

Datenbank

Beispiel 1: Einfache DB-Erstellung

```
CREATE DATABASE `Adressverwaltung`;  
CREATE DATABASE IF NOT EXISTS `Adressverwaltung`;
```

Beispiel 2: Angabe von Zeichensatz und Sortierung

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

Beispiel 3: Datenbank löschen

```
DROP DATABASE `Adressverwaltung`;
```



Tabellen

Einfache Tabellen

Anlegen einer einfachen Tabelle.

```
CREATE TABLE IF NOT EXISTS `Adressverwaltung`.`Person` (  
  `P_Nr` INT NOT NULL ,  
  `P_Vorname` VARCHAR(30) NOT NULL,  
  `P_Nachname` VARCHAR(30) NOT NULL,  
  `P_Geburtstag` DATE NULL  
);
```

Erstellen einer Kopie.

```
CREATE TABLE `Adressverwaltung`.`Person_Kopie`  
LIKE `Adressverwaltung`.`Person`;
```

Erstellen einer Tabelle aus einer Abfrage.

```
CREATE TABLE `Adressverwaltung`.`Person_Filter`  
SELECT P_Nr, P_Nachname  
FROM `Adressverwaltung`.`Person`  
WHERE P_Geburtstag > '1980-01-01';
```

Tabellen ändern

Beispiel 1: Primärschlüssel hinzufügen

```
ALTER TABLE `Person`  
ADD CONSTRAINT `FK_Person` PRIMARY KEY (`P_Nr`);
```

Beispiel 2: Spalten-Eigenschaften ändern

```
ALTER TABLE `Person`  
CHANGE `P_Nr`  
`P_Nr` INT(11) NOT NULL AUTO_INCREMENT;
```

Beispiel 3: Spalte löschen

```
ALTER TABLE `Person`  
DROP COLUMN `P_Geburtstag`;
```

Beispiel 4: Spalte hinzufügen

```
ALTER TABLE `Person`  
ADD `P_Geburtstag` DATE NULL;
```

Primärschlüssel und Fremdschlüssel

Beispiel 1: Schlüssel innerhalb der Spaltendefinition

```
CREATE TABLE `Adressverwaltung`.`Person` (  
  `P_Nr` INT NOT NULL PRIMARY KEY,  
  `P_Vorname` VARCHAR(30) NOT NULL,  
  `P_Nachname` VARCHAR(30) NOT NULL  
);  
  
CREATE TABLE `Adresse` (  
  `A_Nr` INT NOT NULL PRIMARY KEY,  
  `P_Nr` INT NOT NULL REFERENCES `Person` (`P_Nr`),  
  `A_Strasse` VARCHAR(50) NOT NULL,  
  `A_PLZ` VARCHAR(5) NOT NULL,  
  `A_Stadt` VARCHAR(50) NOT NULL  
);
```

Beispiel 2: Schlüssel am Ende der Spaltenliste sowie als benannte Einschränkungen (Constraints).

```
CREATE TABLE `Person` (  
  `P_Nr` INT NOT NULL,  
  `P_Vorname` VARCHAR(30) NOT NULL,  
  `P_Nachname` VARCHAR(30) NOT NULL,  
  CONSTRAINT `FK_Person` PRIMARY KEY (`P_Nr`)  
);  
  
CREATE TABLE `Adresse` (  
  `A_Nr` INT NOT NULL,  
  `P_Nr` INT NOT NULL,  
  `A_Strasse` VARCHAR(50) NOT NULL,  
  `A_PLZ` VARCHAR(5) NOT NULL,  
  `A_Stadt` VARCHAR(50) NOT NULL,  
  CONSTRAINT `FK_Person` PRIMARY KEY (`P_Nr`),  
  CONSTRAINT `FK_Adresse_Person`  
  FOREIGN KEY (`P_Nr`) REFERENCES `Person` (`P_Nr`)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE  
);
```



Eigenschaften von Tabellen und Spalten

Beispiel: Tabellen mit erweiterten Spalten- und Tabellen-Eigenschaften

```
CREATE TABLE `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 CHECK (`P_Geburtstag` <= sysdate())  
) ENGINE = InnoDB  
CHARACTER SET latin1  
COLLATE latin1_german1_ci  
COMMENT = 'Stammdaten von Personen';
```

Index

Tabelle und Index ändern

Beispiel: Nachträgliche Erstellung eines Index.

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

Beispiele

Beispiel 1: Beispeltabelle mit UNIQUE-Schlüssel und drei Textspalten und einem direkt definierten Index.

```
CREATE TABLE `Adressverwaltung`.`Person` (  
  `P_Nr` INT AUTO_INCREMENT UNIQUE KEY,  
  `P_Vorname` VARCHAR(30) NOT NULL,  
  `P_Nachname` VARCHAR(30) NOT NULL,  
  `P_Stadt` VARCHAR(30) NOT NULL,  
  INDEX idx_Stadt (`P_Stadt`)  
);
```

Beispiel 2: Einfacher Index

```
CREATE INDEX idx_Nachname  
ON Adressverwaltung.Person (P_Nachname);  
DROP INDEX idx_Nachname  
ON Adressverwaltung.Person;
```

Beispiel 3: Mehrspalten-Index / Zusammengesetzter Index und Angabe der B-Baum-Speicherung (anstelle einer HASH-Speicherung)

```
CREATE INDEX idx_Name  
USING BTREE  
ON Adressverwaltung.Person (P_Nachname, P_Vorname);
```

Beispiel 4: Index aus dem ersten Buchstaben einer Spalte

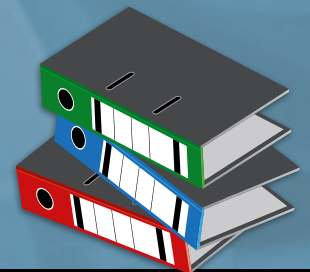
```
CREATE INDEX idx_Nachname  
ON Adressverwaltung.Person (P_Nachname (1) ASC);
```

Beispiel 5: Index für einen UNIQUE-Schlüssel, der dann wiederum einzigartige Werte enthält

```
CREATE UNIQUE INDEX idx_Nr  
ON Adressverwaltung.Person (P_Nr);
```

Beispiel 6: Löschen eines Index

```
DROP INDEX idx_Name  
ON Adressverwaltung.Person;
```



Partitionen

Beispiel 1: Automatische Partitionierung mit vorgegebener Partitionsanzahl und KEY

```
REATE TABLE Person (
Nr INT,
Name VARCHAR(30)
)
PARTITION BY KEY(Nr)
PARTITIONS 5;
```

Beispiel 2: Partitionierung über Angabe von maximalen Werten

```
CREATE TABLE Person (
Nr INT,
Name VARCHAR(30)
)
PARTITION BY RANGE (Nr) (
PARTITION p0 VALUES LESS THAN (10),
PARTITION p1 VALUES LESS THAN (20)
);
```

Beispiel 3: Partitionierung über direkt vorgegebene Werte in VALUES IN und zusätzliche Angabe der Speicherpfade (DATA und INDEX DIRECTORY nicht unter Windows).

```
CREATE TABLE Person (
Nr INT,
Name VARCHAR(30),
Vertrag DATE)
PARTITION BY LIST (YEAR(Vertrag))
(
PARTITION p2010 VALUES IN (2008, 2009, 2010)
DATA DIRECTORY = '/adrverw/2010/data'
INDEX DIRECTORY = '/adrverw/2010/idx',
PARTITION p2013 VALUES IN (2011, 2012, 2013)
DATA DIRECTORY = '/adrverw/2013/data'
INDEX DIRECTORY = '/adrverw/2013/idx'
);
```

Information Schema

Beispiel 1: Allgemeine Informationen zu einer Tabelle

```
SELECT table_name, engine, table_rows, table_comment
FROM INFORMATION_SCHEMA.TABLES
WHERE table_schema = 'Adressverwaltung';
```

Beispiel 2: Allgemeine Informationen zu Spalten

```
SELECT column_name, data_type, is_nullable, column_default
FROM INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'Person'
AND table_schema = 'Adressverwaltung';
```

Backup und Recovery

Backup

Sicherung von allen Datenbanken

```
mysqldump --all-databases > dump.sql
```

Sicherung von einzelnen Datenbanken

```
mysqldump --databases adressverwaltung buchverwaltung > dump.sql
```

Sicherung einer einzelnen Datenbank

SQL-Format

```
mysqldump --databases adressverwaltung > dump.sql
mysqldump adressverwaltung > dump.sql
```

Sicherung von einzelnen Tabellen

```
mysqldump adressverwaltung person kontakt > dump.sql
```

Text-Format

Sicherung mit Komma, Anführungszeichen und Zeilenumbruch in den Ordner backup

```
mysqldump --tab=/backup
--fields-terminated-by=,
--fields-enclosed-by='\"'
--lines-terminated-by=0x0d0a adressverwaltung
```

Recovery

Ausführen eines Dumps und Wiederherstellung inkl. CREATE DATABASE

```
mysql < dump.sql
```

Bei Skripten ohne CREATE und USE

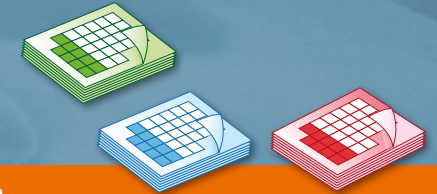
```
mysqladmin create adressverwaltung
```

Sicherheit und Benutzer

```
-- Benutzer erstellen
CREATE USER 'anton'@'localhost' IDENTIFIED BY 'ebenhof';
CREATE USER 'elvira'@'localhost' IDENTIFIED BY 'huelzemann';
-- Benutzer bearbeiten
RENAME USER 'anton'@'localhost' TO 'aebenhof'@'127.0.0.1';
SET PASSWORD FOR 'aebenhof'@'127.0.0.1' = PASSWORD('anton');
DROP USER 'anton'@'localhost';
```

```
-- Berechtigungen vergeben
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP
ON Adressverwaltung.*
TO 'anton'@'localhost';
GRANT SELECT, INSERT, UPDATE, DELETE
ON Adressverwaltung.Person
TO 'elvira'@'localhost';
```

```
-- Berechtigungen entziehen
REVOKE UPDATE, DELETE
ON *.* FROM 'elvira'@'localhost';
```



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;
Beenden	QUIT

Beispiel 1: Anmeldung

```
mysql adressverwaltung
mysql --user=anton --password=ebenhof
```

Beispiel 2: Skript ausführen

```
mysql adressverwaltung < script.sql > output.tab
```

