

Name:

Klasse:

Datum:

Schuljahr: 2022/23

Im letzten Bereich haben Sie Daten importiert, der nächste Schritt ist es, diese nun zu exportieren.

## SQL-Dump

Die einfachste Form des Datenexports ist die Verwendung des Tools `mysqldump`.

Mit nachfolgendem Beispiel wird die komplette Datenbank `gm1` in der Datei `gm1_dump.sql` gespeichert. Das Ergebnis kennen Sie schon, da Sie mit solch einem Dump die Datenbank erzeugt und mit Inhalt gefüllt haben.

```
shell> mysqldump -u test --databases gm1 > gm1_dump.sql
```

Der Inhalt der Datei (Auszug) sieht dann wir folgt aus:

```
-- MySQL dump 10.19  Distrib 10.3.34-MariaDB, for debian-linux-gnu (x86_64)
--
-- Host: localhost    Database: gm1
--
-- Server version      10.3.34-MariaDB-0ubuntu0.20.04.1

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS
=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Current Database: `gm1`
--

CREATE DATABASE /*!32312 IF NOT EXISTS*/ `gm1` /*!40100 DEFAULT CHARACTER SET 2
utf8 */;

USE `gm1`;

--
-- Table structure for table `abteilung`
--

DROP TABLE IF EXISTS `abteilung`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `abteilung` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) DEFAULT NULL,
  `lid` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_abteilung_mitarbeiter` (`lid`),
  CONSTRAINT `fk_abteilung_mitarbeiter` FOREIGN KEY (`lid`) REFERENCES `2
mitarbeiter` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8;
...
```



Informieren Sie sich über den Befehl mysqldump.

**Notiz:**

**AUF-05-1** Führen Sie nun den Export der Datenbanken gm1 und aller weiterer vorhandenen Datenbanken selbst durch.

Wie lautet dabei der Befehl?

**Notiz:**



## Export als Textdatei

Mit dem Befehl `SELECT INTO OUTFILE` kann der Output des Select-Befehls unter Angabe von Parametern in eine Datei geschrieben werden (Das Gegenstück zu `LOAD DATA LOCAL INFILE`).  
Machen Sie sich mit dem Befehl vertraut und notieren Sie sich die wichtigsten Parameter.

### Notiz:

Nachfolgendes Beispiel erzeugt eine CSV-Datei.

```
SELECT id, name, vorname, gebdat
  INTO OUTFILE '/tmp/mitarbeiter.csv'
  CHARACTER SET utf8
  FIELDS TERMINATED BY ','
  OPTIONALLY ENCLOSED BY '"'
  LINES TERMINATED BY '\n'
FROM mitarbeiter;
```

Das Ergebnis (Auszug) sieht dann wie folgt aus:

```
1, "Lorenz", "Sophia", "1975-05-22"
2, "Ritter", "Tatjana", "1978-11-12"
3, "Wolff", "Theodor", "1985-01-09"
4, "Richter", "Hans-Otto", "1966-08-24"
...
```

Tritt beim Ausführen des Befehls der Fehler

```
Access denied for user 'test'@'localhost' ...
```

auf, so fehlt das Recht `FILE`, welches explizit dem User zugewiesen werden muss. Ansonsten kann nur `root` den Export durchführen!

```
GRANT FILE ON *.* TO 'test'@'localhost';
FLUSH PRIVILEGES;
```

**AUF-05-2** Exportieren Sie nun die Tabelle `mitarbeiter` mit allen Spalten in eine Datei mit der Endung `txt`. Als Trennzeichen verwenden Sie das Tab-Zeichen. Die Spalten werden dabei von keinem Zeichen eingebettet.

### Notiz:



## Automatischer Export

Möchte man regelmäßig die Daten exportieren, so hilft der Befehl CREATE EVENT.

```
CREATE EVENT automatic_export_mitarbeiter
ON SCHEDULE EVERY 1 DAY_HOUR STARTS '2022-01-24 23:00:00'
ON COMPLETION PRESERVE
ENABLE
COMMENT 'Save Table to CSV for Excel'
Do SELECT *
INTO OUTFILE '/tmp/mitarbeiter.csv'
CHARACTER SET utf8
FIELDS TERMINATED BY ';'
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
FROM `gml`.`mitarbeiter`;
```

Damit der Event auch gestartet wird, muss das Event-System in der Config-Datei von MariaDB oder über das Ändern der globalen Variablen eingeschaltet sein.

```
set global_event_scheduler=ON
```

**AUF-05-3** Beschreiben Sie die Parameter und deren Funktion bei obigem Beispiel.

**Notiz:**



## Export als XML-Datei

Ruft man `mysql` direkt von der Shell aus auf, so kann man das Ausgabeformat entsprechend festlegen, z. B. mit `--xml` für das XML-Format. Siehe hierzu `mysql Command-line Client`.

Nachfolgendes Beispiel gibt den Inhalt der Tabelle `mitarbeiter` im XML-Format aus.

```
shell> mysql -u test gml --xml -e 'select * from mitarbeiter' > /tmp/␣
mitarbeiter.xml
```

Das Ergebnis (Auszug) sieht dann wie folgt aus:

```
<?xml version="1.0"?>
<resultset statement="select * from mitarbeiter" xmlns:xsi="http://www.w3.org␣
/2001/XMLSchema-instance">
  <row>
    <field name="id">1</field>
    <field name="name">Lorenz</field>
    <field name="vorname">Sophia</field>
    <field name="gebdat">1975-05-22</field>
    <field name="strasse">Hammer Str. 349</field>
    <field name="oid">7940</field>
    <field name="aid">1</field>
    <field name="fid">1</field>
    <field name="eingestellt">2000-01-01</field>
  </row>
  ...
```

**AUF-05-4** Exportieren Sie nun die Tabelle `mitarbeiter` wie oben gezeigt.

**Notiz:**

Auch bei dem Befehl `mysqldump` funktioniert der Parameter `--xml`.

```
shell> mysqldump -u test --xml --databases gml > gml_dump.xml
```

**AUF-05-5** Exportieren Sie nun den Dump in eine XML-Datei wie oben gezeigt. Vergleichen Sie nun die beiden Ergebnisse (SQL, XML). Was fällt Ihnen auf, welche Vorteile haben die jeweiligen Formate?

**Notiz:**

