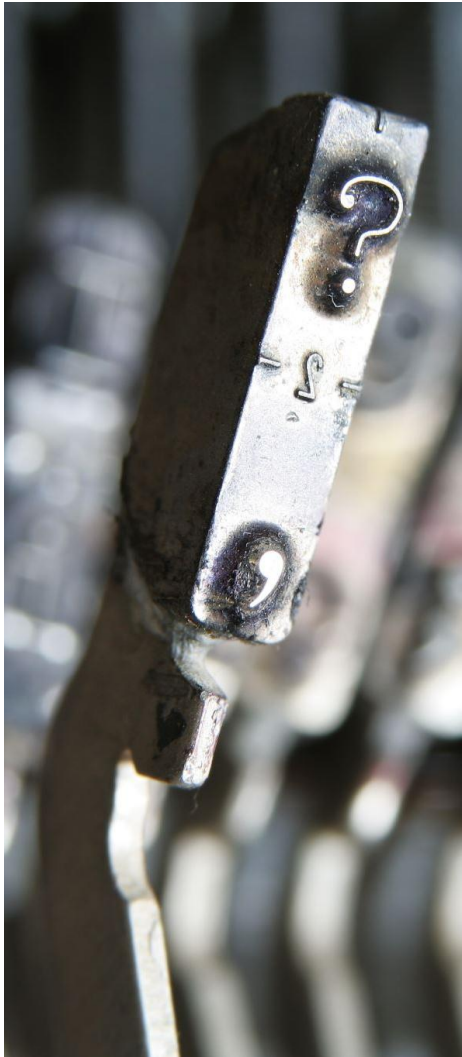


Drillhole Data Capture in QGIS

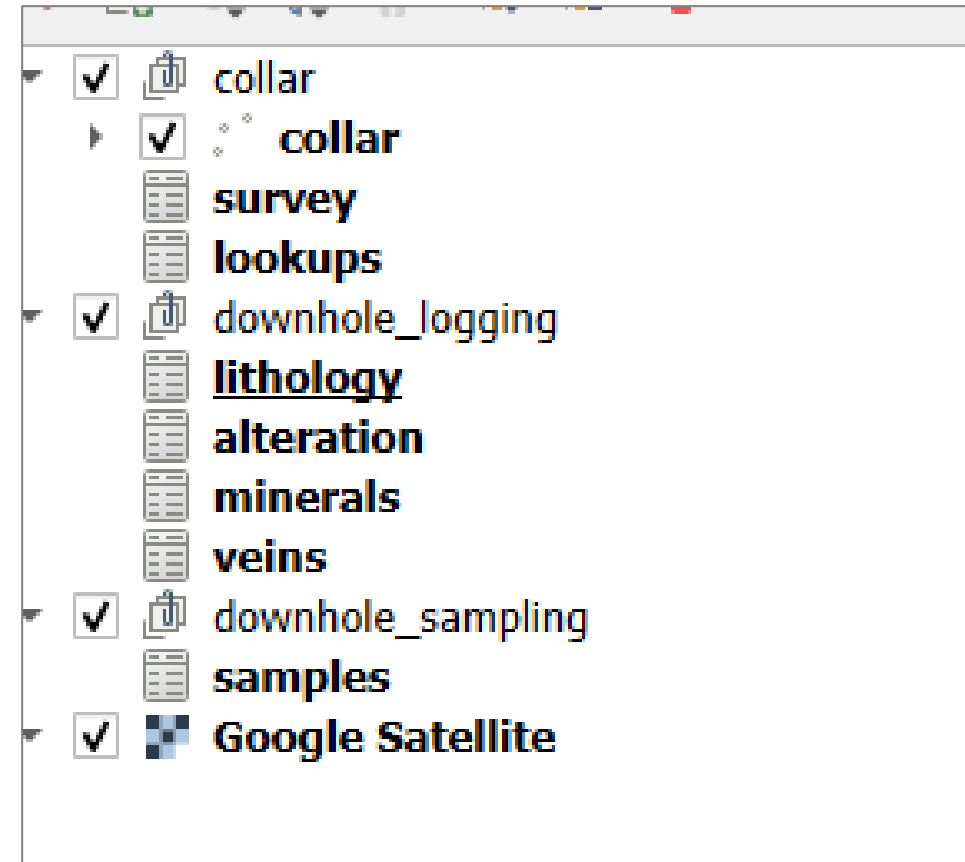
The Objective



- To provide an online/offline, flexible, free and open-source field logging solution that would replicate or exceed the features and functionality provided by current off the shelf solutions.
- The solution would have to be user friendly, customisable, and include extensive data quality checks and constraints to ensure clean and usable data.
- The solution must seamlessly integrate with existing geological databases for the purposes of import, export and library table management.
- Test if the solution is QField (and tablet) compatible.

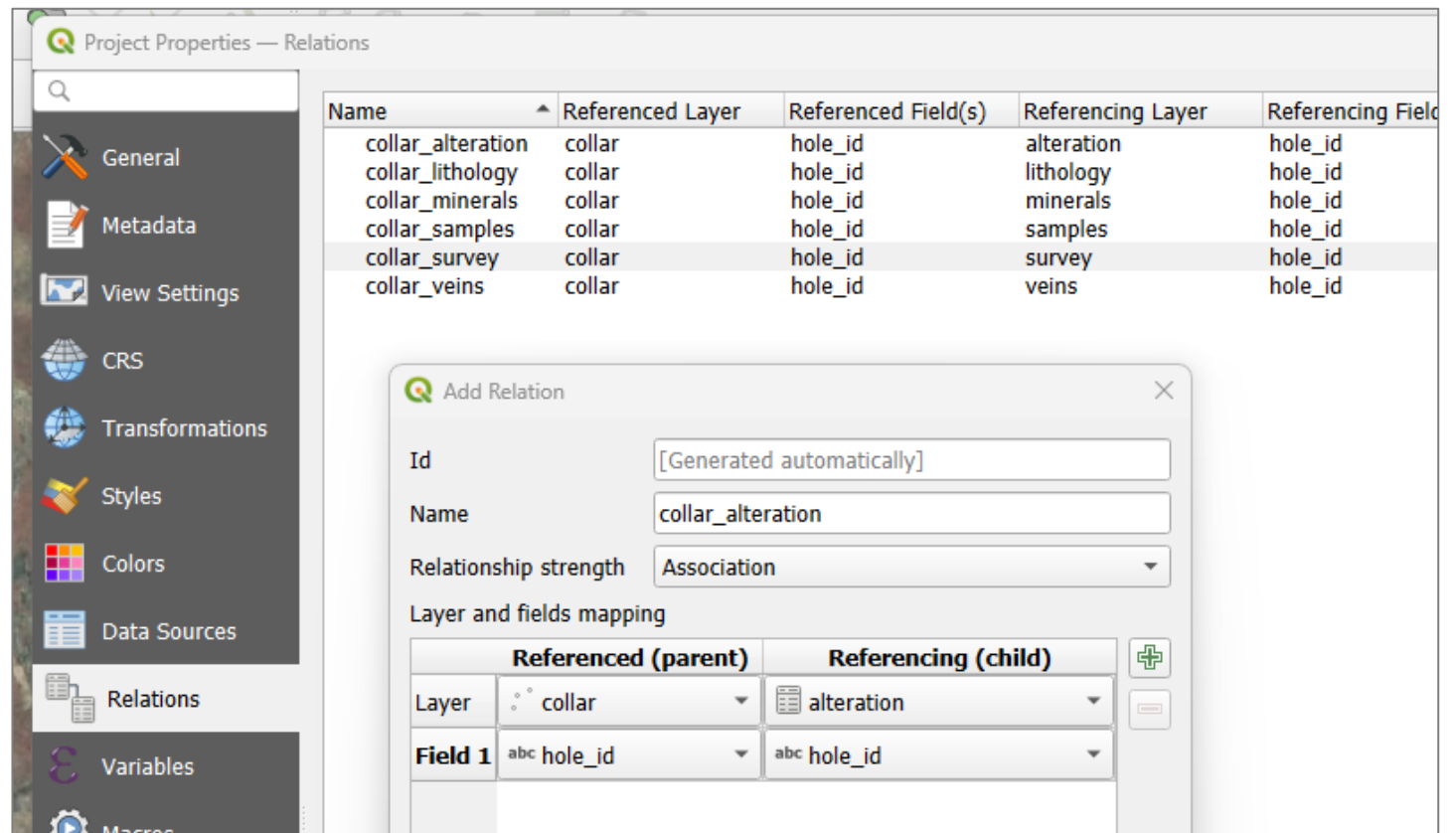
Create Geopackage and Layers

- Create a Geopackage.
- Create your layers (or drag and drop files) and save them to the geopackage.
- Collar table is spatial, rest non-spatial.
- Group tables to your liking.
- Save Project to Geopackage (pick CRS).
- Everything is now contained in a single file, easy to copy and share!
- Add any basemaps or additional layers you find useful.



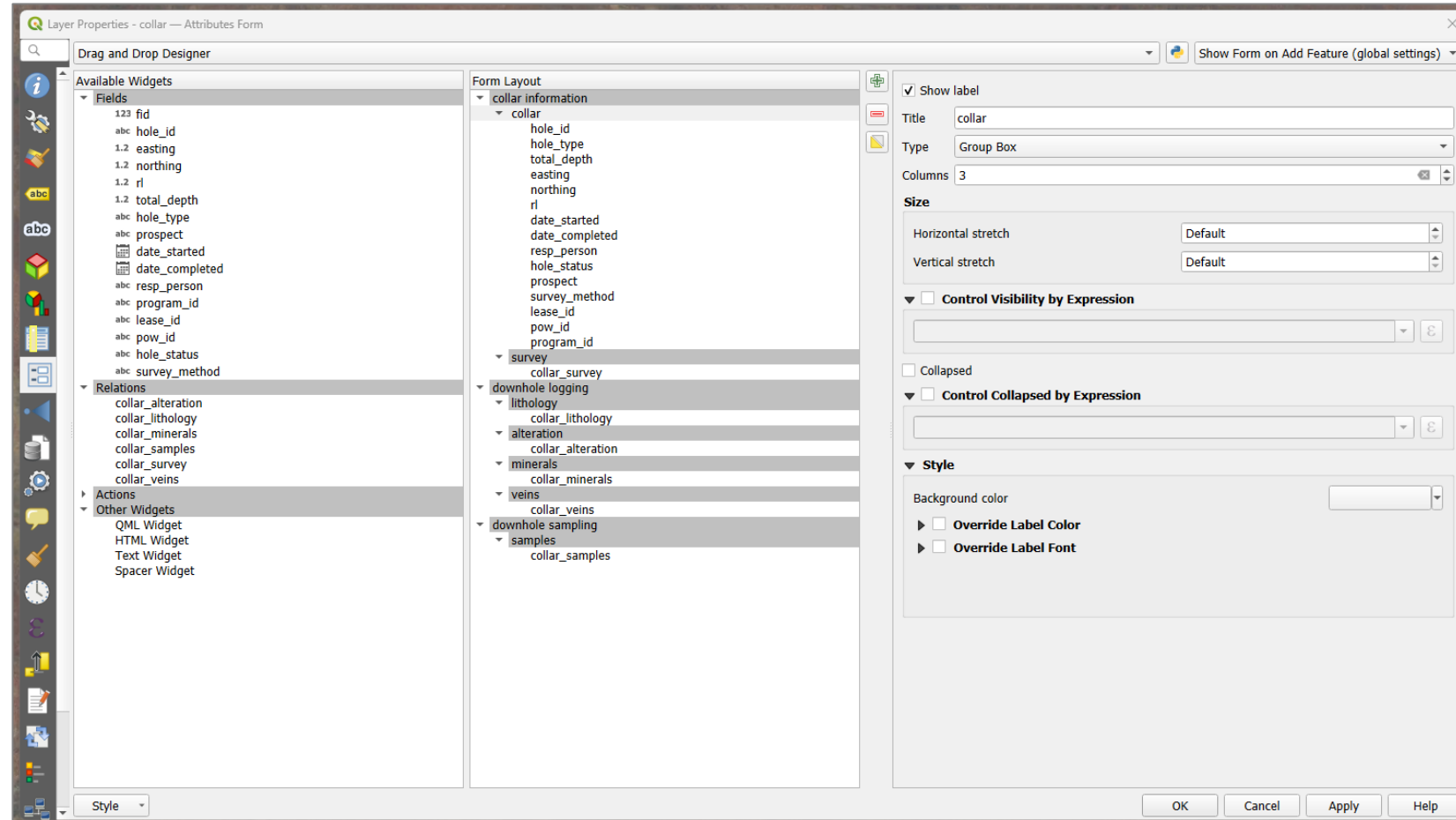
Create the Layer Relationships

- Create relationships between the tables within the Project.
- Define your parent child relationships (think foreign keys).
- Relationship strength: Associated (link) or Composition (cascade delete).
- Repeat for all downhole tables.



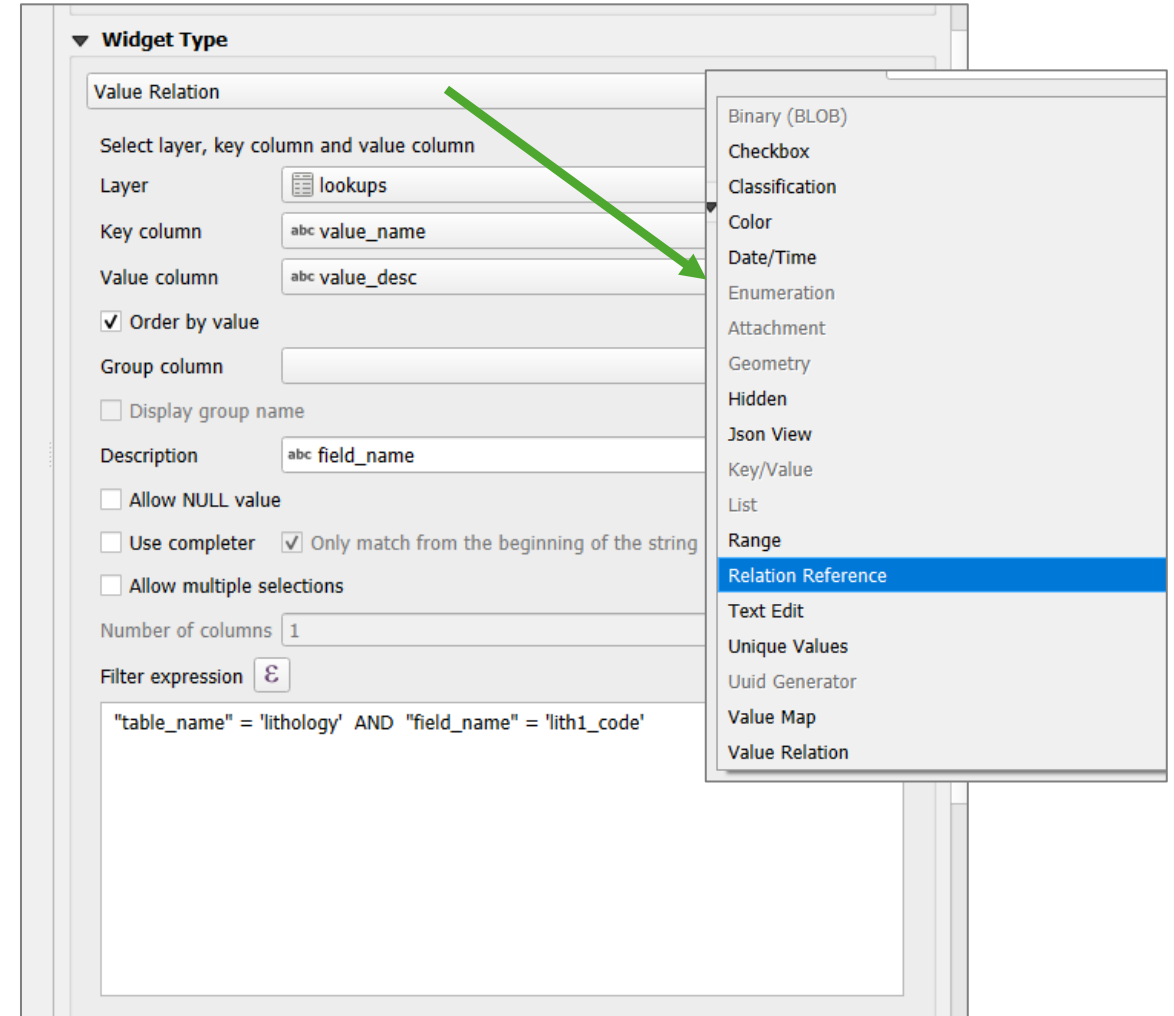
Setup Forms for each Layer

- Adjust form layout using combination of Tab, Group Box or Row options.
- Drop and drag fields on to relevant sections.
- Control visibility or collapse sections based on custom expressions (field values etc.).



Setup Data Entry Fields

- Forms can contain different widget types (dependent on underlying data type).
- Value relation widget allows you to reference a library table.
- Change fonts and colours to identify required fields and guide the user.
- Use aliases where field names are long or confusing.



Library Lookups

- Value relation widgets allow forms to reference another layer.
- Use the filter expression in the Attributes form for each value relation widget to restrict the dropdown options.
- Library table can store the key values plus a description to assist the user or improve sorting and filtering etc.

lookups — Features Total: 20, Filtered: 20, Selected: 0

	fid	table_name	value_name	value_desc	field_name	value_list
1	1	lithology	f	Undifferentiated Felsic	lith1_code	f (Undifferenti...
2	2	lithology	reg	Regolith Undifferentiated	lith1_code	reg (Regolith ...
3	3	lithology	rsl	Lower Saprolite	lith1_code	rsl (Lower Sa...
4	4	minerals	apy	Arsenopyrite	min1_code	apy (Arsenop...
5	5	minerals	bar	Barite	min1_code	bar (Barite)
6	6	minerals	cup	Cuprite	min1_code	cup (Cuprite)
7	7	lithology	mb	Basalt	lith1_code	mb (Basalt)
8	8	lithology	rsu	Upper Saprolite	lith1_code	rsu (Upper Sa...
9	9	lithology	fp	Felsic Porphyry	lith1_code	fp (Felsic Porp...
10	10	lithology	rgvl	residual gravel	lith1_code	rgvl (residual ...
11	11	lithology	adc	Adcumulate	lith1_texture	adc (Adcumul...

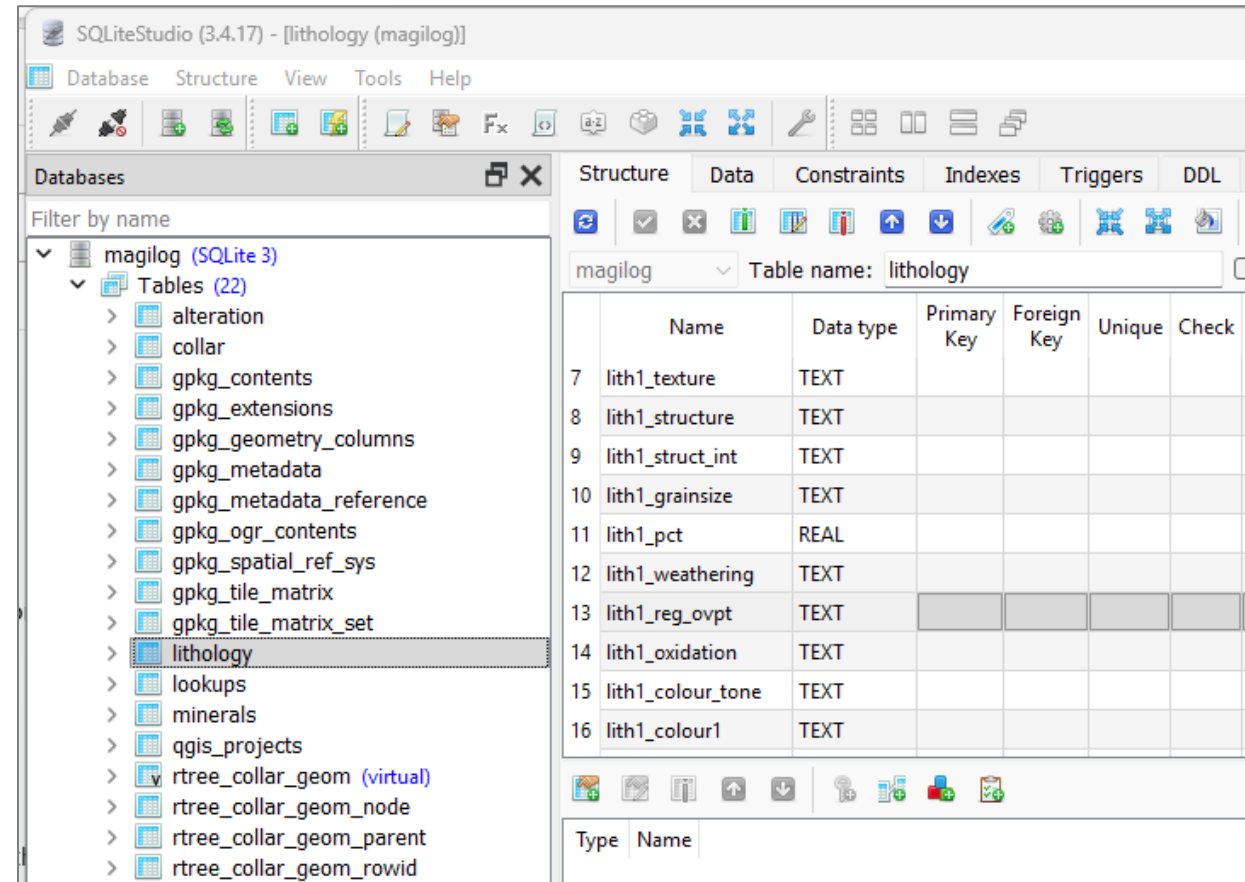
Number of columns 1

Filter expression ϵ

"table_name" = 'lithology' AND "field_name" = 'lith1_code'

Advanced Geopackage Setup

- Geopackages are SQLite databases.
- There are limitations to what modifications can be made in QGIS (once the table has been created).
- More options are available using SQLiteStudio software.
- Users can change data type, field order, field length etc.
- ODBC to SQLite is an option for simple connectivity.



Create Checks and Constraints

- Constraints can be enforced or just issue warnings.
- Some examples are:
 - ✓ `"dip" >= -90 AND "dip" <= 90`,
 - ✓ `"orig_azimuth" >= 0 AND "orig_azimuth" < 360`
 - ✓ `lith2_code is null OR lith1_code != lith2_code`
 - ✓ `"to_m" > "from_m" AND "to_m" >= 0 AND "to_m" <= attribute(get_feature('collar', 'hole_id', 'hole_id'), 'total_depth')`
- Remember case sensitivity!

The screenshot displays a configuration window for a database table, divided into two main sections: 'Constraints' and 'Defaults'.

Constraints Section:

- Not null:** The 'Not null' checkbox is checked. The 'Enforce not null constraint' checkbox is unchecked.
- Unique:** The 'Unique' checkbox is unchecked. The 'Enforce unique constraint' checkbox is unchecked.
- Expression:** The expression field contains the SQL logic: `"from_m" >= 0 AND "from_m" <= attribute(get_feature('collar', 'hole_id', 'hole_id'), 'total_depth')`. There are icons for clearing the field (X) and saving (E).
- Expression description:** The description field contains the text: "From depth must be non-negative and ≤ hole total depth".
- Enforce expression constraint:** This checkbox is unchecked.

Defaults Section:

- Default value:** The field contains the SQL expression: `coalesce(maximum("to_m", group_by:="hole_id"),0)`. There are icons for clearing the field (X) and saving (E).
- Preview:** The preview field shows the value `2`.
- Warning:** A yellow warning triangle icon is followed by the text: "Using fields in a default value expression only works if 'Apply default value on update' is checked."
- Apply default value on update:** This checkbox is unchecked.

Let's go Logging

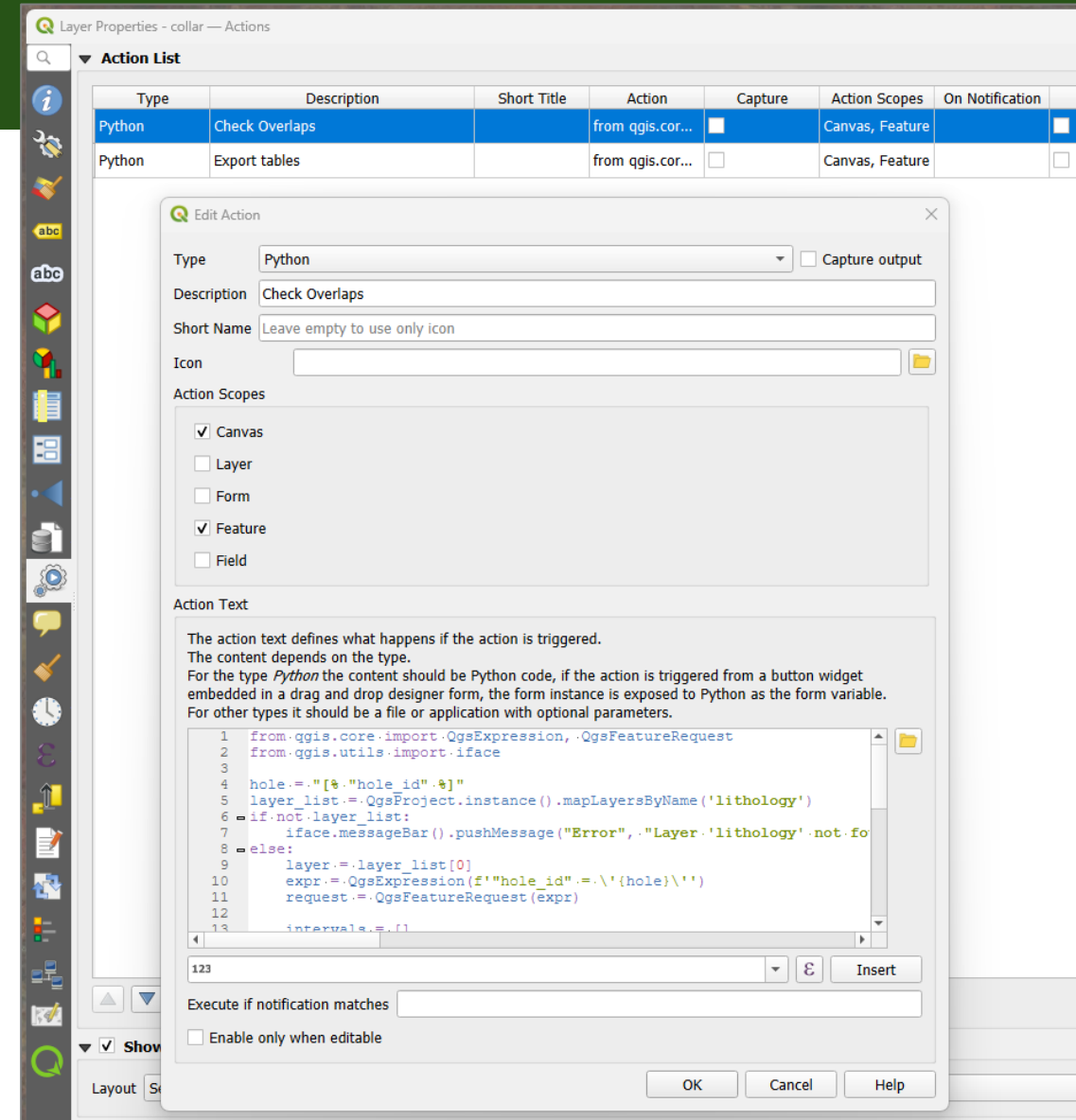
- User can select the form or a datasheet data entry layout.
- Data entry errors are highlighted with error or warning messages.
- Holes or downhole intervals selected via the list boxes on the left.
- Setup conditional formatting to assist logging or identify potential issues.

The screenshot displays the 'collar' application interface. The top bar shows 'collar — Features Total: 6, Filtered: 6, Selected: 0'. The left sidebar lists features: test003, test001, test002, test004, test005, and test006. The main window has three tabs: 'collar information', 'downhole logging' (active), and 'downhole sampling'. The 'downhole logging' tab contains a 'lithology' section with a 'collar_lithology' list box and an 'Expression' field. Below this is a 'Lithology' form with fields for 'Interval' (from_m, to_m, hole_id) and 'Lith1' (lith1_code, lith2_code, lith1_texture, lith1_structure, lith1_struct_int, lith1_grainsize, lith1_pct, lith1_weathering, lith1_reg_ovpt, lith1_oxidation, lith1_colour_tone, lith1_colour1). A 'Lith2' section is also visible. A data table is shown at the bottom, listing logging intervals for hole 'test001'.

	hole_id	from_m	to_m	lith1_code	
1	test001	0	9.8	reg (Regolith Undifferentiated)	NO
2	test001	9.8	15.2	mb (Basalt)	NO
3	test001	15.2	23.1	reg (Regolith Undifferentiated)	rsi
4	test001	23.1	45.8	rgvl (residual gravel)	rsu

Actions

- **Actions** (think macros) - perform tasks or advanced functionality on demand to streamline or automate your workflows.
- Can run python scripts (meaning possibilities virtually unlimited!)
- Actions in this example:
 - ✓ Export all tables to csv's with hole ID prefix and date suffix for a selected hole ID.
 - ✓ Check for overlapping intervals in all tables.
 - ✓ Create samples with automated QAQC



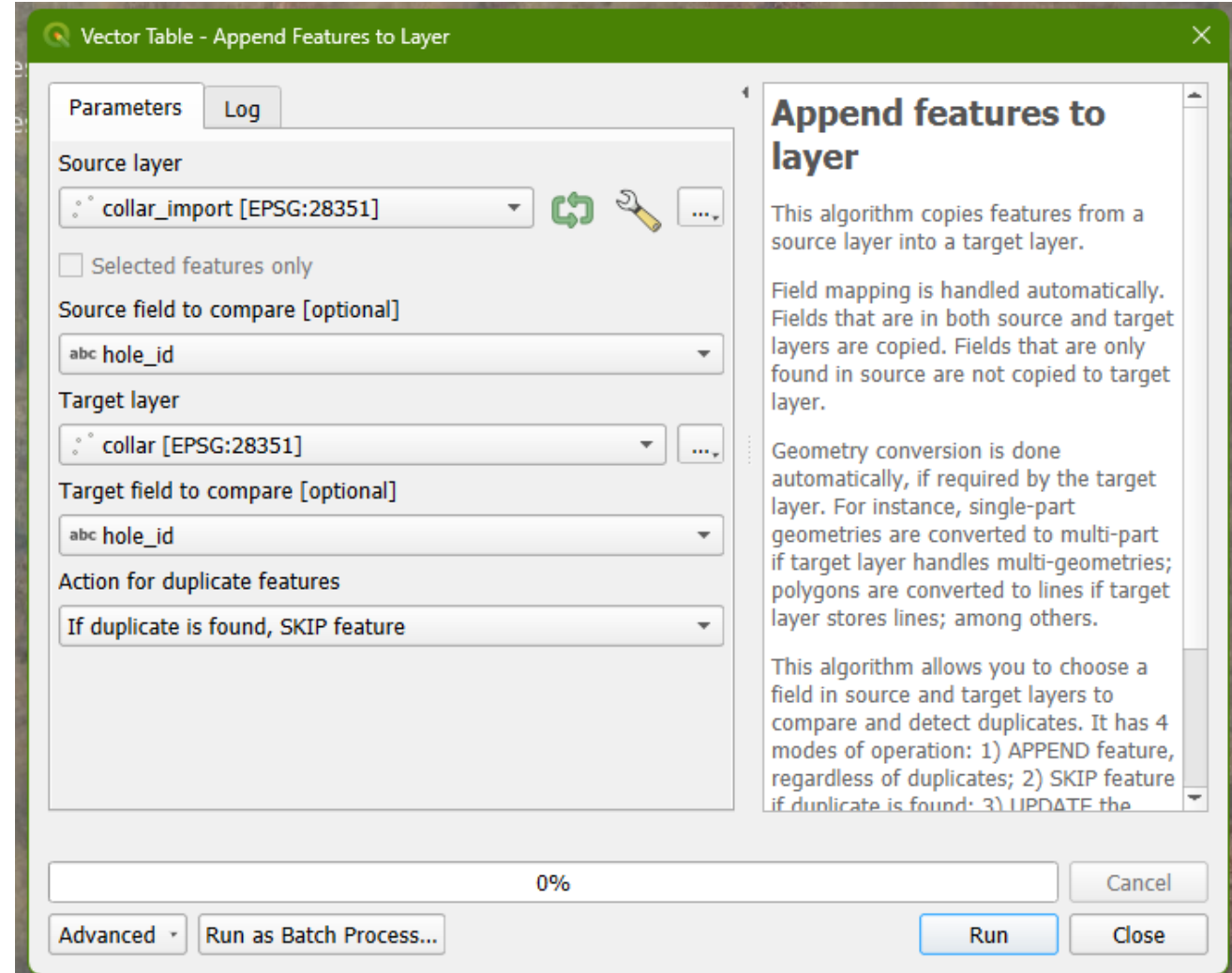
View your Logging on Drill Trace

- Use the **Geoscience** plugin to desurvey the hole and plot downhole logging.
- Display the drill trace (dotted yellow line) to show where no logging exists.
- Dip and azi can be planned from the collar table.
- Check sections while you log.
- View the trace in 3D alongside other drillholes



Easy Data Imports (to get started)

- **Append Features to Layer** plugin allows you to easily import data.
- Plugin checks the target field and allows you to skip, update or append records.
- Columns must match for import to work correctly (mismatched fields ignored).



Next Steps

- Field Testing – currently trialling on ancient Toughbooks now running Linux and KDE Plasma (retired due to win 10). Seems to work great!
- Custom plugins/actions – drillhole and logging data synchronisation.
- Auto populate or validate information spatially – Lease ID, POW, DataSet/Project Code etc.
- Integration with PostGIS and geological databases (library table updates, planned collar imports, logging data synchronisation etc.).
- QField – initial testing suggests it does not support key functionality and likely not able to support this workflow adequately.

THANK YOU!

jeremy@terradata.com.au