

# TachyGIS Geodatenstruktur V2.0.17 (Stand 2025-02)

---

## Zusammenfassung

TachyGIS ist eine Idee zur archäologischen Grabungsvermessung mit Tachymeter und GIS, die den bisherigen CAD-zentrierten Ansatz auf GIS überträgt. Dabei werden die Messwerte eines Tachymeters gemeinsam mit den für eine adäquate Plandarstellung erforderlichen Informationen in eine spezifische Geodatenstruktur übernommen. Eine auf die Anforderungen archäologischer Grabungen ausgerichtete TachyGIS-Geodatenstruktur wird in diesem Artikel beschrieben.

Die Datenstruktur ist bewusst einfach und flach gehalten, damit sie möglichst unkompliziert und nachhaltig genutzt werden kann. Sie konzentriert sich auf die in Grabungsplänen üblichen Zeichnungsobjekte. Die dazu benutzten Attribute werden ausführlich beschrieben. Diese Geodatenstruktur wird im QGIS-Plugin „TachyGIS\_arch“ verwendet. Die Anbindung von zusätzlichen Attributen oder von ganzen Grabungsdatenbank-Strukturen ist möglich, wird aber hier nicht detailliert behandelt.

Die hier vorgelegte Datenstruktur ist auf Tachy2GIS\_arch Version 2 (mit Geopackage) ausgelegt.

## Inhalt

Zusammenfassung.....	1
1 TachyGIS-Geodatenstruktur .....	3
1.1 GrabungsObjekt.....	3
1.2 Zusätzliche Attribute .....	4
1.3 Umsetzung im Geopackage .....	4
2 TachyGIS-Vokabular.....	6
2.1 Vokabular-Struktur Variante A (GDKE-RP-Version mit Konkordanztafel) .....	6
2.2 Vokabular-Struktur Variante B (LfA-SN-Version, Hierarchie mit sortierbarem Code) .....	7
2.2.1 Standard-Vokabulartabelle.....	7
2.2.2 Objekttyp .....	8
2.2.3 Datierung .....	8
2.2.4 Material .....	8
2.3 Hinweise zu den Vokabular-Inhalten.....	9
2.4 Nutzung der Vokabulartabellen im QGIS .....	9
2.4.1 QGIS-Attributformular für die Erfassungslayer .....	9
2.4.2 Styles zur Symbolisierung der Darstellungslayer.....	11
2.4.3 QGIS Layer-Verknüpfungen .....	11

~

~ **Versionshistorie:**

- ~ Version 1.x.x: Als Version 1.x.x wird der Arbeitsstand im November 2022 angesehen, der über GitHub ([https://github.com/Landesamt-fuer-Archaeologie-Sachsen/GIS-Projekt\\_UTM33](https://github.com/Landesamt-fuer-Archaeologie-Sachsen/GIS-Projekt_UTM33)) angeboten wurde, jedoch nicht ausführlich dokumentiert ist.
- ~ Version 2.x.x ist eine Überarbeitung, die ausführlich dokumentiert und in der Entwicklergruppe abgestimmt wird.
- ~ Version 2.0.x sind die Entwürfe des LfA Sachsen
- ~ Überarbeitung nach der Beratung vom 12.12.2022 mit RP
- ~ Version 2.0.08 vom 13.12.2022 → ohne Einsprüche aus der Entwicklergruppe
- ~ Entwurf Version 2.0.09: neue Objekt-Nr, Vokabular erweitert, Vokabular Datierung und Material ergänzt
- ~ Version 2.0.10: die Vokabulare sind noch unklar und werden hier ausgegliedert
- ~ Version 2.0.11: Entscheidung für „Nummern“ als Text (beibehalten)
- ~ Version 2.0.11 (Vokabular): Entscheidung für uncodiertes Vokabular
- ~ Version 2.0.12: Basis und Vokabular wieder zusammengeführt sowie Straffung und Konzentration auf GPKG, Überlegungen → Anhang
- ~ Version 2.0.13: Korrekturen entsprechend der aktuellen GPKG-Vorlage
- ~ Version 2.0.14:
- ~ Version 2.0.15: Layout erneuert und weitere Überarbeitung bzgl. der Trennung vom Vokabular
- ~ Version 2.0.16: die bisher separate **Vokabular-Dokumentation** v2.0.15 ist **hier wieder integriert**
- ~ Version 2.0.17: die Sortierung wurde wieder ins Vokabular aufgenommen, da bei QGIS geplant nutzbar

~

~ **2do:**

- ~ geplant: Version 2.1.x wird die in der Entwicklergruppe abgestimmte Version sein]

## 1 TachyGIS-Geodatenstruktur

Zum besseren Verständnis werden in dieser Dokumentation als Attributnamen vorzugsweise sog. **Konzept-Aliase** verwendet, die von der tatsächlichen Realisierung der Attributnamen z.B. im Shapefile leicht abweichen (um dort der Syntax zu genügen).

### 1.1 GrabungsObjekt

Für ein GrabungsObjekt wird eine einfache tabellarische Struktur mit folgenden Attributen benutzt (Tab. 1).

<b>GrabungsObjekt</b>		
<b>Konzept-Alias</b> (Attribut)	<b>Beschreibung</b>	<b>Anmerkung</b>
<b>*Objekt_UUID</b>	Primärschlüssel, im Gesamtbestand (global) eindeutige ID, am besten eine UUID, wird ggf. von ZusatzAttributen referenziert	
<b>Geometrie (Shape)</b>	Geometrie als WKB (WKT) bzw. Esri-Shape	
<b>Erfassungsdatum</b>	Datum (DateTime) der primären Messung/Erfassung	
<b>Modifikationsdatum</b>	Datum (DateTime) der letzten Änderung	neu
<b>Geo_Quelle</b>	Quelle der Geodaten (bei Korrekturen entsprechend der letzten Änderung): primär (on-site mit Tachymeter erfasst) / sekundär (off-site manuell korrigiert/ ergänzt, ...)	neu (Profile)
<b>Aktivitätskode</b>	eindeutiger Code für die Aktivität (Grabung, Prospektion, ...)	
<b>Objekttyp</b>	Schnitt, Planum, Befund, Fund, Proben, Profil, Punkt; Geologie, ...	
<b>Objektart</b>	feinere Kategorisierung, abh. vom Objekttyp	
<b>Objektspezifikation</b>	weitere feinere Kategorisierung, abh. von Objektart	neu
<b>Schnitt_Nr</b>	archäologiespezifische lesbare Objektnummer, die auch zur Beschriftung verwendet wird (wobei die Nummer großzügig als kurzer String umgesetzt wird, so dass ggf. auch „3a“ oder „3.1.2“ möglich ist, Grabungsdoku.-Richtlinie beachten!)	
<b>Planum_Nr</b>	archäologiespezifische lesbare Objektnummer ...	
<b>Befund_Nr</b>	archäologiespezifische lesbare Objektnummer ...	
<b>gehört_zu_Befund</b>	zur Kennzeichnung von Zusammenhängen und Hierarchien der Befunde	neu
<b>Fund_Nr</b>	archäologiespezifische lesbare Objektnummer ...	
<b>Probe_Nr</b>	archäologiespezifische lesbare Objektnummer ...	
<b>Profil_Nr</b>	archäologiespezifische lesbare Objektnummer ...	
<b>Punkt_Nr</b>	archäologiespezifische lesbare Objektnummer ...	
<b>Objekt_Nr</b>	allgemeine Objektnummer, die sich auf die Nummerierung des eingetragenen Objekttyps bezieht (für alle Fälle, in denen die o.a. archäologiespezifischen Nummern nicht passen, z.B. für Blattsnitte)	neu
<b>Datierung</b>	archäologischen Datierung in Textform (eher für kartografische Zwecke, eine präzisere Datierung ist in der Grabungsdokumentation zu erwarten)	
<b>Datierung2</b>	Raum für eine feinere Kategorisierung der Datierung (ggf. aus hierarchischem Thesaurus gebildet)	bei Bedarf
<b>Material</b>	archäologische Materialangabe in Textform (eher für kartografische Zwecke ...)	
<b>Material2</b>	Raum für eine feinere Kategorisierung des Materials (ggf. aus hierarchischem Thesaurus gebildet) , falls nötig	bei Bedarf
<b>Bemerkung</b>	Raum für Bemerkungen (keine Beschriftung)	

Tab. 1: Standard-Attribute für ein GrabungsObjekt.

## 1.2 Zusätzliche Attribute

Standardmäßig sieht die TachyGIS-Datenstruktur nur die o.a. Attribute vor. Optional lassen sich jedoch über die Objekt-UUID noch zusätzliche Attribute oder auch eine Grabungsdatenbank einbinden.

<b>ZusatzAttribut</b>		
<b>Konzept-Alias</b> <small>(Attribut)</small>	<b>Beschreibung</b>	<b>Anmerkung</b>
*ZA_UUID	Primärschlüssel, eindeutige ID für das Zusatzattribut, am besten als UUID	
Objekt_UUID	Verweis auf das GrabungsObjekt, zu dem das Attribut gehört	
Attributtyp	Typ (oder Art) des Attributes, die durch den folgenden Wert zu beschreibende Eigenschaft	
Attributwert	Wert für die zuvor angegebene Eigenschaft	ggf. für Zahlen und Texte separat

Tab. 2: Optionale ZusatzAttribute.

## 1.3 Umsetzung im Geopackage

Die Datensätze/Layer für Grabungsobjekte liegen in einem Geopackage (.gpkg-Datei). Enthalten sind mindestens:

- E\_Point (Grabungsobjekte als Punkte)
- E\_Line (Grabungsobjekte als Linien)
- E\_Polygon (Grabungsobjekte als Polygone)

Punkte, Linien und Flächen (Polygone) werden in drei separaten Datensätzen/Layern abgelegt (die sich auf „Simple Features“ beziehen).

In der konkreten Umsetzung (Tachy2GIS\_arch) werden Attributnamen verwendet, die ggf. leicht von den hier verwendeten Namen abweichen, eine Referenz dazu findet sich in *Tab. 3*. Dort sind auch die Änderungen zur bisherigen Datenstruktur (bis November 2022) verzeichnet (letzte Spalte und gelbe Markierung). Für die Attributnamen gilt:

- Kleinschreibung,
- ohne Umlaute,
- ohne Ziffern,
- Unterstrich (keine Bindestriche o. Leerzeichen).

In der vorgegebenen QGIS-Datenstruktur werden diese Geodaten durch die drei Layer E\_Point, E\_Line und E\_Polygon referenziert. Die Benennung der GPKG-Datensätze soll dem folgen. Die Attributstruktur der drei Layer ist identisch und wird nachfolgend nur einmal beschrieben.

<b>Konzept-Alias</b> <small>(Attribut)</small>	<b>Geopackage Attribut</b>	<b>Geopackage Alias</b>	<b>Geopackage Typ</b>	<b>SQLS-Typ (LfA-SN)</b>	<b>Änderung bzgl. Version 1</b>
FID	fid	Lokal_ID	integer	int	← id
*Objekt_UUID	obj_uuid	Objekt_UUID	string 50	nvarchar(50)	← uuid
Geometrie (Shape)	shape			geometry	
Erfassungsdatum	erf_datum	Erfassung	date/time	datetime	← messdatum
Modifikationsdatum	mod_datum	Modifikation	date/time	datetime	neu
Geo_Quelle	geo_quelle	Geo_Quelle	string 20	nvarchar(20)	neu
Aktivitätskode	aktcode	Aktivität	string 20	nvarchar(20)	
Objekttyp	obj_typ	Objekttyp	string	nvarchar(200)	← type
Objektart	obj_art	Objektart	string	nvarchar(200)	
Objektspezifikation	obj_spez	Objektspezifikation	string	nvarchar(200)	neu
Schnitt_Nr	schnitt_nr	Schnitt_Nr	string 50	nvarchar(50)	

Planum_Nr	planum_nr	Planum_Nr	string 50	nvarchar(50)	← planum
Befund_Nr	bef_nr	Befund_Nr	string 50	nvarchar(50)	
gehört_zu_Befund	gehört_zu	gehört_zu_Befund	string 50	nvarchar(50)	neu
Fund_Nr	fund_nr	Fund_Nr	string 50	nvarchar(50)	
Probe_Nr	probe_nr	Probe_Nr	string 50	nvarchar(50)	z.T. neu
Profil_Nr	prof_nr	Profil_Nr	string 50	nvarchar(50)	
Punkt_Nr	pt_nr	Punkt_Nr	string 50	nvarchar(50)	← ptnr
Objekt_Nr	obj_nr	Objekt_Nr	string 50	nvarchar(50)	neu
Datierung	zeit	Datierung	string	nvarchar(200)	
Datierung2	zeit_zwei	Datierung2	string	nvarchar(200)	neu
Material	material	Material	string	nvarchar(200)	
Material2	material_zwei	Material2	string	nvarchar(200)	neu
Bemerkung	bemerkung	Bemerkung	string	nvarchar(200)	

Tab. 3: Konkordanz der in den Geopackages verwendeten Attributnamen und Aliase mit den hier verwendeten Namen.

Der programmtechnische Zugriff soll sich auf das jeweilige Attribut (den Attributnamen) beziehen. Der Alias soll dadurch offen bleiben für institutionelle Anpassungen bezogen auf die Nutzeroberfläche.

(Anmerkung: Aus programmtechnischer Sicht könnte es zukünftig ggf. sinnvoll sein, die Nummern-Attribute der Grabungsobjekte nicht mehr als Text, sondern als Zahl umzusetzen.)

## 2 TachyGIS-Vokabular

Die TachyGIS-Version 2 stellt von Shapefiles auf Geopackage (GPKG) um, dabei werden auch die Vokabulare im GPKG verwaltet.

Die Vokabulare dienen der Erleichterung der Datenerfassung und enthalten standardisierte Begriffe, die bei der Erfassung von Attributen der Vermessungsdaten verwendet werden.

Es werden vorerst zwei verschiedene Vokabular-Strukturen angeboten:

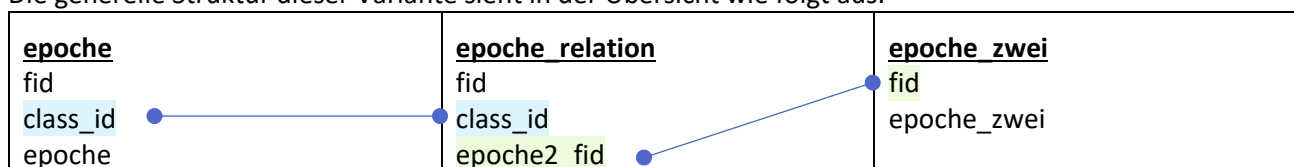
- Variante A (GDKE-RP-Variante, stark an GDKE-Daten orientiert) und
- Variante B (LfA-SN-Vorschlag mit einfacherer Struktur).

Diese beiden Vokabular-Varianten sind in den entsprechenden QGIS-Projektvorlagen funktionsfähig eingebunden (wozu auch entsprechende Einstellungen in den QGIS-Layern erforderlich sind). Die grundsätzliche Geodatenstruktur außerhalb der Vokabulare selbst und die Funktionalität der TachyGIS-Plugins unterscheiden sich jedoch nicht.

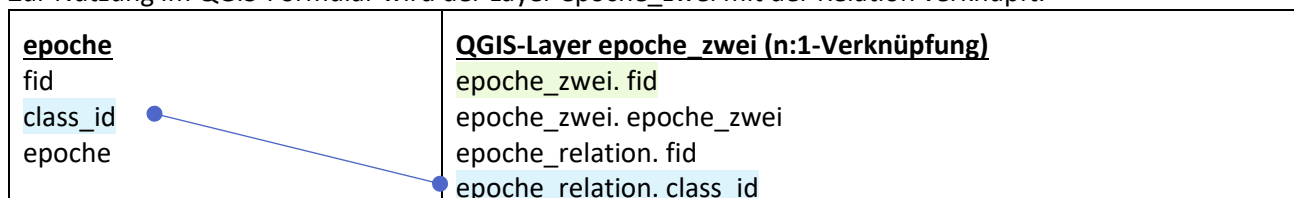
Die Vokabulare liegen gemeinsam mit den Grabungsobjekten im Geopackage (.gpkg-Datei).

### 2.1 Vokabular-Struktur Variante A (GDKE-RP-Version mit Konkordanztabelle)

Die generelle Struktur dieser Variante sieht in der Übersicht wie folgt aus.



Zur Nutzung im QGIS-Formular wird der Layer epoche\_zwei mit der Relation verknüpft.



Die folgenden Tabellen zeigen ein Vokabular-Beispiel (oben die originalen Tabellen, unten deren Nutzung im QGIS (*kursiv die Nutzung im QGIS-Formular*)).

epoche		
fid	class_id	epoche
1	11	Bronzezeit
2	12	Eisenzeit

epoche relation...		
fid	class_id	epoche2_fid
1	11	1
2	11	2
3	12	3
4	12	4

epoche_zwei	
fid	epoche_zwei
1	frühe Bronzezeit
2	späte Bronzezeit
3	frühe Eisenzeit
4	späte Eisenzeit



Nutzung im QGIS-Formular:

epoche		
fid	class_id	epoche
1	11	Bronzezeit
2	12	Eisenzeit
	<i>Schlüssel</i>	<i>Wert</i>



QGIS-Layer epoche_zwei (n:1-Verknüpfung)			
fid	epoche_zwei	fid	class_id
1	frühe Bronzezeit	1	11
2	späte Bronzezeit	2	11
3	frühe Eisenzeit	3	12
4	späte Eisenzeit	4	12
	<i>Schlüssel</i>	<i>Wert</i>	<i>Bedingung</i>



Diese Struktur wiederholt sich sinngemäß für

- das Material (Tabellen material, material\_zwei und material\_relation...) und
- für die Objekttypen (Tabellen obj\_type\_s1, obj\_type\_s2, obj\_type\_s3, obj\_type\_relation\_s1\_s2 und obj\_type\_relation\_s2\_s3).

#### Diskussion:

Die Datenstruktur ist mit der Konkordanztafel (...\_relation\_...) formell auf eine m:n-Beziehung zwischen zwei Klassen (z.B. material und material\_zwei) ausgelegt. Jedoch wird in QGIS nur eine 1:1 (besser n:1) - Verknüpfung realisiert. Auch der Inhalt der Vokabulare ist hierarchisch, so dass damit keine Konflikte entstehen. Wenn man jedoch in die Konkordanztafel Werte übernimmt, die nicht mehr streng hierarchisch sind, hat die QGIS-Verknüpfung einen Konflikt und gibt nicht alle Konkordanzen wieder. Das GDKE-Vokabular ist jedoch streng hierarchisch aufgebaut, so dass dieses Problem dort nicht auftritt.

Da einerseits hierarchische Vokabulare üblich sind und andererseits die m:n Konkordanz im QGIS nicht umgesetzt wird, ist die Nutzung einer Konkordanztafel eigentlich überdimensioniert.

Die Vielzahl der verwendeten Schlüssel ist ein wenig unübersichtlich. Hinzu kommt, dass die GPKG-systemspezifische „fid“ als Autowert konfiguriert ist, deren Werte zumindest in QGIS mit Copy/Paste nicht definiert gesetzt werden können. Dies ist insofern tragisch, als die Konkordanzen definierte „fid“'s benötigen. Eine Ergänzung des bestehenden Vokabulars ist damit nur durch einzelne manuelle Einträge möglich.

In der GDKE RP ergeben sich die Vokabulartabellen direkt aus den dort genutzten Thesauri und begründen damit diese Datenstruktur. Für andere Nutzer mit eigenen (hierarchischen) Vokabularen ist der folgende Alternativvorschlag (LfA-Struktur) jedoch möglicherweise einfacher zu handhaben.

## 2.2 Vokabular-Struktur Variante B (LfA-SN-Version, Hierarchie mit sortierbarem Code)

Der Strukturvorschlag beruht auf Standard-Vokabulartabellen, die sich über einen „kategorie“-Eintrag definiert hierarchisch verknüpfen lassen. (Es ist keine Konkordanztafel erforderlich.)

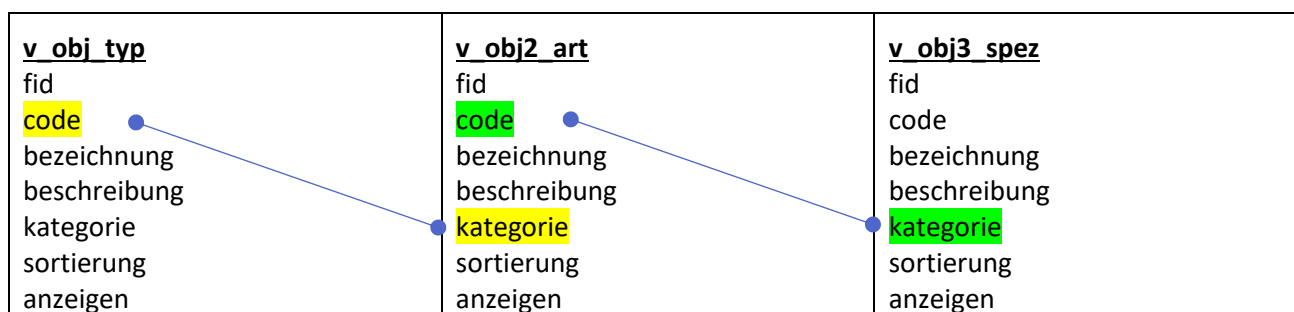
### 2.2.1 Standard-Vokabulartabelle

Nach dem folgenden Muster sind alle Tabellen aufgebaut, sie unterscheiden sich nur im Tabellennamen. (Das Attribut „Sortierung“ wurde im Vorgriff auf neu geplante QGIS-Versionen wieder aufgenommen.)

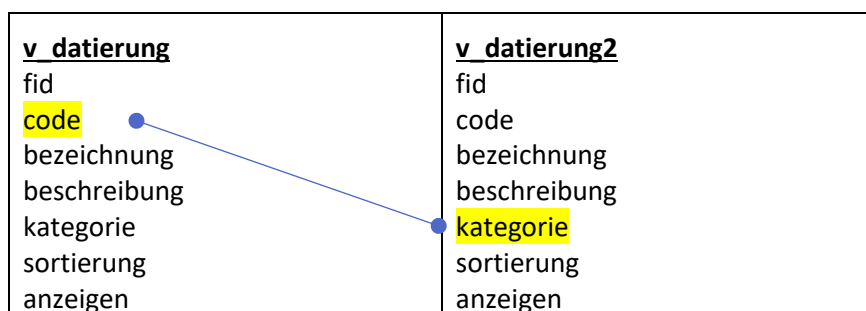
<b>v_vokabular (Muster mit eindeutigen Code)</b>			
<b>Attribut</b>	<b>Typ</b>	<b>Beschreibung</b>	<b>Anmerkung</b>
fid	Auto	(interne QGIS-ID, Auto-Wert, hier ohne Bedeutung)	
*code	text 100	<b>eindeutiger, sortierbarer</b> Code für den Begriff	UNIQUE
bezeichnung	text 254	eindeutige Bezeichnung für den Begriff	
beschreibung	text 254	ausführlichere Beschreibung des Begriffs, falls vorhanden und erforderlich, kann als Tool-Tipp im Formular angezeigt werden	
kategorie	text 100	übergeordnete Kategorie (Hierarchie, gehört zu ...), kann für kontextbezogene Ausfüllhilfe benutzt werden	→ Codes benutzen
sortierung	text 10	zur Sortierung des angezeigten Vokabulars (alphanum.)	
anzeige	bool	zur Einschränkung des angezeigten Vokabulars (z.B. im Pull-down-Feld) false/0 = nicht anzeigen, true/1 = anzeigen	BOOLEAN

Für Objekttyp, Datierung und Material können damit hierarchische Bezüge wie folgt aufgebaut werden.

### 2.2.2 Objekttyp



### 2.2.3 Datierung



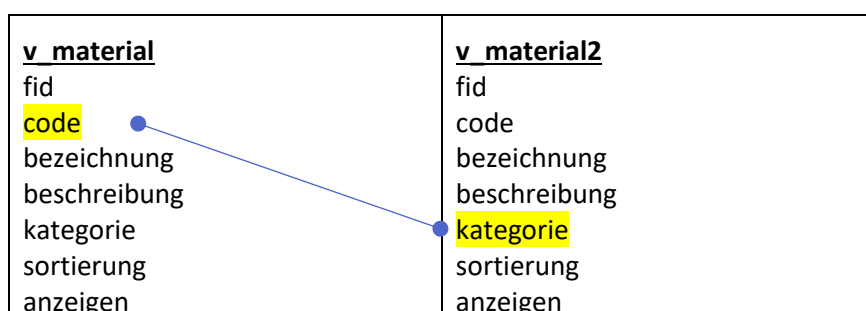
Beispiel zur Datierung:

v_datierung					
fid	code	bezeichnung	kategorie	sortierung	anzeigen
1	4000	Bronzezeit		4000	ja
2	5000	Eisenzeit		5000	ja
	<i>Schlüssel</i>	<i>Wert</i>			<i>Bedingung</i>

v_datierung2					
fid2	code2	bezeichnung2	kategorie2	sortierung2	anzeigen2
1	4100	frühe Bronzezeit	4000	4100	ja
2	4200	späte Bronzezeit	4000	4200	ja
3	5100	frühe Eisenzeit	5000	5100	ja
4	5200	späte Eisenzeit	5000	5200	ja
5	9999	Test		9999	nein
	<i>Schlüssel</i>	<i>Wert</i>	<i>Bedingung</i>	<i>Bedingung</i>	<i>Bedingung</i>

Die kursiven Passagen kennzeichnen die Nutzung im QGIS-Formular.

### 2.2.4 Material



## 2.3 Hinweise zu den Vokabular-Inhalten

Es sollte auf folgendes geachtet werden:

- Leerzeichen entfernen (trim), insbes. bei den Codes!
- Die Inhalte der TopLevel-Tabelle sollen im 2. Level mit auftauchen (gleicher Code)
  - zur Erkennung/Unterscheidung beginnen diese Begriffe in der Bezeichnung mit „~“,
  - im Formular wird im 2. Level der Wert aus dem 1. Level als Defaultwert gesetzt,
  - damit ist gesichert, dass im 2. Level immer ein Wert steht, der für Filter o.ä. ausgewertet werden kann (im 3. Level obj\_spez ist dies momentan noch nicht umgesetzt, weil mühsam und ggf. nicht erforderlich).
- Hierarchie beachten:
  - entweder streng
    - s. Filter bei 2.4.1
  - oder polyhierarchisch
    - Mehrere übergeordnete Begriffe in der „kategorie“ sind durch Trennzeichen (davor/danach) getrennt, z.B. „|B|BO|BU|“ (für Befund, Befundoberkante, Befundunterkante treffen dieselben Unterbegriffe zu).
    - s. Filter bei 2.4.1
- Für den Objekttyp (obj\_typ) kann als Kategorie der Geometrietyp polyhierarchisch angegeben werden, damit z.B. im Polygonlayer keine Höhenpunkte als Wert angeboten werden, also z.B.
  - für einen Höhenpunkt: „|Point|“,
  - für ein Profil „|Line|“,
  - für einen Befund „|Point|Line|Polygon|“.
- Die Sortierung ist bei aktuellen QGIS-Versionen noch nicht nutzbar, jedoch geplant. Es wird von einer alphanumerischen Sortierung (Textfeld) ausgegangen, die insbesondere bei den Datierungen hilfreich sein dürfte, ohne dass man den Schlüssel sortierbar gestalten muss. Dies sollte bei der Gestaltung des Vokabulars gleich mit berücksichtigt werden.

## 2.4 Nutzung der Vokabulartabellen im QGIS

**Um die Austauschbarkeit der beiden Versionen zu gewährleisten, sollten die Vokabulartabellen nur im QGIS-Projekt benutzt werden!**

Eine direkte Nutzung im Code des TachyGIS-Plugins sollte ausgeschlossen werden. Sofern (z.B. für die Profilerstellung) Werte aus den Vokabulartabellen relevant sind, müssen diese konfigurierbar sein!

### 2.4.1 QGIS-Attributformular für die Erfassungslayer

Das zur Erfassung gewünschte Verhalten ist wie folgt:

- Zur Erfassung der entsprechenden Attribute soll das Vokabular aus einer Nachschlageliste ausgewählt werden können.
- Es sollen nur Werte mit dem Attribut „anzeige“ = true angezeigt werden.
- Bei den untergeordneten Vokabularen sollen nur Werte angezeigt werden, die in ihrer „kategorie“ dem zuvor erfassten übergeordneten Wert entsprechen. Dabei bestehen folgende Hierarchien (übergeordnet links):
  - (Geometrietyp<sup>1</sup>) ← obj\_typ ← obj\_art ← obj\_spez,
  - zeit ← zeit\_zwei,
  - material ← material\_zwei.

---

<sup>1</sup> Polygon, Line, Point

Die **Erfassungslayer** sind E\_Point, E\_Line und E\_Polygon. Hier werden die Vokabulare für die entsprechenden Felder benutzt: obj\_typ, obj\_art, obj\_spez, zeit, zeit\_zwei, material und material\_zwei. Die Einstellung in den Layereigenschaften sind wie folgt:

Für die **Sortierung** der Nachschlagewerte gibt es nur zwei Varianten über die Einstellung „Nach Werten sortieren“:

- ohne Haken: Sortierung nach Schlüsselfeld (→ sortierbaren Code benutzen)
- mit Haken: Sortierung nach Wertefeld (→ für Zeitstellungen ungeeignet)

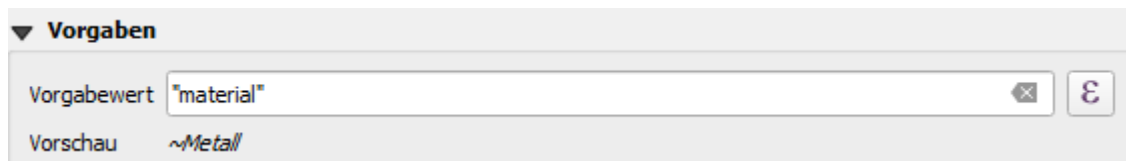
In allen Attributen wird der **Filter** auf „anzeige“ gesetzt. In den untergeordneten Attributen wird zusätzlich nach dem übergeordneten Wert gefiltert. Mit einem kleinen Trick kann man hier auch Polyhierarchien abbilden, indem man nach einem Teilstring zwischen zwei bestimmten Trennzeichen sucht.

Standardfilter für die Attribute als Beispiel:

<b>obj_typ</b>	"anzeige" AND strpos("kategorie", ' Polygon ') > 0 <i>[bzw. Line oder Point]</i>
<b>obj_art*</b>	"anzeige" AND ( "kategorie"=current_value('obj_typ') OR strpos("kategorie", concat( ' ', current_value( 'obj_typ' ),' ' ) ) > 0)
<b>obj_spez*</b>	"anzeige" AND ( "kategorie"=current_value('obj_art') OR strpos("kategorie", concat( ' ', current_value( 'obj_art' ),' ' ) ) > 0)
<b>zeit</b>	"anzeige"
<b>zeit_zwei</b>	"anzeige" AND "kategorie" = current_value( 'zeit' )
<b>material</b>	"anzeige"
<b>material_zwei</b>	"anzeige" AND "kategorie"= current_value('material')
<b>*</b>	<i>Diese Abfragen unterstützen Polyhierarchie mit Kategorien wie z.B. ' Wert 1 Wert 2 '.</i>

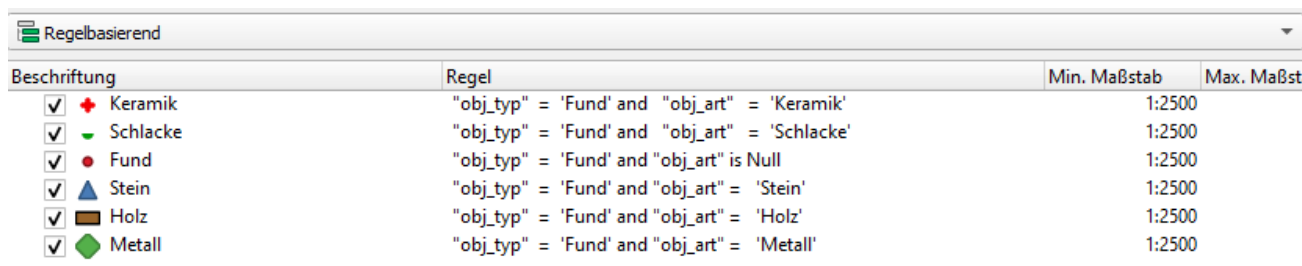
Um die Styles gut regelbasiert symbolisieren zu können ist es sinnvoll, übergeordnete Werte bei der Ersterfassung auch in die untergeordneten Attribute der zweiten Hierarchieebene zu spiegeln, um dann dort







einfach filtern zu können. Dazu setzt man in obj\_art, zeit\_zwei und material\_zwei den Vorgabewert auf den zuvor erfassten übergeordneten Wert der ersten Hierarchieebene, z.B. im Attribut „material\_zwei“ so



### 2.4.2 Styles zur Symbolisierung der Darstellungslayer

Die Darstellungslayer werden regelbasiert symbolisiert. Dabei werden Regeln benutzt, die auf Werte bestimmter Attribute ausgerichtet sind, z.B.



Beschriftung	Regel	Min. Maßstab	Max. Maßstab
<input checked="" type="checkbox"/>  Keramik	"obj_typ" = 'Fund' and "obj_art" = 'Keramik'		1:2500
<input checked="" type="checkbox"/>  Schlacke	"obj_typ" = 'Fund' and "obj_art" = 'Schlacke'		1:2500
<input checked="" type="checkbox"/>  Fund	"obj_typ" = 'Fund' and "obj_art" is Null		1:2500
<input checked="" type="checkbox"/>  Stein	"obj_typ" = 'Fund' and "obj_art" = 'Stein'		1:2500
<input checked="" type="checkbox"/>  Holz	"obj_typ" = 'Fund' and "obj_art" = 'Holz'		1:2500
<input checked="" type="checkbox"/>  Metall	"obj_typ" = 'Fund' and "obj_art" = 'Metall'		1:2500

Die konkrete Umsetzung kann entsprechend institutioneller oder individueller Vorgaben angepasst werden.

### 2.4.3 QGIS Layer-Verknüpfungen

In QGIS Layern können 1:1 (bzw. n:1) -Verknüpfungen erstellt werden. Aber Achtung: Falls sich andere Beziehungen in den Daten verstecken, wird nur ein Datensatz gezeigt!

Die Vokabular-Struktur Variante A (GDKE-RP) nutzt dies, um untergeordnete Vokabulare (z.B. epoche\_zwei) mit der entsprechenden Konkordanztabelle (epoche\_relation...) zu verknüpfen. In der Variante B (LfA-SN) ist dies nicht erforderlich.

Für komplexere Beziehungen (1:n oder m:n) muss man die QGIS-Dokumentation zu Rate ziehen.

# Anlage 1: Vergleich der Versionen 1 und 2 (Shapefile und Geopackage)

Der Umstieg auf Geopackage wurde zugleich mit der Konsolidierung der Datenstruktur durchgeführt, so dass eine Umsetzung mit Shapefile obsolet ist. Hier der Vollständigkeit halber noch mal die für Shapefiles in Frage kommende Konkordanz.

Vergleich von Version 1 (Shapefile) mit Version 2 (Geopackage)

	Version 1 (Shapefile)		Version 2 (Geopackage)	
Konzept-Alias <small>(Attribut)</small>	Attribut	Typ	Attribut	Typ
FID			fid	integer
*Objekt_UUID	uuid	string 50	obj_uuid	string 50
Shape (Geometrie)	shape	shape	???	geometry ???
Erfassungsdatum	messdatum	date/time	erf_datum	date/time
Modifikationsdatum	x		mod_datum	date/time
Geo_Quelle	x		geo_quelle	string 20
Aktivitätskode	aktcode	string 20	aktcode	string 20
Lokal_ID	id	integer	x	
Objekttyp	type	string 254	obj_typ	string
Objektart	obj_art	string 254	obj_art	string
Objektspezifikation	x		obj_spez	string
Schnitt_Nr	schnitt_nr	string 50	schnitt_nr	string 50
Planum_Nr	planum	string 50	planum_nr	string 50
Befund_Nr	bef_nr	string 50	bef_nr	string 50
gehört_zu_Befund	x		gehoert_zu	string 50
Fund_Nr	fund_nr	string 50	fund_nr	string 50
Probe_Nr	probe_nr	string 50	probe_nr	string 50
Profil_Nr	prof_nr	string 50	prof_nr	string 50
Punkt_Nr	ptnr	string 50	pt_nr	string 50
Objekt_Nr	x		obj_nr	string 50
Datierung	zeit	string 254	zeit	string
Datierung2	x		zeit_zwei	string
Material	material	string 254	material	string
Material2	x		material_zwei	string
Bemerkung	bemerkung	string 254	bemerkung	string

Tab. 4: Konkordanz der in den Shapefiles verwendeten Attributnamen und Aliase mit den hier verwendeten Namen.

## Anlage 2: Vorüberlegungen und Diskurs

---

Die Geodatenstruktur soll im QGIS-Plugin „Tachy2GIS\_arch“ zur Dokumentation von archäologischen Grabungen benutzt werden.

Die Datenstruktur konzentriert sich auf die in Grabungsplänen üblichen Zeichnungsobjekte, hier nachfolgend „GrabungsObjekt“ genannt. Weitere Details einer vollständigen Grabungsdokumentation werden hier bewusst außer Acht gelassen, um die bestehenden Grabungsdatenbanken nicht in Frage zu stellen. Solche Grabungsdatenbanken (oder Teile daraus) können mit den Zeichnungsobjekten über deren ID verknüpft werden.

Für die GrabungsObjekte wird eine einfache flache (tabellarische) Struktur aus folgenden Gründen favorisiert:

- die Daten dafür werden in Shapefiles tabellarisch erfasst (QGIS),
- die Auswertung erfolgt vorwiegend ebenfalls tabellarisch (Shapefile/GIS, Excel),
- keine Transformationen für den Im-/Export nötig,
- leichte Verständlichkeit.

Zum Vergleich wurden folgende Erfahrungen bzw. Planungen mit erörtert:

- In BW werden die Grabungspläne mit „survey2GIS“ tabellarisch erfasst und als Shapefile archiviert.
- In HH ist eine hierarchische Datenstruktur geplant: Fund → Befund → Planum/Schnitt → Grabung, wobei sich die den „Plan“ betreffenden attributiven Anteile (Geometrie, Label, Style) von Stufe zu Stufe nicht unterscheiden, diese also genauso gut aus der hierarchischen Struktur entfernt und zusammengefasst werden könnten (wie der hier vorgestellte Vorschlag).

Abweichungen von einer flachen 1-Tabellen-Struktur wurden wie folgt erwogen:

- Die Geometrie wird für verschiedene Objekttypen erfasst (Funde, Befunde, ...), was unterschiedliche Attribute vermuten lässt. Da wir uns jedoch bewusst auf die Ebene des „Plans“ (der Zeichnung) reduzieren, kommen die unterschiedlichen archäologischen Attribute hier nicht zum Tragen. Die Objekttypen werden lediglich als verbindliches Vokabular für das entsprechende Attribut angesehen.
- Objektarten dito. Hier soll jedoch noch die hierarchische Unterordnung unter bestimmte Objekttypen beachtet werden.
- Zeichenvorschriften bestehen eigentlich lt. Richtlinie für die Objekttypen. Dies kann indirekt über spezifische Abfragen und Layouts im GIS gewährleistet werden, ohne dass in der Datenstruktur Angaben darüber erhoben werden. Sofern jedoch individuelle Layouts aufgehoben werden müssen (noch zu klären), kommt eine der folgenden Varianten ins Spiel (die alle nicht problemlos sind):
  - für jede Geometrie separat in der Objekttablelle speichern (flache Struktur, Umsetzung im GIS fraglich)
  - in einer separaten Tabelle speichern und mit dem Objekt verknüpfen bzw. auch die Standardlayouts eintragen und mit den Objekttypen verknüpfen (spart ggf. Platz, Umsetzung im GIS fraglich, Konsistenz fraglich)
  - es werden zusätzlich Layer erfasst und per Link mit Zeichenvorschriften verknüpft (Abweichungen von der Richtlinie vorprogrammiert)
- Für einen Standard-Plan nach Richtlinie ergibt sich eigentlich keine Notwendigkeit für eine flexible Attributierung. Sollte es dennoch notwendig sein, benutzerdefinierte Attribute flexibel erfassen zu können, wird eine separate Tabelle mit Link zu den Grabungsobjekten erforderlich. Dazu gibt es jedoch bei der Erfassung im GIS bisher keine Möglichkeit, so dass spezielle Entwicklungen notwendig wären. Eine Anregung dazu wird weiter unten gegeben (vgl. Abschnitt 1.2).
- lt. Richtlinie sind die Attribute zum Grabungsplan eigentlich vorgegeben, so dass diese Vorgabe in einer flachen Tabelle erfüllt werden kann
- Hierarchien bzw. die Beziehungen zwischen verschiedenen Objekttypen spielen, wie oben diskutiert, für den Plan an sich erst mal keine Rolle. Aus pragmatischen Gründen wurde jedoch der Prototyp so

gestaltet, dass für die hauptsächlichen Objekttypen entsprechende auf der Grabung vergebene Nummern parallel angegeben werden können (damit entsteht eine gewisse Struktur). Hier ist fraglich, ob bei der Erfassung diesbezügliche Constraints (z.B. strenge Hierarchie?) gewährleistet werden können. Es ist jedoch auch fraglich, ob solche Constraints hier überhaupt angebracht sind. Man kann sich auch auf den Standpunkt stellen, dass einfach alle Beobachtungen eingegeben werden. Auf eine bestimmte Struktur ist dann natürlich kein Verlass. Es ist erst einmal naheliegend, diesen Pragmatismus zu übernehmen.

- Die Konsistenz der Attributwerte hinsichtlich (strukturiertem) Vokabular muss bei der Eingabe gewährleistet werden, weil Korrekturentscheidungen nur vor Ort getroffen werden können. Insofern müssen inkonsistente Werte bei der Übernahme in die GDB weitgehend toleriert werden.
- Bei Bedarf können optional zusätzliche Attribute sehr flexibel über eine weitere Tabelle „ZusatzAttribut“ integriert werden, die über die ID mit entsprechenden GrabungsObjekten verknüpft ist. Hier können zu je einem GrabungsObjekt mehrere weitere Attribute mit Typ und Wert angegeben werden.
- Bei Bedarf können optional Bereiche einer kompletten Grabungsdatenbank mit entsprechenden GrabungsObjekten verknüpft werden.

#### Sonstiges:

- Bisher wird das Vokabular direkt und ohne Codes verwendet (obwohl z.B. in SN für Datierung usw. Codes zur Verfügung stehen). Die Verwendung von Codes setzt jedoch einen anderen Auswahl- und Anzeigemechanismus voraus (Vokabular aussuchen, Code eintragen), der mit QGIS evaluiert werden müsste. Dazu wird vorerst keine generelle Lösung angestrebt
- Attribute zu Genauigkeit und Qualität wurden diskutiert, momentan jedoch wegen mangelnder Relevanz bei der Erfassung abgelehnt (die Erfassungsmethode bleibt im Wesentlichen gleich und wird in der Grabungsdoku. beschrieben). Ggf. kann die Spalte Geo\_Quelle darüber indirekt Auskunft geben, indem man entsprechende Bezeichnungen ergänzt (z.B. „primär unscharf“ für unscharfe Objektumrisse, die aber mit Tachymeter primär gemessen wurden). Wobei das Angebot solcher Attribute ggf. weitere Unklarheiten nach sich zieht und daher als Option gern ausgeschlossen wird (→ man soll sich entscheiden).
- Die Aufnahme einiger weiterer Attribute in die Datenstruktur wurde diskutiert (z.B. Datierung\_von, Datierung\_bis, Kulturgruppe, Befundbeschreibung), dies wurde aber mit dem anfangs angeführten Argument abgelehnt, dass hier schwerpunktmäßig der Grabungsplan abgebildet wird und detailliertere Angaben der (ggf. separaten oder ergänzenden) Grabungsdokumentation vorbehalten sind.
- Die Umwidmung der „Nummern“ von Text in Zahlen wurde diskutiert. Dies wäre inhaltlich schlüssig. Es passt jedoch nicht in Situationen, wo „Nummern“ wie 1a oder 2.3 als Ausnahme zugelassen sind. Für die Zulassung gibt es archäologisch gesehen Pro und Contra. Im Zweifel sollte man die Ausnahme zulassen.
- **Es wird erst einmal davon ausgegangen, dass die Nummern als Text bestehen bleiben!**
- **Abhängig von der Funktionalität, die in RP gerade an GPKG angepasst wird, wird jedoch eine spätere Entscheidung für eine Umwidmung in Nummern nicht ausgeschlossen.**

## Anlage 3: alte CSV-Vokabulare und Diskurs zur Vokabular-Struktur

---

In der **TachyGIS-Version 1** waren die Vokabulare als CSV in folgender Struktur hinterlegt:

- Tabelle Objekttypen
  - field\_1 (Objekttyp Bezeichnung)
  - field\_2 (leer?)
  - field\_3 (leer?)
  - field\_4 (Anzeige?)
  - field\_5 (Bezug zu Geometriertyp?)
- Tabelle Objektarten
  - field\_1 (Objektart Bezeichnung)
  - field\_2 (Bezug zu Objekttyp)
  - field\_3 (Code, Kürzel)
  - field\_4 (Anzeige?)
- Tabelle Material
  - field\_1 (Material Bezeichnung)
  - field\_2 (Bezug zu Objekttyp)
  - field\_3 (Code, Kürzel)
  - field\_4 (Anzeige?)
- Tabelle Zeiten
  - field\_1 (Code)
  - field\_2 (Zeit Level 1, sehr grob)
  - field\_3 (Zeit Level 2, grob)
  - field\_4 (Zeit Level 3, fein)
  - field\_5 (Zeit Level 4, sehr fein)
  - field\_6 (Zeit Bezeichnung, alle Level)
  - field\_7 (Anzeige?)
- Tabelle geo\_arch
  - field\_1 (Code geo oder arch)
  - field\_2 (Bezeichnung Geologie oder Archäologie)

Die **TachyGIS-Version 2** stellt von Shapefiles auf Geopackage (GPKG) um, dabei sollen auch die Vokabulare im GPKG verwaltet werden.

Nachfolgend einige **Grundgedanken zu Vokabularen**.

Im Vokabular werden Begriffe gespeichert, je Datensatz ein Begriff. Dazu gehört unbedingt die Angabe einer Bezeichnung und je nach Nutzungszweck noch ein Code, eine Beschreibung und eine Kategorie als Verweis auf einen übergeordneten Begriff.

Einige Vokabulare bilden Hierarchien (bzw. Bezüge, Abhängigkeiten):

- v\_objektyp
  - v\_objektart
    - v\_objektspezifikation
- v\_material
  - v\_material2
- v\_datierung
  - v\_datierung2

Die wichtigsten **Probleme und Diskurse aus der Entwicklungsphase** im LfA SN sind nachfolgend aufgelistet.

- Vokabulare sind hierarchisch aufgebaut. Grundsätzlich kann man die Hierarchie
  - in einer gemeinsamen Tabelle abbilden (wie unsere Zeitstellungen),
    - das ist etwas übersichtlicher, falls man daran etwas ändern muss,
  - oder in mehreren separaten Tabellen abbilden,
    - das bietet bessere Möglichkeiten zu Abbildung von Hierarchien.
- Die bisherigen Vokabulare waren nicht streng hierarchisch aufgebaut. Durch Mehrfachnennungen, die in den Formularen entsprechend ausgewertet wurden, konnten Polyhierarchien gebildet werden.
  - Dies widerspricht in dieser Form zwar einem strengen Datenbankentwurf, weil keine Fremdschlüssel gebildet werden können, funktioniert aber in der Praxis fehlerfrei.
- Natürlich kann man Polyhierarchie auch mit m:n-Verknüpfungen (Konkordanztabelle) aufbauen, wie in der GDKE-Variante.
  - Dabei entsteht aber eine Struktur, die sich nicht mehr ganz einfach korrigieren lässt.
  - Die GDKE-Variante verwendet dazu einerseits QGIS-FIDs als Schlüssel, die jedoch als Autowert schlecht zu ändern sind (falls man das Vokabular ändern/ergänzen möchte). Die Übernahme von externen Vokabulardaten über QGIS (Zwischenablage importieren) führte nicht zu definierten FIDs sondern zu Autowerten, die nicht der Vorlage entsprachen und damit falsche Beziehungen bildeten.
  - Andererseits nutzt die GDKE-Variante QGIS-Layer-Verknüpfungen, die jedoch nicht m:n, sondern nur n:1 funktionieren und so bei echten polyhierarchischen Daten versagen. Eine Behebung des Problems ist nicht gelungen.
- Die QGIS-Formulare erwarten (zumindest, wenn man mit Wertbeziehungen = Vokabular-Nachschlage-Tabellen arbeitet) für die Filterung nach einem hierarchisch übergeordneten Feld den Wert der Schlüsselspalte (die Bezeichnung wird im Filter nicht akzeptiert).
  - Dies kann in einer gemeinsamen Tabelle nicht gewährleistet werden.
  - → Es werden separate Tabellen für jedes Attribut benutzt, auch bei hierarchischen Abhängigkeiten.
- Die Verwendung von **Wertebeziehungen mit Codes** (im QGIS-Erfassungsformular definiert) **führt** in QGIS an verschiedenen anderen Stellen (oder generell???) **dazu, dass die Codes abgespeichert werden, während die bezogenen Werte angezeigt werden.**
  - Beim Datenexport kann z.B. angegeben werden, dass nicht Codes sondern Werte exportiert werden.
  - Ob an anderen Stellen (z.B. bei Filtern) Codes oder Werte benutzt werden, muss beachtet werden!
  - Für WYSIWYG (oder KISS) würde man also besser auf die Verwendung von Codes verzichten oder zumindest sprechende Codes verwenden.
- Es bleibt aber noch das Problem der **Sortierung der Nachschlagewerte** für die Anzeige. Hier bietet QGIS nur 2 Möglichkeiten: nach dem Schlüsselfeld oder nach dem Wertefeld.
  - zumindest bei den Datierungen kann nicht nach den Werten sortiert werden → hier sind also sortierbare Schlüssel (Codes) erforderlich
- Da sich in der GDKE-Variante mit einfachen Mitteln kein Vokabular importieren ließ, entschied sich LfA SN letztendlich doch für eine separate Vokabularvariante mit folgenden Eigenschaften:
  - je eine Vokabulartabelle für jedes betroffene Attribut
  - alle Vokabulartabellen haben dieselbe einfache Struktur
  - es werden von der FID unabhängige, eindeutige und sortierbare Codes als Schlüssel benutzt
  - die Hierarchie wird n:1 über ein integriertes Feld „Kategorie“ als Verweis auf die übergeordnete Tabelle abgebildet
  - über spezifische Abfragen für die Vokabulare in den Erfassungsmasken kann eine einfache Polyhierarchie ganz gut nachgebildet werden, ohne eine separate Konkordanztabelle zu bemühen
- Es sollen daher beide Vokabular-Varianten in je einer Projektvorlage zur Nutzung angeboten werden.
- Beide Varianten sollen sich neutral zu den Plugins verhalten und damit problemlos austauschbar sein.