



Datenbank

```
CREATE DATABASE 'Adressverwaltung'  
DEFAULT CHARACTER SET latin1  
COLLATE latin1_german1_ci;
```

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

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

```
DROP {DATABASE | SCHEMA} [IF EXISTS] db_name
```

MySQL-Programm

DB-Verbindung	-h server -u benutzer -p passwort
DBs anzeigen	SHOW DATABASES
DB auswählen	USE datenbank
Tabellen anzeigen	SHOW TABLES;
Spalten anzeigen	DESCRIBE tabelle;
Index anzeigen	SHOW INDEX FROM tabelle;
Beenden	QUIT

Transaktionen und Sperren

Transaktionen

```
SET {GLOBAL | SESSION} TRANSACTION ISOLATION  
LEVEL  
{ READ UNCOMMITTED | READ COMMITTED | REPEAT-  
ABLE READ | SERIALIZABLE }
```

```
START TRANSACTION | BEGIN [WORK]  
COMMIT [WORK] [AND [NO] CHAIN] [[NO] RELEASE]  
ROLLBACK [WORK] [AND [NO] CHAIN] [[NO] RELEASE]
```

Sicherungspunkte

```
SAVEPOINT bezeichner  
ROLLBACK [WORK] TO SAVEPOINT bezeichner  
RELEASE SAVEPOINT bezeichner
```

Tabellen sperren

```
LOCK TABLES  
tabelle [AS alias] {READ [LOCAL] | [LOW_PRIORITY] WRITE}  
[, tabelle [AS alias] {READ [LOCAL] | [LOW_PRIORITY] WRITE}]  
...  
UNLOCK TABLES
```

Tabellen

```
CREATE TABLE 'adressverwaltung'. 'Person' (  
'P_Nr' INT NOT NULL ,  
'P_Anrede' SET( 'Herr', 'Frau' ) NOT NULL ,  
'P_Vorname' VARCHAR( 30 ) NOT NULL ,  
'P_Nachname' VARCHAR( 30 ) NOT NULL ,  
'P_Geburtstag' DATE NULL ,  
PRIMARY KEY ( `P_Nr` )  
) ENGINE = MYISAM  
CHARACTER SET latin1  
COLLATE latin1_german1_ci  
COMMENT = 'Stammdaten von Personen'
```

```
ALTER TABLE 'person'  
CHANGE 'P_Nr'  
'P_Nr' INT(11) NOT NULL AUTO_INCREMENT
```

```
ALTER TABLE 'person' ADD INDEX ( 'P_Nachname' )
```

```
CREATE TABLE 'adressverwaltung'. 'kontakt' (  
'K_Nr' INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
'P_Nr' INT NOT NULL ,  
'K_Art' SET( 'Fest', 'Mobil', 'Email' ) NOT NULL ,  
'K_Wert' VARCHAR( 30 ) NOT NULL  
) ENGINE = MYISAM  
COMMENT = 'Kontakte einer Person'
```

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tabelle  
[[create_definition,...]]  
[tabelle_optionen] [select_statement]
```

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tabelle  
[() LIKE old_tabelle {}];
```

```
create_definition:  
spalte_definition  
| [CONSTRAINT {symbol}] PRIMARY KEY [index_typ] (index_spalte,...)  
| KEY [index_name] [index_typ] (index_spalte,...)  
| INDEX [index_name] [index_typ] (index_spalte,...)  
| [CONSTRAINT {symbol}] UNIQUE [INDEX]  
| [index_name] [index_typ] (index_spalte,...)  
| FULLTEXT [INDEX] [index_name] (index_spalte,...)  
| [WITH PARSER parser_name]  
| SPATIAL [INDEX] [index_name] (index_spalte,...)  
| [CONSTRAINT {symbol}] 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

index_spalte:
spalte [(length)] [ASC | DESC]

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

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'

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] ...
```

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] ...
```

Passwort

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

Basis-Syntax

```
SELECT
[ALL | DISTINCT | DISTINCTROW ]
[HIGH_PRIORITY]
[STRAIGHT_JOIN]
[SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
[SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
select_ausdruck, ...
[FROM tabelle_verweise
[WHERE filter]
[GROUP BY {spalte | ausdruck | position}
[ASC | DESC], ... [WITH ROLLUP]]
[HAVING filter]
[ORDER BY {spalte | ausdruck | position}
[ASC | DESC], ...]
[LIMIT [{beginn.} anzahl | anzahl OFFSET beginn]]
[PROCEDURE procedure_name(argument_list)]
[INTO OUTFILE 'datei' optionen
| INTO DUMPFILE 'datei']
[FOR UPDATE | LOCK IN SHARE MODE]]
```

Verknüpfung

```
SELECT P_Nachname AS Nachname,
       K_Art AS Art,
       K_Wert AS Wert
FROM kontakt k INNER JOIN person p
ON k.P_Nr = p.P_Nr
WHERE P_Nachname = 'Ebenhof'
ORDER BY K_Art
```

tabelle_verweise:
tabelle_verweis [, tabelle_verweis] ...

tabelle_verweis:
tabelle_faktor
| join_table

tabelle_faktor:
tabelle [[AS] alias]
{[USE|IGNORE|FORCE] INDEX (schluessel_liste)}
{ (tabelle_verweise)
| { ON tabelle_verweis LEFT OUTER JOIN tabelle_verweis
ON bedingungs_ausdruck }

join_table:
tabelle_verweis [INNER | CROSS] JOIN tabelle_faktor [bedingung]
| tabelle_verweis STRAIGHT_JOIN tabelle_faktor
| tabelle_verweis STRAIGHT_JOIN tabelle_faktor ON bedingung
| tabelle_verweis LEFT [OUTER] JOIN tabelle_verweis bedingung
| tabelle_verweis NATURAL [LEFT [OUTER]] JOIN tabelle_faktor
| tabelle_verweis RIGHT [OUTER] JOIN tabelle_verweis bedingung
| tabelle_verweis NATURAL [RIGHT [OUTER]] JOIN tabelle_faktor

bedingung:
ON bedingungs_ausdruck
| USING (spalte_liste)

Vereinigung

```
SELECT ...
UNION [ALL | DISTINCT] SELECT ...
[UNION [ALL | DISTINCT] SELECT ...]
```

Datenmanipulation

Einfügen

```
INSERT INTO 'adressverwaltung'. 'person'
( 'P_Anrede', 'P_Vorname', 'P_Nachname',
  'P_Geburtstag' )
VALUES
( 'Herr', 'Anton', 'Ebenhof', '1978-04-06' );
```

```
INSERT INTO 'kontakt'
( 'P_Nr', 'K_Art', 'K_Wert' )
VALUES ( '1', 'Email', 'anton@ebenhof.com' );
```

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
[INTO] tabelle [(spalte,...)]
VALUES (ausdruck | DEFAULT),...,(...),...
[ ON DUPLICATE KEY UPDATE spalte=ausdruck, ... ]
```

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
[INTO] tabelle
SET spalte={ausdruck | DEFAULT}, ...
[ ON DUPLICATE KEY UPDATE spalte=ausdruck, ... ]
```

```
INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]
[INTO] tabelle [(spalte,...)]
SELECT ...
[ ON DUPLICATE KEY UPDATE spalte=ausdruck, ... ]
```

Aktualisieren

```
UPDATE 'kontakt'
SET 'K_Wert' = 'anton@ebenhof.de'
WHERE 'K_Nr' = 1
```

```
UPDATE [LOW_PRIORITY] [IGNORE] tabelle
SET spalte1=ausdruck1 [, spalte2=ausdruck2 ...]
[WHERE filter]
[ORDER BY ...]
[LIMIT anzahl]
```

```
UPDATE [LOW_PRIORITY] [IGNORE] tabelle_verweise
SET spalte1=ausdruck1 [, spalte2=ausdruck2 ...]
[WHERE filter]
```

Ersetzen

```
REPLACE INTO 'person'
VALUES ( '1', 'Herr', 'Anton', 'Ebenhof', '1978-04-06' )
```

```
REPLACE [LOW_PRIORITY | DELAYED]
[INTO] tabelle [(spalte,...)]
VALUES (ausdruck | DEFAULT),...,(...),...
```

```
REPLACE [LOW_PRIORITY | DELAYED]
[INTO] tabelle
SET spalte={ausdruck | DEFAULT}, ...
```

```
REPLACE [LOW_PRIORITY | DELAYED]
[INTO] tabelle [(spalte,...)]
SELECT ...
```

Löschen

```
DELETE FROM 'kontakt'
WHERE 'K_Nr' = 1
```

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM tabelle
[WHERE filter]
[ORDER BY ...]
[LIMIT anzahl]
```

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
tabelle[*] [, tabelle[*]] ...
FROM tabelle_verweise
[WHERE filter]
```

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
FROM tabelle[*] [, tabelle[*]] ...
USING tabelle_verweise
[WHERE filter]
```

Vollständig löschen

```
TRUNCATE [TABLE] tabelle
```

Datei laden

```
LOAD DATA LOCAL INFILE 'C:\\person.csv'
REPLACE INTO TABLE 'person'
FIELDS TERMINATED BY ';'
ENCLOSED BY '"'
ESCAPED BY '\\'
LINES TERMINATED BY '\r\n'
```

```
LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE 'datei'
[REPLACE | IGNORE]
INTO TABLE tabelle
[FIELDS
[TERMINATED BY 'string']
[OPTIONALLY] ENCLOSED BY 'char'
[ESCAPED BY 'char']
]
[LINES
[STARTING BY 'string']
[TERMINATED BY 'string']
]
[IGNORE anzahl LINES]
[(spalte_oder_variable,...)]
[SET spalte = ausdruck,...]
```

Sichten

```
CREATE
[OR REPLACE] | ALTER
[ALGORITHM = (UNDEFINED | MERGE | TEMPTABLE)]
[DEFINER = { benutzer | CURRENT_USER }]
[SQL SECURITY { DEFINER | INVOKER }]
VIEW sicht [(spaltenliste)]
AS select_anweisung
[WITH [CASCADED | LOCAL] CHECK OPTION]
```

```
DROP VIEW [IF EXISTS]
sicht [, sicht] ...
```

```
SHOW CREATE VIEW sicht
```

Programmierbarkeit

Prozeduren & Funktionen

```
Prozeduren & Funktionen
CREATE PROCEDURE sp_name (
[ [ IN | OUT | INOUT ]
param_name typ [...]]
[merkmale ...] koerper
```

```
CREATE FUNCTION sp_name (
[param_name typ [...]])
RETURNS typ
[merkmale ...] koerper
```

merkmale:

```
LANGUAGE SQL
| [NOT] DETERMINISTIC
| { CONTAINS SQL | NO SQL | READS SQL DATA |
MODIFIES SQL DATA }
| SQL SECURITY { DEFINER | INVOKER }
| COMMENT 'zeichenkette'
```

```
ALTER {PROCEDURE | FUNCTION} sp_name [merkmale ...]
DROP {PROCEDURE | FUNCTION} [IF EXISTS] sp_name
```

Routine-Körper

```
CALL sp_name([parameter[,...]])
```

```
[beginn_label:] BEGIN
[anweisungen]
END [ende_label]
```

Variablen

```
DECLARE var_name[,...] typ [DEFAULT wert]
SET var_name = ausdruck [, var_name = ausdruck] ...
SELECT spalte[,...] INTO var_name[,...] tabelle_ausdruck
```

Trigger

```
CREATE
[DEFINER = { benutzer | CURRENT_USER }]
TRIGGER trigger_name trigger_time trigger_event
ON tabelle FOR EACH ROW trigger_stmt
```

```
DROP TRIGGER [schema_name.]trigger_name
```

