maximale und minimale Reihen an, die Pfade für Daten- und Indexdateien sowie die Grenze bzw. Werte für die Partitionszuordnungen.

```
partitionsdefinition:
PARTITION name
[VALUES
{LESS THAN {(ausdruck) | MAXVALUE}
| IN (werteliste)}]
[[STORAGE] ENGINE [=] speicherart]
[COMMENT [=] 'text']
[DATA DIRECTORY [=] 'datenverzeichnis']
[INDEX DIRECTORY [=] 'indexverzeichnis']
[MAX_ROWS [=] max_reihen]
[MIN_ROWS [=] min_reihen]
[TABLESPACE [=] tablespace_name]
```

### Partitionen ändern

Über den ALTER TABLE-Befehl ändert man auch die Einstellungen für Partitionen, verwaltet sie oder definiert neue Partitionsvorgaben.

```
ALTER [ONLINE | OFFLINE] TABLE name
ADD PARTITION (partitionsdefinition)
| DROP PARTITION name
| COALESCE PARTITION nummer
| REORGANIZE PARTITION {namen INTO (definitionen)}
| ANALYZE PARTITION {namen | ALL }
| CHECK PARTITION {namen | ALL }
| OPTIMIZE PARTITION {namen | ALL }
| REBUILD PARTITION {namen | ALL }
| REPAIR PARTITION {namen | ALL }
| PARTITION BY ausdruck
| REMOVE PARTITIONING
```



## Sicherheit

### Benutzer

```
CREATE USER benutzer [IDENTIFIED BY [PASSWORD] 'passwort']
[, benutzer [IDENTIFIED BY [PASSWORD] 'passwort']] ...
DROP USER benutzer [, benutzer] ...
RENAME USER benutzer1 TO benutzer2
[, benutzer3 TO benutzer4] ...
```

### Passwort

```
SET PASSWORD = PASSWORD('passwort')
SET PASSWORD FOR benutzer = PASSWORD('passwort' ,)
```

### Berechtigungen

```
GRANT recht_typ [(spalte_liste)] [, recht_typ [(spalte_liste)]] ...
ON [objekt_typ] {tabelle | * | *.* | db_name.*}
TO benutzer [IDENTIFIED BY [PASSWORD] 'passwort']
[, user [IDENTIFIED BY [PASSWORD] 'passwort']] ...
[REQUIRE
NONE |
[[SSL | X509]]
[CIPHER 'cipher' [AND]]
[ISSUER 'issuer' [AND]]
[SUBJECT 'subject']]
[WITH option [option] ...]
objekt_typ =
TABLE
| FUNCTION
| PROCEDURE
option =
GRANT OPTION
| MAX_QUERIES_PER_HOUR anzahl
| MAX_UPDATES_PER_HOUR anzahl
| MAX_CONNECTIONS_PER_HOUR anzahl
| MAX_USER_CONNECTIONS anzahl
REVOKE recht_typ [(spalte_liste)] [, recht_typ [(spalte_liste)]] ...
ON [objekt_typ] {tabelle | * | *.* | db_name.*}
FROM benutzer [, benutzer] ...
REVOKE ALL PRIVILEGES, GRANT OPTION FROM benutzer [, benutzer] ...
```

## SHOW-Befehl

Mit dem SHOW-Befehl lassen sich verschiedene automatisierte Abfragen zum System und seinem Inhalt durchführen. Eine Auswahl:

Anzeige aller Tabellen und Sichten mit Typ-Angabe (FULL-Modus).

```
SHOW [FULL] TABLES [{FROM | IN} db_name]
[LIKE 'pattern' | WHERE expr]
```

Anzeige aller gerade geöffneten Tabellen.

```
SHOW OPEN TABLES [{FROM | IN} db_name]
[LIKE 'pattern' | WHERE expr]
```

Anzeige aller Spalten (im FULL-Modus auch Kommentare oder Privilegien).

```
SHOW [FULL] COLUMNS {FROM | IN} tbl_name
[{FROM | IN} db_name]
[LIKE 'pattern' | WHERE expr]
```

Anzeige von Indizes.

```
SHOW {INDEX | INDEXES | KEYS}
{FROM | IN} tbl_name
[{FROM | IN} db_name]
[WHERE expr]
```



Ausgabe der CREATE-Befehle für DB-Objekte.

```
SHOW CREATE {DATABASE | SCHEMA} db_name
SHOW CREATE {TABLE | VIEW} name
SHOW CREATE {FUNCTION | PROCEDURE | TRIGGER}
name
```

Ausgabe des Quelltextes programmierbarer Objekte.

```
SHOW {FUNCTION CODE | | PROCEDURE CODE} name
SHOW TRIGGERS [{FROM | IN} db_name]
[LIKE 'pattern' | WHERE expr]
```

Sicherheit

```
SHOW GRANTS [FOR user]
SHOW PRIVILEGES
```

## Index

Ein Index dient der Beschleunigung der Abfragen, welche mit einem Filter auf die indizierten Spalten zugreifen.

### Allgemeine Syntax

Indizes können direkt in der Tabellendefinition oder später über ALTER TABLE sowie über CREATE INDEX definiert werden.

```
CREATE [UNIQUE|FULLTEXT|SPATIAL] INDEX name
[index_typ]
ON tabelle (index_spalte,...)
[index_option] ...
```

Man nennt entweder nur die im Index aufzunehmende Spalte oder kann auch nur den ersten Teil des Spaltenwerts indizieren, indem man die Länge dieses Teils angibt.

```
index_spalte:
name [(länge)] [ASC | DESC]
```

Für InnoDB und MyISAM wird B-Baum-Speicherung der Indizes verwendet. Für nur im Hauptspeicher befindliche Tabellen kann man auch Hash-Speicherung verwenden.

```
index_typ:
USING {BTREE | HASH}
```

Bei Volltext-Indizes kann man einen Volltext-Parser nennen. Die Byte-Größe von Schlüsselwert-Indizes kann als Vorgabewert angegeben werden.

```
index_option:
KEY_BLOCK_SIZE [=] wert
| index_typ
| WITH PARSER parser_name
| COMMENT ,zeichenkette'
```

### Tabelle und Index ändern

Ein neuer Index kann über die ALTER TABLE-Anweisung definiert werden.

```
ALTER [ONLINE | OFFLINE] TABLE name
ADD {INDEX|KEY} [index_name]
[index_type] (index_spaltenname,...) [index_option]
...
| ADD [CONSTRAINT [name]]
UNIQUE [INDEX|KEY] [index_name]
[index_type] (index_spaltenname,...) [index_option]
...
| ADD FULLTEXT [INDEX|KEY] [index_name]
(index_spaltenname,...) [index_option] ...
| DROP {INDEX|KEY} index_name
```