

CAD-Dokumentation zu GIS mit SpatiaLite migrieren

Christoph Rinne

30. Juli 2022

Inhaltsverzeichnis

Vorwort	2
1 Einführung	2
1.1 Verwendete Software & Informationen	2
2 Wittelsberg 7	2
2.1 AutoCAD Quelldatei (dxf)	2
2.2 SpatiaLite GUI	4
2.2.1 Datenkontrolle	4
2.2.2 Variante: Viele Tabellen	5
2.2.3 Variante: Zwei wichtige Tabellen	5
2.2.3.1 Befundnummern	5
2.2.3.2 Befunde mit Schraffur	6
2.2.3.3 Profile	7
2.2.3.4 Schnitte	7
2.2.3.5 Steine oder andere Objekte	8
2.2.3.6 Text und Bockattribute	9
2.2.4 Am Ziel	11
Literatur	11

Vorwort

1 Einführung

Ziel ist die Überführung von Ausgrabungsplänen aus CAD-Dateien in ein GIS. Ausgang ist die gut strukturierte Retrodigitalisierung (2D) einer Papierdokumentation aus 2 Grabungskampagnen auf der Fundstelle Wittelsberg 7, Ldkr. Marburg Biedenkopf¹.

Anmerkung

- Menüpfade oder Abfolgen von Fenstern werden mit schlichten Pfeilen dargestellt: "Datei > Speichern".
 - Tastaturkürzel, die ich gerne nutze, stehen in Spitzklammern je Taste: <strg> + <c>.
 - Schalter auf Formularen werden in [] gesetzt: [OK]
 - Zur Darstellung von Befehlen im Text nutze ich die in Markdown übliche Darstellung von Code oder eben Anweisungen an den Computer: **anweisung**.
 - Der Text enthält viele Links die auf Papier nicht funktionieren. Sparen Sie bitte Papier und verzichten Sie auf den Ausdruck.
-

1.1 Verwendete Software & Informationen

- OS Windows 10
- QGIS 3.22.4-Białowieża Quelle: [<https://qgis.org>]
- SpatiaLite SpatiaLite GUI 2.1.0 beta1, SpatiaLite 5.0.0, SQLite 3.33.0, Quelle [<http://www.gaia-gis.it>]
- AutoCAD 2010, Quelle für aktuelle *kostenlose* Schulversionen: [<https://www.autodesk.de/education/edu-software/overview>]
- SpatiaLite Cookbook html [<http://www.gaia-gis.it/gaia-sins/spatialite-cookbook/index.html>]
- SpatiaLite Funktionen [<http://www.gaia-gis.it/gaia-sins/spatialite-sql-5.0.0.html>]

AutoCAD ist eine sehr komplexe Software und Ausgrabungen können eine komplexe Struktur annehmen, die es zu dokumentieren gilt. Erwarten Sie nicht, dass die notwendige Kompetenz beim Erstellen der digitalen Daten stets vorhanden war, auch der Autor (Chr. Rinne) ist hier nur Autodidakt.

Rechnen Sie mit Fehlern im originalen Datenbestand und einer ggf. nicht optimalen Struktur oder erwarten Sie nicht die von Ihnen bevorzugte Struktur. Korrektur von Fehler und Anpassungen der Struktur erfolgen sicher am besten im originalen Arbeitsumfeld, also CAD.

Neben AutoCAD gibt es teils kostengünstigere Alternativen, u.a.:

- BricsCAD [<https://www.bricsys.com>]
- MegaCAD [<https://www.megacad.de/>]

2 Wittelsberg 7

2.1 AutoCAD Quelldatei (dxf)

Es handelt sich um eine durch ein Grabensystem befestigte spätneolithische Siedlung der Wartberg Gruppe. Nach einigen Prospektionen erfolgte 1988 eine erste Ausgrabung durch das Landesdenkmalamt Hessen, Außenstelle Marburg und 1990 eine zweite Ausgrabung durch die Universität Marburg, Fachbereich Vor- und Frühgeschichte (Fiedler (1991)). Beide Grabungen wurden in der seinerzeit üblichen Form mit Theodolit vermessen und auf Papier dokumentiert. Für eine erneute Ausgrabung ab 2021 wurden alle alten Planzeichnungen in einer CAD-Zeichnung fortlaufend digitalisiert. Das alternative Modell, je Zeichenblatt eine Zeichnung und nachfolgende Zusammenstellung zu 4 Gesamtzeichnungen bzw. für vier Plana wird am Bsp. von Odagsen durchgeführt (s.o.).

¹Mit freundlicher Genehmigung des Landesdenkmalamtes Hessen, Außenstelle Marburg.

Die CAD-Datei wurde in AutoCAD 2010 (Schulversion) und 2010 bearbeitet, sie beinhaltet einen entsprechenden Hinweis. Die Layernamen der Zeichnung sind sehr normiert (s. Abb. 1) und es werden zahlreiche Blöcke auch mit Attributen verwendet.

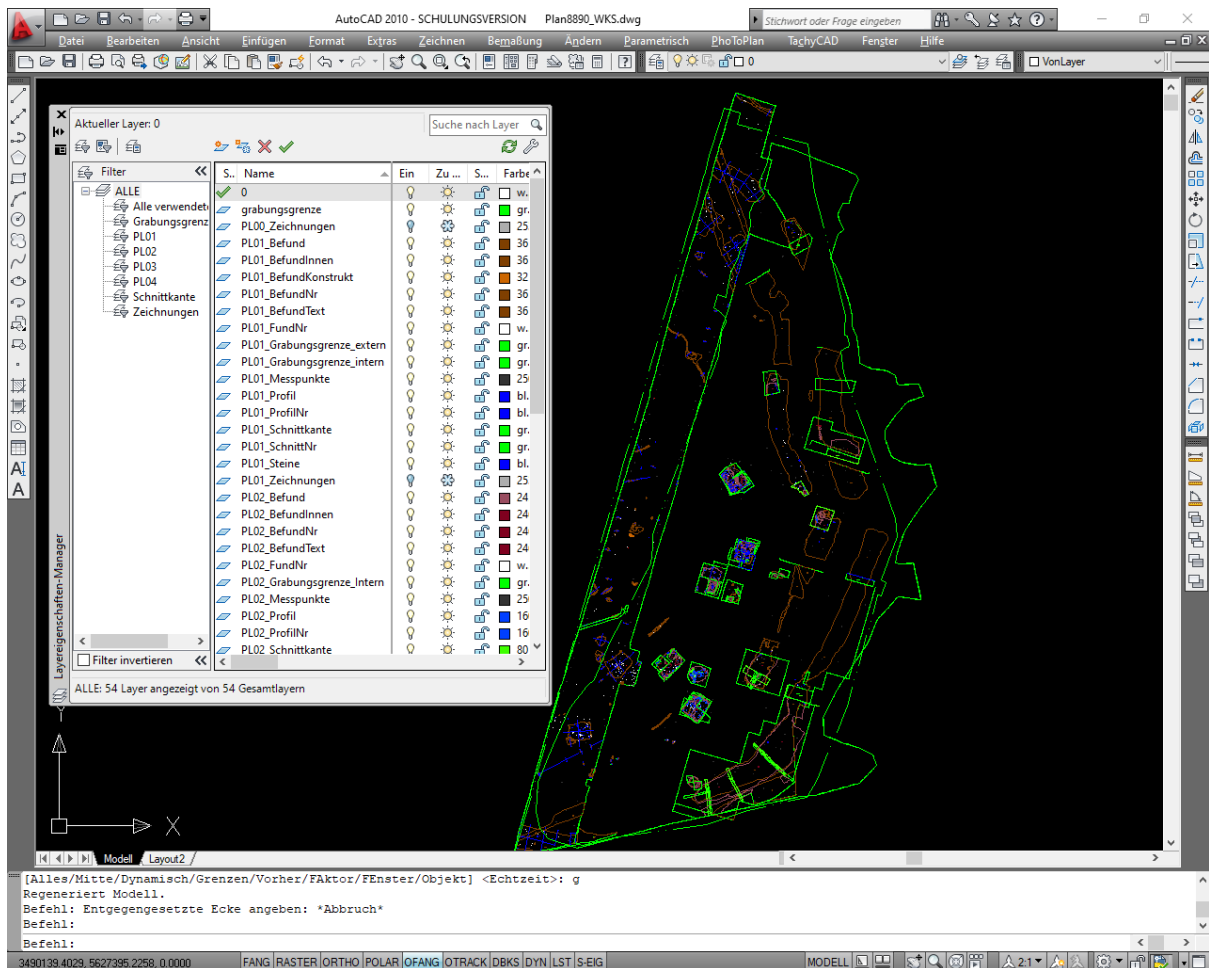


Abbildung 1: Abb. 1 CAD-Datei Wittelsberg 7 in AutoCAD

Die Zeichnung enthält 1230 Polylinien (2d), 72 Linien, 18 Schraffuren, 1945 Blockreferenzen und 420 Textelemente. Die externen Referenzen auf die zahlreichen Papierzeichnungen wurden entfernt und sind nicht Teil der dxf-Datei.

Name	Farbe	Linientyp
0	weiß	Continuous
grabungsgrenze	grün	ACAD_ISO12W100
PL01_Befund	36	Continuous
PL01_BefundInnen	36	Continuous
PL01_BefundKonstrukt	32	Continuous
PL01_BefundNr	36	Continuous
PL01_BefundText	36	Continuous
PL01_FundNr	weiß	Continuous
PL01_Grabungsgrenze_extern	grün	ACAD_ISO12W100
PL01_Grabungsgrenze_intern	grün	ACAD_ISO10W100
PL01_Messpunkte	250	Continuous
PL01_Profil	blau	Continuous
PL01_ProfilNr	blau	Continuous
PL01_Schnittkante	grün	ACAD_ISO02W100
PL01_SchnittNr	grün	Continuous

Name	Farbe	Linientyp
PL01_Steine	blau	Continuous
PL02_Befund	245	Continuous
...		
PL03_Befund	245	Continuous
...		
PL03_BefundNr	247	Continuous
PL04_Befund	245	Continuous
...		

Diese eindeutige Nomenklatur ist von großem Vorteil für die nachfolgende Organisation der Daten in einer SpatiaLite Datenbank. Sie erlaubt es mit einer SQL-Anweisung alle Filterbedingungen zu formulieren. Bei einer heterogenen Nomenklatur müssen ggf. zwei *update*-Anweisungen formuliert oder mit komplexeren *where*-Anweisungen agiert werden. Für die Visualisierung in QGIS kann auf die Planumsangabe im Layernamen sehr einfach zugegriffen werden. Alternativ muss mit einer ergänzenden Spalte zum Planum operiert werden (s.u.).

Wichtig Innerhalb von jedem Befund steht als Text und als Block mit Attribut eine Befundnummer, an jeder Profillinie steht ein Text mit der Bezeichnung des Profiles oder des jeweiligen Endpunktes. Diese Informationen werden in GIS räumlich verbunden.

2.2 SpatiaLite GUI

Nach dem Start der GUI bitte "Menu > Create a new (empty) SQLite DB" ausführen. In diese wird die dxf-Datei mit der gesamten Ausgrabung importiert: "Menu > Advanced > Import DXF drawings". Im Fenster zum Import bitte folgende Angaben: (x) Import selected DXF drawing file only, SRID: 31467, [v] Append to already existing tables, Dimensions: (x) automatic 2D/3D, (x) mixed layers, Special Rings handling (x) none. Diese Angaben beziehen sich auf den aktuellen Import und müssen ggf. angepasst werden. Die Option 'mixed layers' trennt nur die Typen (Punkt, Linie Polygon, Text), die Layer werden als Attribut (Spalte) angelegt. Die Alternative trennt erst die Layer und dann nach Typen, erstellt also das x-Fache der vorhandenen CAD-Layer als Tabellen.

Als Ergebnis sind folgende Tabellen überwiegend mit den Spalten *feature_id*, *filename*, *layer* und *geometry* als auch und eine *view* (gespeicherte Abfrage) vorhanden:

- hatch_layer_2d (benötigt)
- hatch_layer_2D_pattern (nicht benötigt)
- line_layer_2d (benötigt)
- polyg_layer_2d (benötigt)
- text_layer_3d (benötigt)
- text_layer_3d_attr (nicht benötigt)
- text_layer_3d_view (nicht benötigt)

2.2.1 Datenkontrolle

Prüfen Sie die jeweils resultierende Geometrie, vor allem ob MULTI-Typen vorliegen, z.B. MULTIPOLYGON. Das ist hier nicht der Fall und damit können Teilbefunde nicht zu einem kombinierten Objekt verschmolzen werden. Bei Bedarf lautet die Funktion in SpatiaLite: [CastToMulti](#). An dieser Stelle ein schnelle Kontrolle des Datenbestandes durch eine Tabellierung der Layer in jeder Tabelle

```
select layer, geometrytype(geometry) as GeoType, count(*) as n
from hatch_layer_2d
group by 1, 2
```

Die Syntax aggregiert auf die erste und zweite Spalte, den Layernamen und die Abfrage des Geometrietyps. Dazu liefert sie die Anzahl der Einträge. Warum einige Geometrien nicht vorliegen (NULL) kann ich nicht sagen. Im vorliegenden Fall handelt es sich durchgängig um eine Kreuzschaffur (ANSI37) für Tiergänge oder andere Störungen.

```
select layer, geometrytype(geometry) as GeoType, count(*) as n
from line_layer_2d
group by 1, 2
```

Diese Abfrage liefert im vorliegenden Fall zwei Problemgruppen: PL01_Steine hat elf und PL02_Befunde hat zwei Linienobjekte. In allen Fällen sollten das Polygone sein. Die Korrektur erfolgt am besten in CAD: beim Layer Steine ist es ein Stein der aus zehn einzelnen Linien besteht und beim Layer Befunde sind es zwei nicht geschlossene Befunde. Die Grabungsgrenzen sind im vorliegenden Fall aufgrund der Vorgaben heterogen sowohl als Polygon als auch als Linie vorhanden. Hier ist die jeweilige Fragestellung relevant: z.B. Fläche bemessen oder 1:1-Abbildung des Papiers.

Die Tabelle `text_layer_3d_attr` und die Sicht (*view*) `text_layer_3d_attr_view` liefern in diesem Fall keine erhellenden Informationen und können wie `hatch_layer_2d` gelöscht werden.

2.2.2 Variante: Viele Tabellen

Eine Option sowohl die Informationen aus 'polyg_layer_2d' nach Befunden, Grabungsgrenzen, Steinen etc. zu trennen und mit dem Feld 'label' aus 'text_layer_3d' zu verbinden kann über Abfrage simuliert und dann als *create-table*-Anweisung umgesetzt werden. Dies führt zu sehr vielen Tabellen und ist hier nicht Ziel des Vorgehens.

```
select a.*, b.label
from polyg_layer_2d as a
join text_layer_3d as b on st_within(b.geometry, a.geometry)
where a.layer = 'PL01_Befund'
and b.layer = 'PL01_BefundNr';
```

Wird der vorangehenden Abfrage ein `create table as PL01_BefundeNr` vorangestellt wird die entsprechende Tabelle erstellt. Nach einem *refresh* der Datenbank ist diese Tabelle sichtbar. Die Angaben zur dort enthaltene Geometrie kann erst mit 'Check geometries' aus dem Kontextmenü überprüft und dann mit 'Recover geometry column' wieder hergestellt werden. Vergleichbare Vorgänge für jede Informationseinheit wiederholen.

Das ist die arbeitsreiche und in der Datenhaltung schwierigere Variante wegen der zunehmend vielen Tabellen. Das ist nicht zu empfehlen. Der eigentliche Vorteil: der Lernaufwand liegt nahe NULL.

2.2.3 Variante: Zwei wichtige Tabellen

Bei dieser Variante wird die Tabelle 'polyg_layer_2d' um zwei sinnvolle Spalten für wichtige Informationen erweitern: (Befund-)Nummer und Info. Ebenso für 'line_layer_2d'. Dabei gilt: unbedingt die folgenden Updates bei exklusivem Zugriff von SpatiaLite ausführen und nachfolgend im Kontextmenü der Geometriespalte 'Update Layer Statistics' aufrufen.

Anmerkung: SQLite hat *type affinity*, trivialisierte Info: in einem Feld vom Typ *integer* kann auch '8820A' gespeichert werden.

```
-- Mit BEGIN TRANSACTION ... COMMIT werden mehrere Anweisung gebündelt --
BEGIN TRANSACTION;
ALTER TABLE "polyg_layer_2d" ADD COLUMN number integer;
ALTER TABLE "polyg_layer_2d" ADD COLUMN info text;
ALTER TABLE "line_layer_2d" ADD COLUMN number integer;
ALTER TABLE "line_layer_2d" ADD COLUMN info text;
UPDATE geometry_columns_statistics set last_verified = 0;
SELECT UpdateLayerStatistics();
COMMIT
```

2.2.3.1 Befundnummern Nun die Befundnummer mittels *spatial join* nachgeschlagen und in dem Feld 'number' eingetragen.

```
update polyg_layer_2d as c
set number = (select b.label
from polyg_layer_2d as a
```

```

join text_layer_3d as b on st_within(b.geometry, a.geometry)
where a.layer = 'PL01_Befund'
    and b.layer = 'PL01_BefundNr'
    and a.feature_id = c.feature_id)
where c.number is null;

```

Diesen Vorgang für alle Plana wiederholen. Wenn der Text neben dem Befund steht ist eine Option zunehmende Puffer um den Punkt des Textes (Befundnummer) zu rechnen und dann st_intersect zu nutzen. Sollte das Feld 'number' noch NULL sein wird es sukzessive gefüllt:

```

update polyg_layer_2d as c
set number = (select b.label
from polyg_layer_2d as a
join text_layer_3d as b on st_intersects(st_buffer(b.geometry,0.2), a.geometry)
where a.layer = 'PL01_Befund'
    and b.layer = 'PL01_BefundNr'
    and a.feature_id = c.feature_id)
where c.number is null;

```

Nicht alles wird korrekt laufen. In QGIS die Beschriftung der Befunde einrichten, kontrollieren und die Probleme von Hand analog der folgenden Anweisung bereinigen. Die "Massenzuweisung" sind überwiegend die vielen einzelnen Befunde der tiefern Plana die nicht individuell benannt wurden. Optional zur Handarbeit über die Liste von ID's könnte auch mit einem großen Puffer um die Befundnummer (s.o.) gearbeitet werden. Allerdings sollte ein Pfosten in einem Grubenhaus besser eine neue Befundnummer erhalten oder die einzelnen Quadranten einer Grube im Planum 1 sollten doch eher zu einem Befund verschmolzen werden. Dies sind individuelle und situativ zu treffende Entscheidungen.

```

BEGIN TRANSACTION;
update polyg_layer_2d
set number = '9013'
where feature_id = 111;
update polyg_layer_2d
set number = '20B'
where feature_id = 80;
update polyg_layer_2d
set number = '9011'
where feature_id in (438, 439, 814, 815, 817, 818, 819, 820);
update polyg_layer_2d
set number = '9009'
where feature_id in (102, 444, 445, 446, 812, 813, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009);
COMMIT

```

Abschließend ggf. die 0-Nummern wieder auf NULL setzen.

```

update polyg_layer_2d
set number = NULL
where number = 0

```

Für die Darstellung der Befunde über alle Plana in QGIS kann die 'polyg_layer_2d' der Karte hinzugefügt werden. Objektfiler setzen auf "layer" like 'PL0%_Befund', dann bei der Symbologie auf 'Abgestuft' wechseln und als Wert mit substr("layer",4,1) nur die Planumsnummer aus dem standardisierten Layernamen "Pl01_Befunde" holen. Alles klassifizieren und bei Legendenformat: "Planum %%2" eintragen.

2.2.3.2 Befunde mit Schraffur Im vorliegenden Fall sind die Tiergänge und andere Störungen schraffiert. Leider sind im vorliegenden Fall einige fehlerhafte Geometrien dabei. In der Spalte 'info' wird ein "TG" ergänzt wenn die feature_id in der Liste der Befunde mit einer innenliegenden Schraffur steht.

```

update polyg_layer_2d
set info = 'TG'
where layer like '%_Befund' and
feature_id in (select a.feature_id

```

```

from polyg_layer_2d as a
inner join hatch_layer_2d as b on st_within(centroid(b.geometry),a.geometry)
where GeometryType(b.geometry) not null);

```

Die Kontrolle wegen der fehlenden Geometrien zeigt eine Fehlzuzuweisung wegen der Lage des Tierganges in einem anderen Befund. Deshalb die folgende Korrekturen:

```

BEGIN TRANSACTION;
update polyg_layer_2d
set info = NULL
where feature_id = 69;
update polyg_layer_2d
set info = 'TG'
where feature_id in (71, 72, 34, 31, 47, 22, 28);
COMMIT

```

2.2.3.3 Profile Bei den Profilen müssen die Linien mit Namen versehen werden.

1. Schritt alle Profillinien bekommen die Nummer des Befundpolygons das sie berühren (= zugehörige Befundnummer).

```

update line_layer_2d as b
set number = (select a.number
from polyg_layer_2d as a
where intersects(a.geometry,b.geometry) and
a.layer like 'PL%_Befund')
where number is null;

```

2. Schritt, die Nagelnamen, z.B. A, B, werden in die Info-Spalte geschrieben.

```

update line_layer_2d
set info = (select group_concat(b.label, ' - ')
from line_layer_2d as a
inner join text_layer_3d as b on intersects(b.geometry,a.geometry)
where a.layer='PL01_Profil' and b.layer='PL01_ProfilNr'
group by a.feature_id
having a.feature_id=line_layer_2d.feature_id)
where info is null

```

Optional für das schnelle Abfragen der Profilinfos in den Layereigenschaften von QGIS folgende “Anzeige” (pop ups, ballon, Sprechblase) bei den HTML-Kartenhinweisen definieren:

```

<p>Befund-Nr.: [%"number"%] <br>
Nägel: [%"info"%]<br>
Pl.: [%substr("layer",4,1)%]</p>

```

2.2.3.4 Schnitte Bei den Schnitten sind es die Layer mit ‘Grabungsgrenze’ oder ‘Schnitt’ die bearbeitet werden müssen. Es sind z.B. die Dokumentationsgrenzen um die jeweiligen Befunde oder auch die “echten” Schnitte, z.B. durch den Graben. Beides wird hier erstmal zusammengefasst behandelt. Die Update-Anweisung beinhaltet mehrfach verschachtelte Unterabfragen.

```

update polyg_layer_2d as c
set info = (select b.label
from (select *
from polyg_layer_2d
where layer like '%Grabungsgrenze%' or layer like '%Schnitt%') as a
inner join (select *
from text_layer_3d
where layer like '%SchnittNr') as b
on st_within(b.geometry,a.geometry)
where c.feature_id = a.feature_id)
where info is null;

```

Im vorliegenden Fall sind die umfassenden Schnitte der layer ‘Grabungsgrenze_extern’, ‘..._intern’ oder ‘grabungsgrenze’ falsch beschriftet. Handarbeit ist hier die schnellere Alternative.

```
BEGIN TRANSACTION;
update polyg_layer_2d
set info = 'Grabungsgrenze 1990 Dia 1451-5'
where feature_id = 424;
update polyg_layer_2d
set info = 'Grabungsgrenze Fiedler 1991'
where feature_id = 425;
update polyg_layer_2d
set info = 'Grabungsgrenze 1988 CAD Plan'
where feature_id = 147;
COMMIT
```

Im vorliegenden Fall ist ein **ärgerlicher Fehler** dabei, die konstruierte externe Grabungsgrenze um die gesamte Fläche von 1989 nach dem CAD-Plan und einem Luftbilddia von 1990 war nicht geschlossen und ist bei den Linien gelandet. Folgende Schritte zur Lösung dieses oder vergleichbarer Fälle:

```
BEGIN TRANSACTION;
-- Problem identifizieren:
select *, st_length(geometry) from line_layer_2d
where layer like 'Pl01_G%';
-- Zur Kontrolle, Linie zu Polygon konvertieren:
-- Hier wird der LINESTRING-WKT zu einem POLYGON-WKT umgeschrieben.
select st_geometrytype(geomfromtext('POLYGON (' ||
  substr(st_astext(geometry),11) ||')')) as geo
from line_layer_2d
where feature_id=148;
-- Problem lösen durch INSERT-Anweisung:
-- Die Geometrie wird um die SRID ergänzt, dazu noch die weiteren Felder
insert into polyg_layer_2d (filename, layer, geometry)
select filename, layer, geomfromtext('POLYGON (' ||
  substr(st_astext(geometry),11) ||')', 31467) as geo
from line_layer_2d
where feature_id=148;
-- Kontrolle & Info ergänzen:
select * from polyg_layer_2d
where feature_id = (select max(feature_id)
  from polyg_layer_2d);
update polyg_layer_2d
set info = 'Grabungsgrenze 1989/90 extern'
where feature_id = (select max(feature_id)
  from polyg_layer_2d);
COMMIT
```

2.2.3.5 Steine oder andere Objekte Zahlreiche **Steine** sind ebenfalls in der Zeichnung als Polygone digitalisiert und liegen je Planum auf einem eigenen Layer. Damit ist die Aktualisierung der Spalte ‘Info’ sehr einfach.

```
update polyg_layer_2d
set info = 'Stein'
where layer like '%Steine' and info is null;
```

Dazu soll aber in der Spalte ‘number’ die zugehörige Befundnummer ergänzt werden.

```
update polyg_layer_2d as c
set number = (select a.number
  from (select number, layer, geometry from polyg_layer_2d
```



```

where layer like '%Befund' and number not null) as a
inner join (select feature_id, info, geometry
  from polyg_layer_2d where info = 'Stein') as b
on st_relate(b.geometry, a.geometry)
where c.feature_id = b.feature_id)
where info = 'Stein';

```

Einige wenige Steine liegen außerhalb eines Befundes in jeweils einem Fall in Planum 2 (Bef. 9011) und 3 (9003). Da es sich beim anstehenden Boden um Löss handelt ist die Abgrenzung der Befunde offensichtlich schwierig gewesen und muss bedingt hinterfragt werden. Im Planum 1 können an der Grenze noch Steine des Pflughorizontes vorkommen.

2.2.3.6 Text und Blockattribute Blöcke mit Attributen in CAD sind ein wichtiges Element in CAD-Zeichnungen mit zahlreichen Optionen und einem enormen Mehrwert, z.B. einem einfachen Export in Tabellen mit zahlreichen weiteren Informationen. Dies ist die Optimale Möglichkeit, diese Informationen zu Sichern und nachnutzbar zu machen. Leider sind beim Import der DXF-Datei in der SpatiaLite-GUI diese nicht übernommen worden. Es sind im vorliegenden Fall Blöcke für die Niv-Werte, die Holzkohle, den Rotlehm und die Fundnummern vorhanden.

Blöcke in AutoCAD können einen unterschiedlichen Grad an Komplexität haben, der Import in ein GIS ist deshalb nicht trivial und kann jeweils zu unterschiedlichen Ergebnissen führen. Allgemein werden Objekte in einer Blockdefinition auf dem Layer '0' angelegt. Beim Einfügen werden die Objekte dann dem jeweils aktiven Layer zugewiesen und erhalten auch dessen Attribute, z.B. die Farbe. Beim Import werden diese komplexen Strukturen unterschiedlich aufgelöst aber immer als Punkt-Objekt des Einfügekpunktes in AutoCAD behandelt. In QGIS zeigt die Attributtabelle zu den importierten Punktoobjekten etwa folgendes:

layer	subclasses	entityhandle	text
PL01_BefundNr	AcDbEntity:AcDbBlockReference	77	
PL01_BefundNr	AcDbEntity:AcDbText:AcDbText	1665	9021
0	AcDbEntity:AcDbText:AcDbAttribute	461	9021
0	AcDbEntity:AcDbText:AcDbAttribute	113F	9005-Keramik
0	AcDbEntity:AcDbText:AcDbAttribute	6F7	03/04/2021
0	AcDbEntity:AcDbText:AcDbAttribute	BD4	207,95
PL01_FundNr	AcDbEntity:AcDbText:AcDbText	2CF	BL

Die ersten drei Zeilen der Tabelle zeigen die Daten für den Block der Befund-Nr 9021: 1. die eigentliche Blockreferenz, 2. den mit dem Block auch eingefügten Text und 3. das Attribut des Blockes. Letzteres wird aber aufgelöst auf den ursprünglichen Layer '0' der Layerdefinition abgelegt. Das gleiche passiert auch mit den Attributen der Fundnummern, den Datumsangaben der Erfassung und den Niv-Werten. Nur die als Text und nicht als Attribut oder grafisches Symbol eingefügten Hinweise auf Brandlehm (BL) werden auf dem jeweiligen Layer aufgelöst.

Bis auf die Informationen zur Lage von Brandlehm und Holzkohle sind diese Informationen ohne den Bezug zum Planum bzw. dem ursprünglichen Layer in CAD unbrauchbar. Die DXF-Import-Erweiterung "[AnotherDXFImporter](#)" bietet faszinierende Möglichkeiten, ist bei dieser Aufgabe aber auch keine bessere Option.

Die Lösungen für das Import-Problem der CAD-Blöcke können sein:

1. Kontrollieren Sie die vorliegende digitale Dokumentation, wer mit Blöcken in CAD arbeitet weiß meist was er tut und exportiert diese Daten in CSV-Tabellen.
2. Nehmen Sie den Export der Blockinformationen aus AutoCAD selber vor, ggf. unter Verwendung einer kostengünstigeren Alternative wie BricsCAD. AutoCAD gibt es für die Lehre auch 1 Jahr kostenlos und BricsCAD als Demo-Version.
3. Importieren Sie die originale DWG- oder DXF-Datei in Esri ArcMap, dort werden Blöcke mit Attributen korrekt zu Punkten mit entsprechenden Attributen aufgelöst.

Eine in CSV exportierte Blockliste kann so aussehen:

Name	Layer	NUMMER	Position X	Position Y	Position Z
Brandlehm	PL01_FundNr		3490060.7999	5627322.4548	0.0000
Fund	PL01_FundNr	9005-Keramik	3490067.9820	5627321.1025	0.0000

Für jeden Block wird der Name des Blocks angegeben, der Layer, alle Attribute als getrennte Spalten, z.B. Nummer und die Koordinaten. Der Import mit folgender Kartierung und Auswertung sind damit leicht möglich.

Wichtig: AutoCAD verwendet den “.” als Dezimaltrennzeichen, das “,” ist der Spaltentrenner. Ein Text mit “,” wie z.B. “210,15 m üNN” ohne eine Textauszeichnung (“”) führt dies zu einem Spaltenfehler. Verwenden Sie z.B. [RegExp](#) in [Notepad ++](#), um dieses Problem effizient zu lösen.

Der **Import der CSV-Tabelle** mit Daten ist über das entsprechende Icon einfach möglich. Beachten Sie im folgenden Fenster 1. den Tabellennamen (z.B. block_attr_3d), 2. [x] First line ..., 3. Column separator (x) Comma, 4. Decimal separator (x) Point, 5. Charset Encoding (vermutl. UTF-8). Nachfolgend wird eine Geometriespalte ergänzt und aus den Daten gefüllt. Bitte beachten: anders als in GIS sind Punkte in CAD immer 3D, nur ggf. mit dem Z-Wert 0.

```
BEGIN TRANSACTION;
-- Geometriespalte ergänzen
SELECT AddGeometryColumn('block_attr_3d', 'geometry', 31467, 'POINTZ', 'XYZ');
-- Geometriespalte mit dem WKT aus den Koordinaten füllen.
update block_attr_3d as a
set geometry = (select st_geomfromtext(('POINTZ(' || "Position_X" || ' ' ||
"Position_Y" || ' ' || "Position_Z" || ')'), 31467)
from block_attr_3d
where a.PK_UID = PK_UID)
```

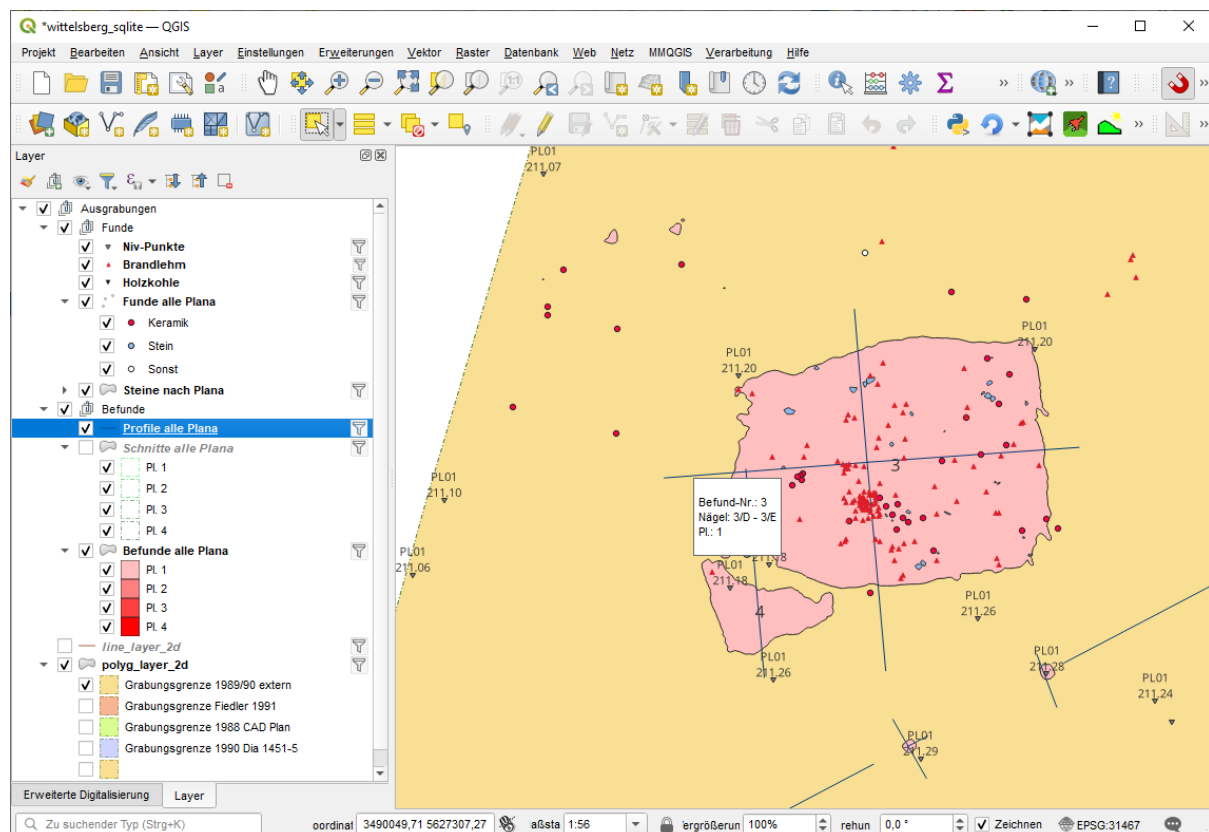


Abbildung 2: Abb. 2 Das fertige Projekt in QGIS

2.2.4 Am Ziel

Die QGIS-Datei ist eine gepackte (zip) XML-Datei (.gpz, .gps). Grundsätzlich werden in dieser nur die Anweisungen und nicht die Daten hinterlegt. QGIS-Projekte speichern per Definition als Standard relative Pfadangaben zu den Daten. Dies funktioniert bei der SpatiaLite-Datei nicht. Soll dies spontan auch auf anderen Systemen funktionieren, wobei Datenbank und QGIS-Datei in einem Ordner liegen, dann editieren Sie (eine Kopie!) der entpackten Projektdatei und ersetzen Sie die absolute Pfadangabe durch eine relative, z.B.:

“D:/mein/absoluter/Pfad/Wittelsberg7.sqlite” > “./Wittelsberg7.sqlite”

Literatur

Fiedler, L., 1991. Eine Befestigte Siedlung Der Jungsteinzeit Bei Wittelsberg, Kreis Marburg-Biedenkopf. Zur Archäologie Des 4. Und 3. Jahrtausends Vor Christus. Denkmalpflege in Hessen 1991, 23–27.