

SQL (Structured Query Language) ist eine international genormte Sprache zur Kommunikation mit relationalen Datenbanksystemen (RDBMS) wie zB. MySQL. MySQL implementiert SQL mit ein paar Abweichungen.

1. Eine Datenbank erzeugen und benutzen

Zunächst muss mit dem Befehl `CREATE DATABASE test_dbname` eine Datenbank erzeugt werden.

```
CREATE DATABASE test_Videothek;
```

Dann müssen Sie sicherstellen, dass Sie die richtige Datenbank bearbeiten. Mit dem Befehl `USE test_dbname` legen Sie die Datenbank fest, in der Sie die Tabellen erzeugen wollen.

```
USE test_Videothek;
```

2. Eine Tabelle erzeugen

Struktur der Datenbank überlegen, welche Tabellen samt Inhalt (Attribute im ERM) brauche ich?

Als Tupel: filme(filmnr, titel, genre, fsk, regisseur, laenge, veroeffentlichung)

Als Tabelle:

filme

<i>filmnr</i>	<i>titel</i>	<i>genre</i>	<i>fsk</i>	<i>regisseur</i>	<i>laenge</i>	<i>veroeffentlichung</i>
1000	Star Wars I	Sci-Fi	0	George Lucas	133	19-08-1999
1001	Star Wars II	Sci-Fi	6	George Lucas	142	16-05-2002
...

Der Code:

```
CREATE TABLE filme (
  filmnr INT(4) DEFAULT '1000' NOT NULL AUTO_INCREMENT,
  titel CHAR(100) NOT NULL,
  genre CHAR(30),
  fsk INT NOT NULL,
  regisseur CHAR(30),
  laenge INT,
  veroeffentlichung DATE,
  PRIMARY KEY (filmnr)
);
```

Die allgemeine Syntax:

```
CREATE TABLE [IF NOT EXISTS] tabellenname (
  spaltenname1 Datentyp Option,
  spaltenname2 Datentyp Option,
  PRIMARY KEY (spaltenname)
);
```

Wenn die Tabelle schon existiert kommt es zu einer Fehlermeldung. Dies kann mit dem Zusatz `IF NOT EXISTS` verhindert werden.

3. Datentypen

<code>INT[(N)] [UNSIGNED]</code>	Ganze bzw natürliche Zahl der Länge N
<code>SMALLINT[(N)] [UNSIGNED]</code>	-32768 bis 32767 bzw. 0 bis 65535.
<code>CHAR(N)</code>	String mit maximaler Länge N
<code>DATE</code>	Datum '1999-12-31'

Eckige Klammern bedeuten, dass der innere Ausdruck optional ist. Die Klammern werden nicht mitgeschrieben. (N) gibt die Länge N des Datentyps an, `UNSIGNED` steht für Vorzeichenlos, d.h. negative Zahlen fallen weg.

`INT (4) DEFAULT '1000' NOT NULL AUTO_INCREMENT` bedeutet, dass die Zahl immer um 1 erhöht wird, hier beginnt sie ab 1000. Durch die Länge 4 ist sie beschränkt, nimmt also maximal den Wert '9999' an.

4. Schlüssel

Ein Attribut wird zum Primärschlüssel (`PRIMARY KEY`) gemacht, in dem man den Befehl `PRIMARY KEY` der Aufzählung der Schlüssel voranstellt.

Die Attribute müssen vorher eingeführt werden und dürfen nicht `Null` sein.

```
CREATE TABLE kunden
(
  kNr INT NOT NULL,
  ...
  PRIMARY KEY (kNr)
);
```

Fremdschlüssel (`FOREIGN KEY`):

Im Beispiel Film- und Kundendatei möchte man in der Kundendatei den Lieblingsfilm `filme.filmnr` speichern.

```
CREATE TABLE kunden
(
  kNr INT NOT NULL,
  filmnr INT(4) DEFAULT '1000' NOT NULL AUTO_INCREMENT,
  ...
  PRIMARY KEY (kNr),
  FOREIGN KEY (filmnr) REFERENCES filme(filmnr)
);
```

Das Attribut `filme.filmnr`, also `filme(filmnr)`, muss erzeugt worden sein, bevor darauf referenziert werden kann.