

SygiTruck POI Datenbank erstellen (poi.dat)

Vorbereitung (muss nur einmal gemacht werden)

Die "poi.dat" SQLite-Datenbank hat einen strukturellen Fehler. Ob mit Absicht, sei dahingestellt. Um auch die erweiterten Einträge schreiben zu können, müssen wir einmalig eine Anpassung vornehmen.

Zuerst benötigen wir eine unbenutzte "poi.dat".

Diese befindet sich:

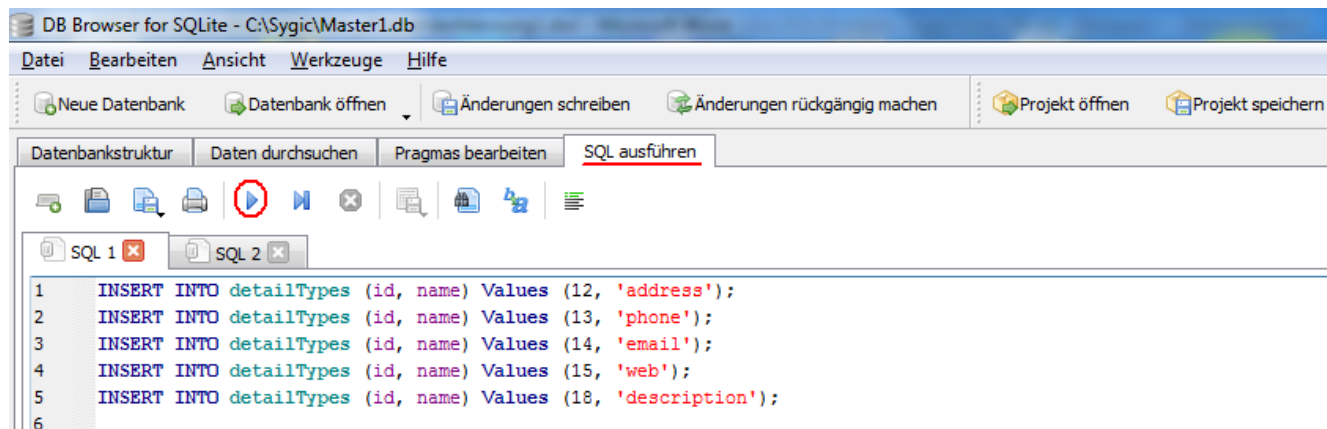
auf der Internen SD unter /storage/emulated/0/Android/data/com.sygi.truck/files/SygiTruck/Res/db,
oder auf der Externen SD unter /storage/1234-4321/Android/data/com.sygi.truck/files/SygiTruck/Res/db

Diese löschen wir und starten SygiTruck neu. Dann kopieren wir die neu erstellte leere "poi.dat" auf den PC. Sie wird unsere Sygi-Masterdatenbank.

Wir können auch eine schon vorhandene **originale**, schon mit POIs gefüllte Datenbank weiterbefüllen, das Verfahren ist das gleiche wie bei einer Leeren. Sie wird dann natürlich keine Masterdatenbank.

Wir öffnen die "poi.dat" im "DB Browser für SQLite" und führen folgende Anweisungen aus.

```
INSERT INTO detailTypes (id, name) Values (12, 'address');  
INSERT INTO detailTypes (id, name) Values (13, 'phone');  
INSERT INTO detailTypes (id, name) Values (14, 'email');  
INSERT INTO detailTypes (id, name) Values (15, 'web');  
INSERT INTO detailTypes (id, name) Values (18, 'description');
```



Diese Datenbank speichern wir jetzt als unsere **Sygi-Masterdatenbank** ab. (z.B. sygi_master.db)

Jede **neue** Sygi Datenbank erstellen wir aus einer Kopie dieser Masterdatenbank.

Die Icons müssen von **Hand** auf das Gerät kopiert werden.

Nach: "/storage/1234-4321/Android/data/com.sygi.truck/files/SygiTruck/Res/icons/rupi/"

Oder: "/storage/emulated/0/Android/data/com.sygi.truck/files/SygiTruck/Res/icons/rupi/"

1234-4321 ersetzen wir mit dem Namen unserer externen SD Card.

Die Icon-Namen müssen mit denen unter Punkt 3.) verwendeten übereinstimmen.

Jetzt erstellen wir noch ein Verzeichnis "C:\Sygi" und kopieren hier unsere vorher erstellten POI-Datenbanken rein.

1.) Wir Öffnen eine **Kopie** unserer Sygic-Masterdatenbank, oder eine schon erstellte Datenbank zum weiterbearbeiten, und importieren die (erste) POI Datenbank. Hier mal meine "Bordatlas MasterDB.db"

```
ATTACH DATABASE 'C:/Sygic/Bordatlas MasterDB.db' AS 'Master';
```

2.) Hieraus erstellen wir eine Hilfstabelle nur mit den gewünschten Pois und Daten (z.B. getrennt nach Ländern) Sie muß für jede POI-Datenbank, je nach vorhandenen Spalten, speziell angepasst werden. Wir brauchen jede Spalte. Wenn wir keine Daten haben, füllen wir sie mit "nichts" (""). Dadurch kann die Vorgehensweise immer gleich bleiben.

```
CREATE TABLE Import (Longitude, Latitude, Name, Adresse, Telefon, Web, Email, Beschreibung, Land);
```

```
INSERT INTO Import
SELECT longitude, latitude, trim(replace(Platzanzahl||Platztyp||', '||Preis||', '||Name||', '||Offen||', '||VE, ', ', ', ', '), Adresse,
Telefon, Web, Email, trim(Nebenkosten||', '||Beschreibung, ', '), Land
FROM Bordatlas
WHERE Land IN ('DE', 'AT', 'CH');
```

3.) In der Tabelle " **categories** " wird eine neue Spalte erstellt. Den Namen, den Icon Namen (hier Bordatlas) und den Namen der SD Card (hier 78D5-D50E) anpassen!!! Für die interne SD Card natürlich den ganzen Pfad. Der "Start" Eintrag und doppelte Einträge, falls wir den gleichen Kategorie-Namen noch mal benutzen sollten, werden wieder gelöscht.

```
INSERT INTO categories (id, name)
Values (999, 'dummy');
```

```
INSERT INTO categories (id, name, icon, isMain, color, poiOnRoute)
VALUES ((SELECT max(id)+1 FROM categories), 'Bordatlas', '/storage/ 78D5-
D50E/Android/data/com.sygic.truck/files/SygicTruck/Res/icons/rupi/Bordatlas.bmp', '1', '', '0');
DELETE FROM categories WHERE id = 999;
DELETE FROM categories WHERE rowid NOT IN (SELECT min(rowid) FROM categories GROUP BY name);
```


3a.) Statt automatisch erzeugte fortlaufende id's, können wir für die "id" auch feste Werte zwischen 1000 und 9999 verwenden. Wir müssen dann aber darauf achten, sie nicht doppelt zu vergeben. Sie werden sonst gelöscht. Dann brauchen wir nur diese Zeilen.

```
INSERT INTO categories (id, name, icon, isMain, color, poiOnRoute)
VALUES ('1234', 'Bordatlas', '/storage/ 78D5-
D50E/Android/data/com.sygic.truck/files/SygicTruck/Res/icons/rupi/Bordatlas.bmp', '1', '', '0');
DELETE FROM categories WHERE rowid NOT IN (SELECT min(rowid) FROM categories GROUP BY name);
```

4.) Jetzt füllen wir die Tabelle "**pois**". Hier löschen wir auch die Punkte aus den Koordinaten. (Sygic-Koordinaten müssen darum immer fünf Stellen hinterm Punkt haben. Sonst mit Nullen auffüllen. Diese Spalten immer als "TEXT" formatieren, sonst werden die Zahlen wieder gekürzt)

```
INSERT INTO pois (name, category, lat, lon, latEntry, lonEntry, iso, customerId, color)
SELECT name, (SELECT max(id) FROM categories), replace(latitude, '.', ''), replace(longitude, '.', ''), replace(longitude, '.', ''), Land, RowId, "
FROM Import;
```

Datenbankstruktur | Daten durchsuchen | Pragmas bearbeiten | SQL ausführen



SQL 1

```
3  /* In der Tabelle "categories" erstellen wir eine neue Spalte mit neuem "Namen" und neuem "Icon-Namen", löschen den Dummy-Eintrag und doppelte Einträge*/
4
5  Insert into categories (id, name, icon, isMain, color, poiOnRoute)
6  Values ((Select max(id)+1 From categories), 'KategorieName', /storage/emulated/0/SygicIcons/IconName.bmp, '1', '', '0');
7
8  DELETE FROM categories WHERE id = 999;
9  DELETE FROM categories WHERE rowid NOT IN (SELECT Min(rowid) FROM categories GROUP BY name);
10
11
12  /* Jetzt füllen wir die Tabelle "pois". Wenn die Spalte "Land" nicht leer ist, dann XX, XY, YY durch die gewünschten Ländernamen aus dieser Spalte ersetzen.
13  Bei einer leeren Spalte "Land" werden alle Länder genommen.*/
14
15  Insert into pois (name, category, lat, lon, latEntry, lonEntry, iso, customerId, color)
16  Select name, (Select max(id) From categories), Replace(latitude5, '.', ''), Replace(longitude5, '.', ''), Replace(latitude5, '.', ''), Replace(longitude5, '.', ''), 'EU_', RowId, ''
17  From Import
18  Where Land = '' Or Land In ('XX', 'XY', 'YY');
19
20  /* Und jetzt die Tabelle "poiDetails": */
21
22  Insert into poiDetails
23  Select pois.id, 12, Adresse
24  From Import, pois
25  Where pois.customerId = Import.RowId And (Select max(id) from categories) = pois.category And Import.Adresse != '';
```

5.) Und jetzt die Tabelle "poiDetails"

```
INSERT INTO poiDetails
SELECT pois.id, 12, address
FROM Import, pois
WHERE pois.customerId = Import.RowId AND (SELECT max(id) FROM categories) = pois.category AND Import.address != '';
```

```
INSERT INTO poiDetails
SELECT pois.id, 13, phone
FROM Import, pois
WHERE pois.customerId = Import.RowId AND (SELECT max(id) FROM categories) = pois.category AND Import.phone != '';
```

```
INSERT INTO poiDetails
SELECT pois.id, 14, email
FROM Import, pois
WHERE pois.customerId = Import.RowId AND (SELECT max(id) FROM categories) = pois.category AND Import.email != '';
```

```
INSERT INTO poiDetails
SELECT pois.id, 15, web
FROM Import, pois
WHERE pois.customerId = Import.RowId AND (SELECT max(id) FROM categories) = pois.category AND Import.web != '';
```

```
INSERT INTO poiDetails
SELECT pois.id, 18, description
FROM Import, pois
WHERE pois.customerId = Import.RowId AND (SELECT max(id) FROM categories) = pois.category AND Import.description != '';
```

6.) Zum Schluss müssen wir die Hilfstabelle wieder löschen und die Masterdatenbank aushängen

```
DROP TABLE Import;
```

```
COMMIT;
```

```
DETACH DATABASE 'Master';
```

```
VACUUM;
```

Speichern, bestätigen und fertig.

7.) Wir können hier Schluß machen, oder mit der nächsten POI Datenbank weitermachen.
Dazu tauschen wir überall im Text "Bordatlas" gegen den Namen der neuen Daten.

Dann führen wir wieder die Punkte **1.) – 6.)** aus.

(Wir können aber auch gleich meine fertigen Abfragen, hier "Bordatlas_SygiTruck_DB.sql" und "Campercontact_SygiTruck_DB.sql" direkt nach "SQL ausführen" kopieren, und nacheinander ausführen.)
Nicht vergessen, den SD Card Namen anzupassen.

Ps. Auf Wunsch kann ich auch die fertigen Abfragen zur Verfügung stellen.