

Eigene POIs erstellen mit dem SQLite DB_Browser am Beispiel der ACSI Camping Europa Trialversion. (ACSI_Gardasee)

Die ACSI Campingplatz-Daten sind über mehrere Tabellen in mehreren Datenbanken verteilt. Mein erster Schritt ist dann immer, alle Länderdatenbanken in einer zu vereinigen. Das brauchen wir hier erst mal nicht. Wir beginnen erst einmal nur mit der Trial Datenbank für den Gardasee. Diese finden wir, nachdem wir die ACSI Europa App installiert haben, unter “android/data/eu.acsi.europa/files/pointofinterest.1.EURO.10.sqlite”.

Wir starten den SQLite DB-Browser und öffnen die Datenbank “pointofinterest.1.EURO.10.sqlite”. Dann klicken wir auf “Daten durchsuchen” und hier auf “PointOfInterest”. Dies ist die Haupttabelle und enthält unter anderem die Koordinaten, den Namen und die Länderkennung der POIs. In den anderen Tabellen stehen noch Daten, wie z.B. die Öffnungszeiten, Preise, Beschreibungen usw.

Hier meine in den Abfragen verwendeten Farben:

Datenbankbefehle: blau
Tabellennamen: braun
Spaltennamen: grün
Spalteninhalt: schwarz
Strings: rot

Anweisungen und Tabellennamen können Groß- oder kleingeschrieben werden. Tabelleninhalte immer so wie sie sind. Tabelleninhalte müssen beim Einfügen oder Bearbeiten immer in Hochkommas gesetzt werden. Sollte der Text selbst Hochkommas enthalten, dann Ersatzweise auch in Anführungszeichen.

The screenshot shows the SQLite DB Browser for Windows interface. The title bar reads "DB Browser for SQLite - G:\CAMPING- und STELLPLÄTZE\BEARBEITEN\01_GRUND-DATENBANKEN erstellen\01_ACSI\11_ACSI_komplette_EURO+CCA Datenbank erstellen\01 ROH-...". The menu bar includes "Datei", "Bearbeiten", "Ansicht", "Werkzeuge", and "Hilfe". The toolbar contains icons for "Neue Datenbank", "Datenbank öffnen", "Änderungen schreiben", "Änderungen rückgängig machen", "Projekt öffnen", "Projekt speichern", and "Datenbank anhängen". Below the toolbar, there are three red annotations: "1" next to "Datenbank öffnen", "2" next to "Daten durchsuchen", and "3" next to "SQL ausführen". The "Daten durchsuchen" button is highlighted with a red circle. Below this, the "Tabelle auswählen" dropdown menu is open, showing "PointOfInterest" selected. The main area displays a table with the following data:

	<u>I</u> d	<u>N</u> ame	<u>S</u> earchName	<u>L</u> atitude	<u>L</u> ongitude	<u>R</u> egionId	<u>C</u> ountryIso3	<u>C</u> hainId	<u>S</u> ourceKey
	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern
1	463	Camping Lago...	camping lago ...	45.93592	10.81192	526	ITA	NULL	121458
2	515	Camping Pro...	camping pro...	45.63342	10.61275	572	ITA	NULL	121356
3	1853	Camping Baia...	camping baia ...	45.56143	10.55359	572	ITA	NULL	117704
4	2415	Camping Clau...	camping claudia	45.78842	10.82306	514	ITA	NULL	115100
5	2434	Zoo-Camping	zoo-camping	45.93272	10.89294	526	ITA	NULL	114957

Wir können hier schon mal eine Testabfrage starten, um uns diese vier Spalten anzeigen zu lassen. Dazu öffnen wir den Reiter “SQL ausführen” geben folgende Abfrage ein und klicken dann auf den kleinen blauen Pfeil.

```
SELECT Longitude, Latitude, Name, CountryIso3 AS Land
FROM PointOfInterest
WHERE CountryIso3 = 'ITA'; -->(macht hier natürlich keinen Sinn und kann weggelassen werden)
```

Mit dem grünen Kreuz ganz links, kann man weitere Abfragefenster öffnen. Man kann in komplexen Abfragen testweise auch nur einzelne Befehle ausführen. Man muß diese dann vor dem Ausführen bis einschließlich Semikolon markieren. Alles was in einer Zeile hinter zwei -- Zeichen steht, ist Kommentar und wird ignoriert.

Datenbankstruktur | Daten durchsuchen | Pragmas bearbeiten | **SQL ausführen**

Abfragen ausführen

```

1 SELECT Longitude, Latitude, Name , CountryIso3 AS Land
2 FROM PointOfInterest
3 WHERE CountryIso3 = 'ITA';
4

```

mit "AS" kann man der Spalte einen neuen Namen zuweisen

	Longitude	Latitude	Name	Land
1	10.81192	45.93592	Camping Lago di Tenno	ITA
2	10.61275	45.63342	Camping Promontorio	ITA
3	10.55359	45.56143	Camping Baia Verde	ITA
4	10.82306	45.78842	Camping Claudia	ITA
5	10.89294	45.93272	Zoo-Camping	ITA

hier steht das Ergebnis

Ergebnis: 66 Zeilen in 17ms zurückgegeben
In Zeile 1:
SELECT Longitude, Latitude, Name , CountryIso3 AS Land
FROM PointOfInterest
WHERE CountryIso3 = 'ITA';

hier steht das Erfolgsergebnis, oder in rot, die Fehlermeldung

Wenn wir in anderen Datenbanken nur die Daten von einem oder mehreren Ländern brauchen (hier haben wir nur ITA), können wir dies durch eine “WHERE-Klausel“ einschränken. Wir können sie aber auch weglassen. Die WHERE-Klausel ist eines der mächtigsten Auswahlwerkzeuge. Hier mal einige Beispiele. Mit ODER können wir z.B. mehrere Länder auswählen.

`WHERE CountryIso3 = 'DEU' OR CountryIso3 = 'AUT';`

Eine kürzere Schreibweise ist :

`WHERE CountryIso3 IN ('DEU', 'AUT', 'CHE');`

Sucht man nur nach einem Teilstück eines Tabelleninhaltes nimmt man z.B.:

`WHERE Name LIKE '%Camping%';`

Die Prozentzeichen stehen dabei für den Rest des Textes. Mein Lieblingsfehler, ist diese zu vergessen. Will man nur einen Buchstaben ersetzen, z.B. weil ein Wort mal groß- und mal kleingeschrieben wird, nimmt man den Unterstrich “_”.

Man kann die ganzen Befehle auch verneinen, indem man ein **NOT**, oder beim Gleichheitszeichen ein Ausrufungszeichen davor setzt **!=** (ungleich).

Es gibt noch jede Menge andere Möglichkeiten zum gezielten Auswählen von Daten. Dazu später in den Anwendungen.

Wenn Daten über mehrere Tabellen verteilt sind, füge ich diese immer zuerst in die Haupttabelle ein. Teilweise muß man auch Spalten der Haupttabelle selber kopieren, damit man sie weiter bearbeiten kann. Darum werden auch alle neuen Spalten als Text formatiert.

Es ist dann zwar keine regelkonforme Datenbank mehr, erleichtert aber die weiteren Arbeiten ungemein.

Da Sonderzeichen wie [“, >, <, &] in den meisten Navi-POIs zu Problemen führen, muß man sie entfernen oder umbenennen. (Wie hier in Strasse weiter unten). Da ich immer das Semikolon als Trennzeichen benutze, muß man auch dieses umbenennen, wenn es vorkommt.

Da einige Datenbanken ohne Dezimalpunkt arbeiten (Sygic), bekommen die Koordinaten hinterm Punkt immer fünf Stellen. Sie müssen deshalb als Text formatiert werden. Vor allen Dingen beim Import darauf achten!

Wir beginnen mal mit eine kompletten Durchlauf für unsere ACSI_Gardasee MasterDB

Wir nennen (am besten eine Kopie) die Datei "PointOfInterest.1.EURO.10" in "ACSI_Gardasee_MasterDB.db" um und öffnen sie im DB Browser.

--Wir fügen neue Spalten hinzu um alle Daten in einer Tabelle zu haben, alle formatiert als Text. Entweder leer (") oder schon befüllt ('CP')

```
ALTER TABLE PointOfInterest ADD COLUMN Longitude5 TEXT DEFAULT "";
ALTER TABLE PointOfInterest ADD COLUMN Latitude5 TEXT DEFAULT "";
ALTER TABLE PointOfInterest ADD COLUMN Platzanzahl TEXT DEFAULT "";
ALTER TABLE PointOfInterest ADD COLUMN Platztyp TEXT DEFAULT 'CP';
ALTER TABLE PointOfInterest ADD COLUMN Preis TEXT DEFAULT "";
ALTER TABLE PointOfInterest ADD COLUMN Offen TEXT DEFAULT "";
ALTER TABLE PointOfInterest ADD COLUMN Land TEXT DEFAULT "";
ALTER TABLE PointOfInterest ADD COLUMN Plz TEXT DEFAULT "";
ALTER TABLE PointOfInterest ADD COLUMN Ort TEXT DEFAULT "";
ALTER TABLE PointOfInterest ADD COLUMN Strasse TEXT DEFAULT "";
ALTER TABLE PointOfInterest ADD COLUMN Adresse TEXT DEFAULT "";
ALTER TABLE PointOfInterest ADD COLUMN Telefon TEXT DEFAULT "";
ALTER TABLE PointOfInterest ADD COLUMN Email TEXT DEFAULT "";
ALTER TABLE PointOfInterest ADD COLUMN Beschreibung TEXT DEFAULT "";
```

"NULL" Einträge in Spalten sind tödlich und müssen immer durch leer (" = zwei Hochkomma) ersetzt werden.
z.B. XY = " WHERE Platzanzahl IS NULL;

Dann kopieren wir die Daten aus den anderen Tabellen alle in die Tabelle PointOfInterest in die entsprechenden Spalten und nehmen gleich ein paar Anpassungen vor. Verknüpft werden die Tabellen über ihre Id.
Viele Sonderzeichen wie &, ", <, > und das Semikolon werden hierbei gleich ersetzt oder gelöscht.

```
UPDATE PointOfInterest SET Platzanzahl = (SELECT Value FROM PointOfInterestPropertyInt WHERE
PropertyIntId = 7 AND Id = PointOfInterestId);
UPDATE PointOfInterest SET Platzanzahl = " WHERE Platzanzahl IS NULL;
```

```
UPDATE PointOfInterest SET Preis = (SELECT Value FROM PointOfInterestPropertyDecimal WHERE
PropertyDecimalId = 1 AND Id = PointOfInterestId)|| '€';
UPDATE PointOfInterest SET Preis = " WHERE Preis IS NULL;
UPDATE PointOfInterest SET Preis = Replace(Preis, '.', ',') WHERE Preis != "";
```

```
UPDATE PointOfInterest SET Plz = (SELECT Value FROM PointOfInterestPropertyText WHERE
PropertyTextId = 3 AND Id = PointOfInterestId);
UPDATE PointOfInterest SET Plz = " WHERE Plz IS NULL;
```

```
UPDATE PointOfInterest SET Ort = (SELECT Value FROM PointOfInterestPropertyText WHERE
PropertyTextId = 4 AND Id = PointOfInterestId);
UPDATE PointOfInterest SET Ort = " WHERE Ort IS NULL;
```

```
UPDATE PointOfInterest SET Strasse = (SELECT Value FROM PointOfInterestPropertyText WHERE
PropertyTextId = 2 AND Id = PointOfInterestId);
UPDATE PointOfInterest SET Strasse = replace(Strasse, ';', ',') WHERE Strasse LIKE '%;';
UPDATE PointOfInterest SET Strasse = replace(Strasse, '&', 'und') WHERE Strasse LIKE '%&';
UPDATE PointOfInterest SET Strasse = " WHERE Strasse IS NULL;
```

```
UPDATE PointOfInterest SET Telefon = (SELECT Value FROM PointOfInterestPropertyText WHERE
PropertyTextId = 5 AND Id = PointOfInterestId);
UPDATE PointOfInterest SET Telefon = " WHERE Telefon IS NULL;
```

```
UPDATE PointOfInterest SET Email = (SELECT Value FROM PointOfInterestPropertyText WHERE
PropertyTextId = 6 AND Id = PointOfInterestId);
UPDATE PointOfInterest SET Email = " WHERE Email IS NULL;
```

--Öffnungszeiten sind etwas komplexer, da hier mehrere Spalten zusammen gefügt werden müssen

```
UPDATE PointOfInterest SET Offen = (SELECT '('||group_concat(substr(StartDate, 9, 2)||'.'||substr(StartDate, 6,
2)||'-'||substr(EndDate,9, 2)||'.'||substr(EndDate,6, 2), ', ')||')' FROM PointOfInterestPropertyDatePeriods WHERE
PropertyDatePeriodsID = 1 AND Id = PointOfInterestId GROUP BY PointOfInterestId);
UPDATE PointOfInterest SET Offen = " WHERE Offen IS NULL;
```

--Jetzt werden noch die Koordinaten formatiert.

```
UPDATE PointOfInterest SET Longitude5 = substr (longitude,1,instr(longitude,'.')->1)|| substr
(longitude||'00000',instr(longitude,'.'),6);
UPDATE PointOfInterest SET Latitude5 = substr (latitude,1,instr(latitude,'.')->1)|| substr
(latitude||'00000',instr(latitude,'.'),6);
```

Ich verwende für Land immer Iso2. Wer also das hier verwendete Iso3 auch umwandeln will, muß noch folgendes für jedes Land machen.

```
UPDATE PointOfInterest SET Land = 'IT' WHERE CountryIso3 = 'ITA';
UPDATE PointOfInterest SET Land = 'DE' WHERE CountryIso3 = 'DEU';
```

Für unsere ausgewählten und bearbeiteten Daten, erstellen wir jetzt eine neue Tabelle. Der Name und die Spaltennamen können frei vergeben werden.

```
CREATE TABLE MyPois (Longitude, Latitude, Preis, Name, Offen, VE, Platzanzahl, Platztyp, Telefon, Email,
Web, Nebenkosten, Beschreibung, Land, Plz, Ort, Strasse, Adresse);
```

--Beim Schreiben in die Tabelle müssen aber die Original-Spaltennamen genommen werden, und die Anzahl der Spalten muß überein stimmen. Nicht vorhandene Spalten werden durch leer (" = zwei Hochkomma) ersetzt. Wenn nötig, "CountryIso3" durch "Land" ersetzen.

```
INSERT INTO MyPois
SELECT Longitude5, Latitude5, Preis, Name, Offen, ", Platzanzahl, Platztyp, Telefon, Email, ", ", Beschreibung,
CountryIso3, Plz, Ort, Strasse, Adresse
FROM PointOfInterest;
```

Nicht mehr gebrauchte Tabellen werden gelöscht und die Datenbank bereinigt

```
DROP TABLE PointOfInterest;
DROP TABLE PointOfInterestPropertyDatePeriods;
DROP TABLE PointOfInterestPropertyDecimal;
DROP TABLE PointOfInterestPropertyInt;
DROP TABLE PointOfInterestPropertyText;
DROP TABLE PointOfInterestPropertyTextLanguage;
DROP TABLE PointOfInterestFacility;
DROP TABLE PointOfInterestPropertyBool;
DROP TABLE PointOfInterestPropertyIntList;
DROP TABLE PointOfInterestThumbnail;
```

```
VACUUM;
```

Die Abfrage bestätigen wir mit "Ja". Damit die Masterdatenbank ist fertig und wir könnten den DB Browser schließen. Auf das Speichern können wir verzichten, das ist schon erledigt.
Wir können aber auch gleich weitermachen.

Unter "Daten durchsuchen" sehen wir unsere neue Tabelle. (Eventuell vorher noch aktualisieren)

DB Browser for SQLite - C:\ACSI_Gardasee\02 FERTIGE Datenbanken\ACSI_Gardasee_MasterDB_1020.db

Datei Bearbeiten Ansicht Werkzeuge Hilfe

Neue Datenbank Datenbank öffnen Änderungen schreiben Änderungen rückgängig machen Projekt öffnen Projekt speichern Datenbank anhängen Datenbank schließen

Datenbankstruktur **Daten durchsuchen** Pragnas bearbeiten SQL ausführen

Tabelle: ACSI_Gardasee

	Longitude	Latitude	Preis	Name	Offen	VE	Platzanzahl	Platztyp	Telefon	Email	Web	Nebenkosten	Beschreibung	Land	Plz	Ort	Strasse	Adresse
Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern
1	10.81192	45.93592	31 €	Camping Lago di Tenno	(04.04-30.09)		61	CP	+393331072137	info@campinglagodi...			Camping Lago di ...	IT	38060	Tenno	Loc. Lago di Tenno 7	IT- 38060 Te...
2	10.61275	45.63342	38 €	Camping Promontorio	(08.04-12.10)			CP	+390365541540	info@campingprom...			Camping Promon...	IT	25088	Toscolano Mad...	Via Promontorio, 73	IT- 25088 Tos...
3	10.55359	45.56143	51 €	Camping Baia Verde	(04.04-10.10)			CP	+390365651753	info@campingbaiv...			Camping Baia Ve...	IT	25080	Manerba del Ga...	Via dell'Edera, 19	IT- 25080 Ma...
4	10.82306	45.78842	35 €	Camping Claudia	(04.04-01.11)		75	CP	+390457400786	info@campingclaud...			Camping Claudia ...	IT	37018	Malcesine	Via Gardesana 394	IT- 37018 Mal...
5	10.89294	45.93272	32 €	Zoo-Camping	(01.03-15.11)		213	CP	+390464516232	info@campingzoo.it			Zoo-Camping er ...	IT	38062	Arco	Via Legionari Ceco...	IT- 38062 Arc...
6	10.64053	45.45728	38 €	Camping Tiglio	(04.04-30.09)		168	CP	+390309904009	info@campingtiglio.it			Camping Tiglio er...	IT	25019	Sirmione	Loc. Punta Grò	IT- 25019 Sir...
7	10.53006	45.52003	63 €	Camping Piantelle	(04.04-12.10)			CP	+390365502013	info@piantelle.com			Camping Piantell...	IT	25080	Moniga del Garda	Via San Cassiano 1a	IT- 25080 Mo...
8	10.73165	45.48918	55 €	Camping Park Delle R...	(08.04-04.10)		414	CP	+390456471181	info@campingparkd...			Camping Park Del...	IT	37017	Lazise	Strada San Gaetan...	IT- 37017 Laz...
9	10.69450	45.44511	56 €	Camping Butterfly	(14.03-01.11)		33	CP	+390456401466	info@campingbutter...			Camping Butterfly...	IT	37019	Peschiera del G...	Lungo Lago Garibal...	IT- 37019 Pes...
10	10.82170	45.78428	36 €	Camping Lombardi	(03.04-01.11)		63	CP	+390457400849	info@campinglomb...			Camping Lombar...	IT	37018	Malcesine	Via Navene Vecchi...	IT- 37018 Mal...

Erstellen der Ersten POI-Dateien (drei Beispiele)

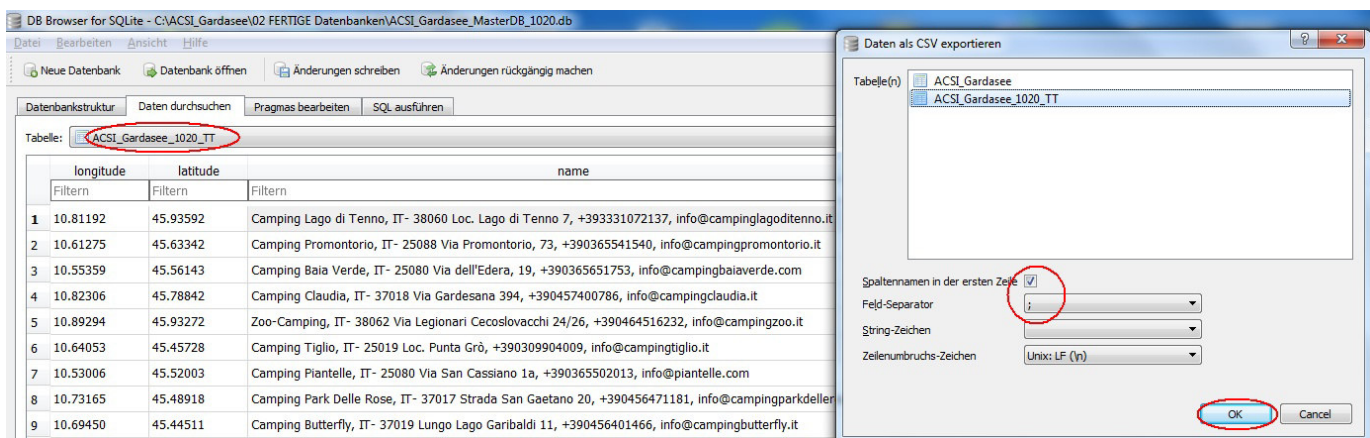
Beginnen wir mit TomTom. Hier im DB Browser erstellen wir erst mal nur die .csv Datei.

TomTom kennt nur Spalten für Koordinaten und Namen. Wir setzen deshalb die "Name" Spalte aus mehreren Spalten zusammen. Hierfür verwendet man zwei senkrechte Striche (Pipe). Die dazwischen zur Aufzählung verwendeten Komma und Leerzeichen kommen in Hochkommas. Die "Name" Spalte darf nicht mehr als 255 Zeichen enthalten. Dazu kürzen wir sie im Bedarfsfall wir mit `substr(Name, 1, 255)`.

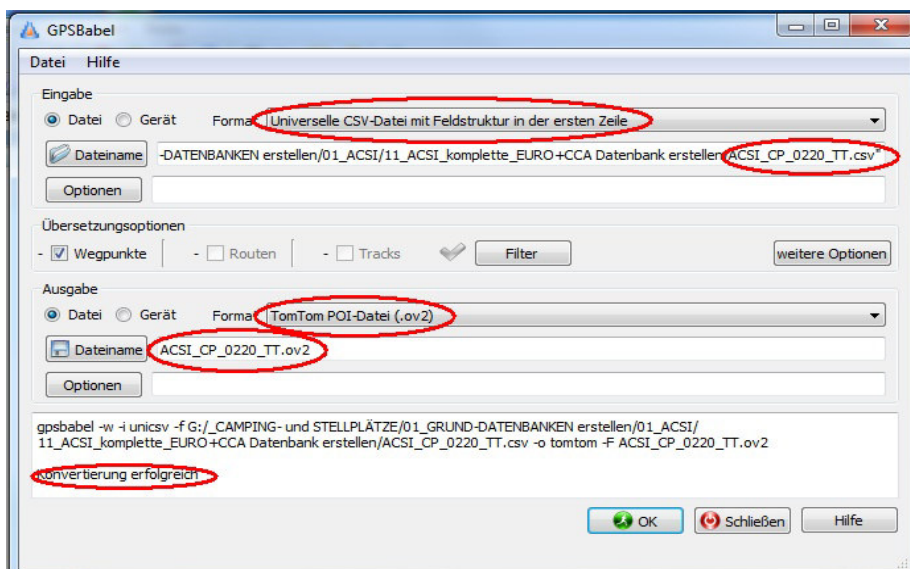
```
CREATE TABLE ACSI_Gardasee_1020_TT (longitude, latitude, name);
```

```
INSERT INTO ACSI_Gardasee_1020_TT
SELECT Longitude, Latitude, substr(Name || ' / ' || Adresse || ' / ' || Telefon || ' / ' || Email, 1, 255)
FROM MyPois;
```

Tabelle exportieren mit Kopfzeile anhängen und “;” als Trennzeichen. DB Browser ohne speichern schließen.



Datei in GPS Babel laden und konvertieren.



Hier für iGO

Wir erstellen eine neue Tabelle mit nur einer Spalte und fügen die einzelnen Zeilen der Reihe nach ein. Alle Daten inklusive <Tags>, Tabs/x'09' und Zeilenumbrüche/x'0A' werden zusammen in eine Spalte geschrieben und mit "||" verbunden.

"VALUES" schreibt feste Werte.

"SELECT" erzeugt immer einen kompletten <Placemark> Abschnitt mit den entsprechenden Daten aus der Tabelle MyPois der Master Datenbank.

Dabei können "name" und "description" nach eigenen Wünschen aus den vorhandenen Spalten der Tabelle zusammengestellt werden. ("trim" und "replace" dienen zum Entfernen überflüssiger Kommas wenn Daten fehlen). Die Zeilen "description" und "phoneNumber" kann man auch weglassen.

(Wer keinen Wert auf eine gut strukturierte "kml" Datei legt, kann die Zeilenumbrüche und Tabs auch weglassen.)

```
CREATE TABLE ACSI_Gardasee_iGO (KML TEXT NOT NULL);
```

```
INSERT INTO ACSI_Gardasee_iGO VALUES ('<?xml version="1.0" encoding="UTF-8"?><kml
xmlns="http://earth.google.com/kml/2.2">');
```

```
INSERT INTO ACSI_Gardasee_iGO VALUES ('<Document>');
```

```
INSERT INTO ACSI_Gardasee_iGO VALUES (x'09' || '<name>"Camping- u. Stellplätze"</name>');
```

```
INSERT INTO ACSI_Gardasee_iGO VALUES
```

```
(char(9) || '<metadata><igoicon><filename>eigene_pois.bmp</filename></igoicon></metadata>');
```

```
INSERT INTO ACSI_Gardasee_iGO VALUES ('<Folder>');
```

```
INSERT INTO ACSI_Gardasee_iGO VALUES (x'09' || '<name>ACSI_Gardasee_1020</name>');
```

```
INSERT INTO ACSI_Gardasee_iGO VALUES
```

```
(x'09' || '<metadata><igoicon><filename>ACSI_CampingPlatz.bmp</filename></igoicon></metadata>');
```

```
INSERT INTO ACSI_Gardasee_iGO SELECT ||x'0A09' || '<Placemark>'
```

```
||char(10)||char(9)||char(9)|| '<name>' || trim(Preis || ', ' || Name, ', ' ) || '</name>'
```

```
||char(10)||char(9)||char(9)|| '<description>' || trim(Platzanzahl || ' ' || Platztyp || ', ' || Offen, ', ' ) || '</description>'
```

```
||char(10)||char(9)||char(9)|| '<phoneNumber>' || Telefon || '</phoneNumber>'
```

```
||char(10)||char(9)||char(9)|| '<Point>'
```

```
||x'0A090909' || '<coordinates>' || Longitude || ', ' || Latitude || '</coordinates></Point>'
```

```
||char(10)||char(9)||char(9)|| '</Placemark>'
```

```
FROM MyPois;
```

```
INSERT INTO ACSI_Gardasee_iGO VALUES ('</Folder>');
```

```
INSERT INTO ACSI_Gardasee_iGO VALUES ('</Document>');
```

```
INSERT INTO ACSI_Gardasee_iGO VALUES ('</kml>');
```

Ohne Spaltennamen und Begrenzungszeichen exportieren als ACSI_Gardasee.kml ---> In Notepad kann/muß man ab Zeile 8 die automatisch erzeugten Anführungszeichen wieder löschen. (testen in GoogleEarth)
Datenbank ohne zu speichern wieder schließen.

Spaltennamen in der ersten Zeile ☐

Feld-Separator

Zeichenkette-Begrenzungszeichen

Zeilenumbruch-Zeichen

Save Cancel

Und auch noch für MapsMe

MapsMe kennt nur Koordinaten, Name und Beschreibung. In der Beschreibung (description) kann man Zeilenumbrüche (
) verwenden. Dann muß man aber in <description> die html Schreibweise verwenden. (<![CDATA[Text]]>)

Um die Farbe der Pins zu ändern, erstellen wir uns eine (temporäre) Spalte "Farbe", Voreinstellung "grün". Je nach Platztyp ändern wir diese für Stellplätze in blau oder für Parkplätze in braun. Beides in unser Testdatenbank aber nicht vorhanden.

```
ALTER TABLE MyPois ADD COLUMN Farbe TEXT DEFAULT 'green';
```

```
UPDATE MyPois SET Farbe = replace(Farbe, 'green', 'blue') WHERE Platztyp = 'SP%';
```

```
UPDATE MyPois SET Farbe = replace(Farbe, 'green', 'brown') WHERE Platztyp = 'PP%';
```

--Hier wieder die Tabelle, Beschreibung wie bei iGO

```
CREATE TABLE ACSI_Gardasee_MapsMe (KML TEXT NOT NULL);
```

```
INSERT INTO ACSI_Gardasee_MapsMe VALUES ('<?xml version="1.0" encoding="UTF-8"?><kml  
xmlns="http://earth.google.com/kml/2.2">');
```

```
INSERT INTO ACSI_Gardasee_MapsMe VALUES ('<Document>');
```

```
INSERT INTO ACSI_Gardasee_MapsMe VALUES (x'09'||<name>ACSI_Gardasee_1020</name>');
```

```
INSERT INTO ACSI_Gardasee_MapsMe SELECT x'09'||<Placemark>  
||x'0A0909'||<name>||trim(replace(Preis||', '||Name||', '||Adresse||', '||Offen, ', ', '), ', ')||</name>  
||x'0A0909'||<description>||<![CDATA[||replace(Platzanzahl||Platztyp||<br>||Telefon||<br>||Beschreibung,  
'<br><br>', '<br>')||]]>||</description>  
||char(10)||char(9)||char(9)||<styleUrl>#placemark-||Farbe||</styleUrl>  
||x'0A0909'||<Point>  
||x'0A090909'||<coordinates>||Longitude||', '||Latitude||</coordinates></Point>  
||x'0A0909'||</Placemark>  
FROM MyPois;
```

```
INSERT INTO ACSI_Gardasee_MapsMe VALUES ('</Document>');
```

```
INSERT INTO ACSI_Gardasee_MapsMe VALUES ('</kml>');
```

Das steht dann in einer Zeile.

Datenbankstruktur	Daten durchsuchen	Pragmas	Datenbankzelle bearbeiten
Tabelle: ACSI_CP_MapsMe	In alle...	Modus: Text	
KML			
Filtern			
1	<?xml version="1.0" encoding="UTF-8"?>		
2	<Document>		
3	<name>ACSI_CP_0121</name>		
4	<Placemark>...		
5	<Placemark>...		
1	<Placemark>		
2	<name>28.35 €, Camping Thüringer Wald, DE 07338 Drogitz, Mutschwiese 1 (30.03-30.10)</name>		
3	<description><![CDATA[250CP +49 1736466039 Campingplatz im Naturpark von Thüringen. Umgeben von viel Wald. Etwa 900m vom Hohenwarte Stausee.]]></description>		
4	<styleUrl>#placemark-green</styleUrl>		
5	<Point>		
6	<coordinates>11.56219,50.59586</coordinates>		
7	</Point>		
8	</Placemark>		
9			
10			