

## OSM Datei direkt aus POI Datenbank erstellen

Wir bereiten jetzt die Datenbank für die Erstellung der OSM Datei vor.

Hierzu erstellen wir temporär neue Spalten, in die wir unsere Daten mitsamt den dazugehörigen TAGs und Zeilenumbrüchen schreiben. Anschließend erstellen wir daraus eine Tabelle mit nur einer Spalte und nach Export und Umbenennen, ist unsere OSM Datei fertig. Genauso erstelle ich auch KML- und GPX Dateien.

--- Alles ab hier kopieren und unter "SQL ausführen" einfügen und ausführen (inkl. Beschreibung)-----

-- temporäre Spalten für die OSM-Datei Erstellung

```
ALTER TABLE Test ADD COLUMN IDLATLON TEXT DEFAULT "";
ALTER TABLE Test ADD COLUMN NAME2 TEXT DEFAULT "";
ALTER TABLE Test ADD COLUMN DESCR TEXT DEFAULT "";
ALTER TABLE Test ADD COLUMN CAPACITY TEXT DEFAULT "";
ALTER TABLE Test ADD COLUMN OPENING TEXT DEFAULT "";
ALTER TABLE Test ADD COLUMN PHONE TEXT DEFAULT "";
ALTER TABLE Test ADD COLUMN EMAIL2 TEXT DEFAULT "";
ALTER TABLE Test ADD COLUMN WEB2 TEXT DEFAULT "";
ALTER TABLE Test ADD COLUMN CATEGORY TEXT DEFAULT "";
```

-- Öffnungszeiten OsmAnd konform anpassen, dadurch sieht man ob zur Zeit offen oder geschlossen ist.

-- Geht nur wenn sie alle einheitlich sind, sonst kommen die Öffnungszeiten mit in die Beschreibung.

```
UPDATE Test SET Offen = replace(Offen, '.01.', 'Jan ') WHERE Offen LIKE '%.01.%';
UPDATE Test SET Offen = replace(Offen, '.02.', 'Feb ') WHERE Offen LIKE '%.02.%';
UPDATE Test SET Offen = replace(Offen, '.03.', 'Mar ') WHERE Offen LIKE '%.03.%';
UPDATE Test SET Offen = replace(Offen, '.04.', 'Apr ') WHERE Offen LIKE '%.04.%';
UPDATE Test SET Offen = replace(Offen, '.05.', 'Mai ') WHERE Offen LIKE '%.05.%';
UPDATE Test SET Offen = replace(Offen, '.06.', 'Jun ') WHERE Offen LIKE '%.06.%';
UPDATE Test SET Offen = replace(Offen, '.07.', 'Jul ') WHERE Offen LIKE '%.07.%';
UPDATE Test SET Offen = replace(Offen, '.08.', 'Aug ') WHERE Offen LIKE '%.08.%';
UPDATE Test SET Offen = replace(Offen, '.09.', 'Sep ') WHERE Offen LIKE '%.09.%';
UPDATE Test SET Offen = replace(Offen, '.10.', 'Oct ') WHERE Offen LIKE '%.10.%';
UPDATE Test SET Offen = replace(Offen, '.11.', 'Nov ') WHERE Offen LIKE '%.11.%';
UPDATE Test SET Offen = replace(Offen, '.12.', 'Dec ') WHERE Offen LIKE '%.12.%';
UPDATE Test SET Offen = substr(Offen,4,4)||substr(Offen,2,2)||'-'||substr(Offen,11,4)||substr(Offen,9,2)
WHERE Offen != "";
```

-- Hochkommas spielen in der OSM Datei eine wichtige Rolle und müssen deshalb in allen Spalten im Text durch die html Schreibweise (&apos;) ersetzt werden (z.B. die Appostroph in FR usw.)

-- Unsere zu erzeugende Strings, müssen deshalb auch in Anführungszeichen statt wie üblich in Hochkommas.

-- (Hier ist die Schreibweise dafür Anführungszeichen Hochkomma Anführungszeichen)

```
UPDATE Test SET Name = replace(Name, "'", '&apos;');
UPDATE Test SET Email = replace(Email, "'", '&apos;');
UPDATE Test SET WEB = replace(WEB, "'", '&apos;');
UPDATE Test SET Nebenkosten = replace(Nebenkosten, "'", '&apos;');
UPDATE Test SET Beschreibung = replace(Beschreibung, "'", '&apos;');
```

-- Das Ersetzen trifft auch für alle anderen vorkommenden Sonderzeichen zu

```
UPDATE Test SET EMAIL = replace(EMAIL, '&', '&amp;') WHERE EMAIL LIKE '%&%';
UPDATE Test SET WEB = replace(WEB, '&', '&amp;') WHERE WEB LIKE '%&%';
```

-- Jetzt können wir die neuen Spalten inklusive Tags erstellen, dabei können "Name" und "Description" nach eigenem Wunsch zusammengestellt werden. Achtung!! Zeilenumbrüche nicht entfernen.

-- ("trim" und "replace" sind zum Entfernen überflüssiger Kommas wenn einzelne Werte fehlen)

-- Platztyp benutzen wir später als Filter in OsmAnd.

-- ROWID ist eine SQLite interne fortlaufende Zahl.

```
UPDATE Test SET IDLATLON = "<node id=''||ROWID||' visible='true' lat=''||Latitude||' lon=''||Longitude||'>";
```

```
UPDATE Test SET NAME2 = "
<tag k='name' v=''||trim(replace(Platztyp||, 'Preis||', 'Name||', 'VE, ', ', '), ', ')||'>";
```

```
UPDATE Test SET DESCR = "
<tag k='description' v=''||trim(Nebenkosten||, 'Beschreibung, ', ')||'>" WHERE Nebenkosten != " AND
Beschreibung != ";
```

```
UPDATE Test SET PHONE = "
<tag k='phone' v=''||Telefon||'>" WHERE Telefon != ";
```

```
UPDATE Test SET EMAIL2 = "
<tag k='email' v=''||Email||'>" WHERE Email != ";
```

```
UPDATE Test SET WEB2 = "
<tag k='website' v=''||Web||'>" WHERE Web != ";
```

```
UPDATE Test SET CAPACITY = "
<tag k='capacity' v=''||Platzanzahl||'>" WHERE Platzanzahl != ";
```

```
UPDATE Test SET OPENING = "
<tag k='opening_hours' v=''||Offen||'>" WHERE Offen != ";
```

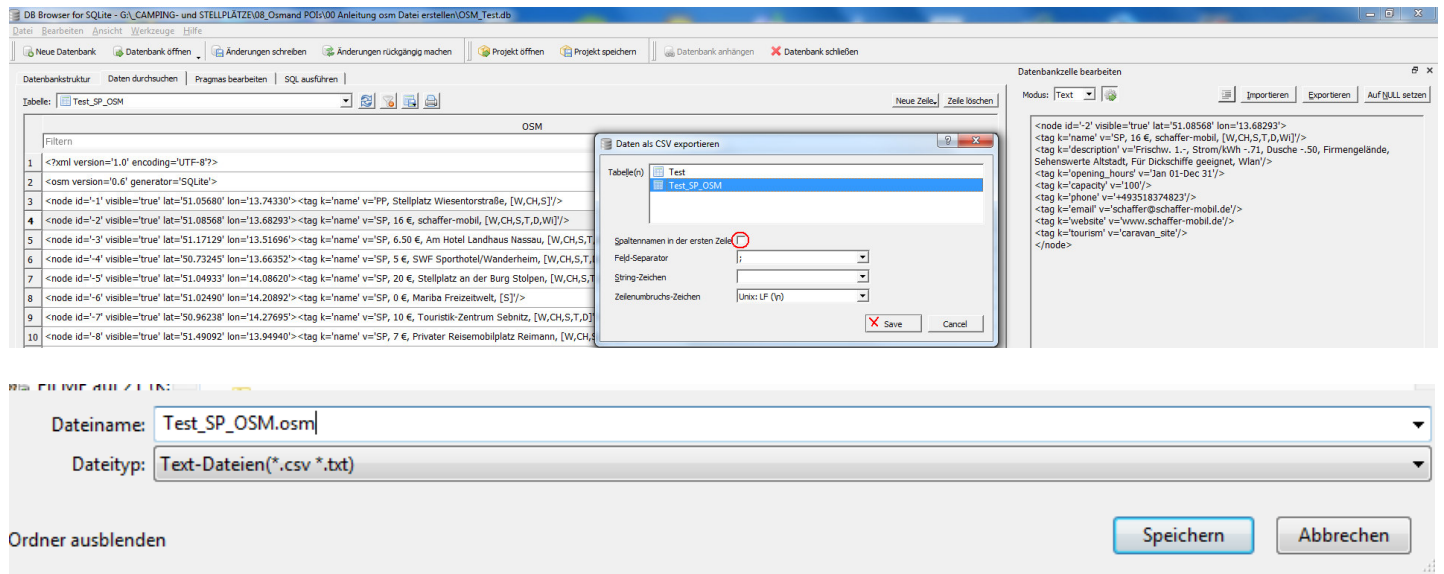
```
UPDATE Test SET CATEGORY = "
<tag k='tourism' v='caravan_site'/>
</node>";
```

--- neue Tabelle mit nur einer Spalte erstellen und die einzelnen Zeilen der Reihe nach einfügen

```
CREATE TABLE Test_SP_OSM (OSM TEXT);
```

```
INSERT INTO Test_SP_OSM Values ("<?xml version='1.0' encoding='UTF-8'?>");
INSERT INTO Test_SP_OSM Values ("<osm version='0.6' generator='SQLite'>");
INSERT INTO Test_SP_OSM SELECT
IDLATLON||NAME2||DESCR||OPENING||CAPACITY||PHONE||EMAIL2||WEB2||CATEGORY FROM Test;
INSERT INTO Test_SP_OSM Values ('</osm>');
```

-- Neue Tabelle exportieren ohne Spaltennamen als \*.osm



-- Datenbank schließen ohne! zu speichern. So bleibt die Masterdatenbank im Original erhalten.

-- Danach mit "OsmAndMapCreator" umwandeln in \*.obf .