Probleme mit Umlauten zwischen Datenbank und Webseite (UTF-8/latin1/ISO8859-1)

Bei der Wiederherstellung oder dem Umzug einer Datenbank kann es manchmal zu Problemen kommen. Sonderzeichen und Umlaute werden dann auf der Webseite nicht mehr korrekt dargestellt. Eine Ursache ist z.B. wenn auf dem alten Server die Datenbank als Latin1 gedumpt wird und auf dem neuen ohne vorherige Konvertierung in eine UTF-8 DB importiert wird. MySQL und PHP gehen dann davon aus, dass die Daten in UTF-8 sind, obwohl diese eigentlich im Latin1 Zeichensatz vorliegen.

Dieses Problem läßt sich wie folgt beheben.

Dump der Datenbank, wir sagen dem mysqldump explizit, dass es sich um Latin1 handelt:

```
mysqldump --skip-extended-insert --default-character-set=latin1 <database> -r <database>-latin1.sql
```

- -skip-extended-insert fasst mehrere Zeilen nicht mehr in einem einzelnen INSERT zusammen. Das erleichtert einen Diff, verlangsamt allerdings bei großen Datenbanken den Reimport deutlich (optional)
- -r speichert den Output direkt in eine Datei. Es soll schon vorgekommen sein, dass durch die Umleitung von STDOUT über > in eine Datei komische Sachen passiert sind

Mit dem Tool iconv wird nun das Dumpfile tatsächlich auf UTF-8 gesetzt. Der Parameter „-c“ ist dabei optional. Er läßt Zeichen, die nicht konvertiert werden können einfach weg (Achtung: möglicher Datenverlust).

```
iconv -f UTF-8 -t UTF-8 -c <database>-latin1.sql > <database>-utf8.sql
```

Wenn wir nun die beiden Dateien vergleichen, sollten in der utf8-Datei die Umlaute korrekt angezeigt werden:

```
diff <database>-latin1.sql <database>-utf8.sql
```

Im Dumpfile müssen eventuelle Datenbanken/Tabellen noch auf UTF-8 gestellt werden:

```
# grep latin1 <database>-utf8.sql
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
  `name` varchar(255) CHARACTER SET latin1 NOT NULL,
  `description` text CHARACTER SET latin1 NOT NULL,
  `source` varchar(255) CHARACTER SET latin1 NOT NULL,


# sed -i 's/latin1/utf8/g' <database>-utf8.sql
```

Der MySQL-Server und die verbindenden Clients sollten nun natürlich auch UTF-8 liefern/schreiben. Dazu brauchts ein paar Ergänzungen an den entsprechenden Stellen in der my.cnf:

```
[client]
[...]
default-character-set   = utf8


[...]


[mysqld]
[...]
character-set-server    = utf8
collation-server        = utf8_unicode_ci
```

Danach den MySQL-Server neu starten (Achtung: Das nachträgliche Umstellen des Zeichensatzes kann unter Umständen bei bestehenden Datenbanken für Probleme sorgen).

```
# service mysql restart
```

Die Einstellungen überprüfen:

```
mysql> show variables like '%character_set_%';
+--------------------------+----------------------------+
| Variable_name            | Value                      |
+--------------------------+----------------------------+
| character_set_client     | utf8                       |
| character_set_connection | utf8                       |
| character_set_database   | utf8                       |
| character_set_filesystem | binary                     |
| character_set_results    | utf8                       |
| character_set_server     | utf8                       |
| character_set_system     | utf8                       |
| character_sets_dir       | /usr/share/mysql/charsets/ |
+--------------------------+----------------------------+
8 rows in set (0.00 sec)
```

Nun kann man den Dump einspielen:

```
mysql <datenbank> < <database>-utf8.sql
```

Revision #1
Created 27 July 2021 12:37:06 by magenbrot
Updated 27 July 2021 12:39:27 by magenbrot