European Environment Agency

Spatial data in the scope of the END reporting data – guidelines

Version: 1.0

Date: 03 February 2023

Author: Stefania Morrone (Epsilon Italia)

Legal notice

The contents of this publication do not necessarily reflect the official opinions of the European Commission or other institutions of the European Union. Neither the European Environment Agency, the European Topic Centre on Air pollution, transport, noise and industrial pollution nor any person or company acting on behalf of the Agency or the Topic Centre is responsible for the use that may be made of the information contained in this report.

Copyright notice

Information about the European Union is available on the Internet. It can be accessed through the Europa server (www.europa.eu).

Author(s)

Stefania Morrone (Epsilon Italia)

Contributor(s)

Miquel Sáinz de la Maza (UAB, ETC/HE)

Contents

1 Introduction

This short document is aimed at facilitating the successful delivery of spatial data in the scope of the END reporting.

Based on experience in importing sample datasets from countries into RN3 (ReportNet 3), the most common errors encountered for spatial tables and suggestions for their resolution are described in the following sections. In particular,

- Section 2 focuses on non-proprietary software e.g. QGIS, DB Browser for SQLite.
- Section 3 focuses on the use of FME.

It is recommended to check the validity of the GeoPackage spatial tables and fix possible issues, e.g., by following the suggestions in this document, before submitting the reporting data.

2 Validation of spatial tables: common errors

Invalid geometries

Error Messages by RN3:

- “Self-intersection”
- “Too few points in geometry component”

How to solve:

The two options provided below show **alternative** ways to validate and correct geometries in QGIS, using respectively:

- The DB Manager and its SQL window (Option 1)
- The geometry plugins (Option 2).

Option 1: use SQL statements

In the *Database* menu, select *DB Manager* **>** *GeoPackage* and then press the right button of the mouse to create a connection to the GeoPackage to be tested (see Figure 1).

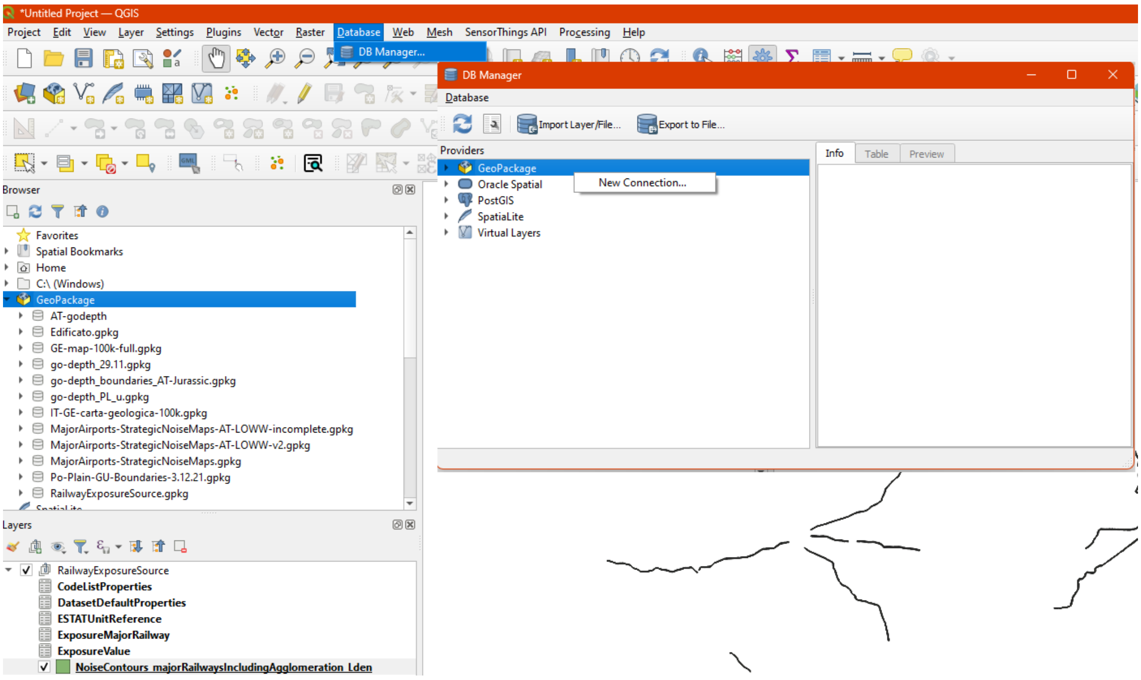
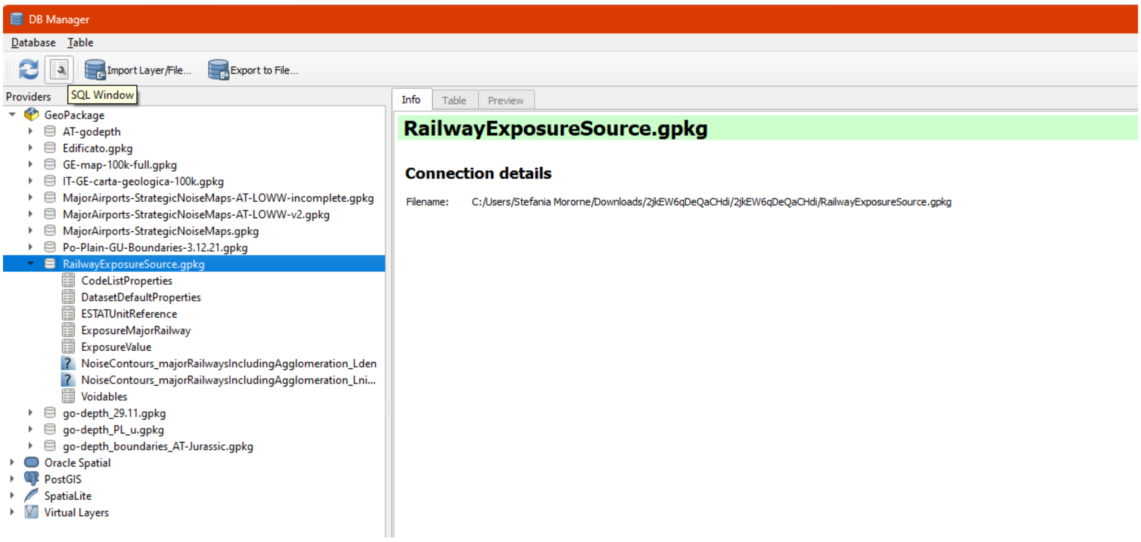


Figure 1: DB Manager

Under the *Providers* section, select the GeoPackage connection just created. By clicking the wrench icon (see Figure 2), the SQL window tab appears.


DB Manager
Database Table
Import Layer/File...
Export to File...
Providers
SQL Window
GeoPackage
AT-godepth
Edificato.gpkg
GE-map-100k-full.gpkg
IT-GE-carta-geologica-100k.gpkg
MajorAirports-StrategicNoiseMaps-AT-LOWW-incomplete.gpkg
MajorAirports-StrategicNoiseMaps-AT-LOWW-v2.gpkg
MajorAirports-StrategicNoiseMaps.gpkg
Po-Plain-GU-Boundaries-3.12.21.gpkg
RailwayExposureSource.gpkg
CodeListProperties
DatasetDefaultProperties
ESTATUnitReference
ExposureMajorRailway
ExposureValue
NoiseContours_majorRailwaysIncludingAgglomeration_Lden
NoiseContours_majorRailwaysIncludingAgglomeration_Lni...
Voidables
go-depth_29.11.gpkg
go-depth_PL_u.gpkg
go-depth_boundaries_AT-Jurassic.gpkg
Oracle Spatial
PostGIS
SpatiaLite
Virtual Layers
Info Table Preview

RailwayExposureSource.gpkg

Connection details

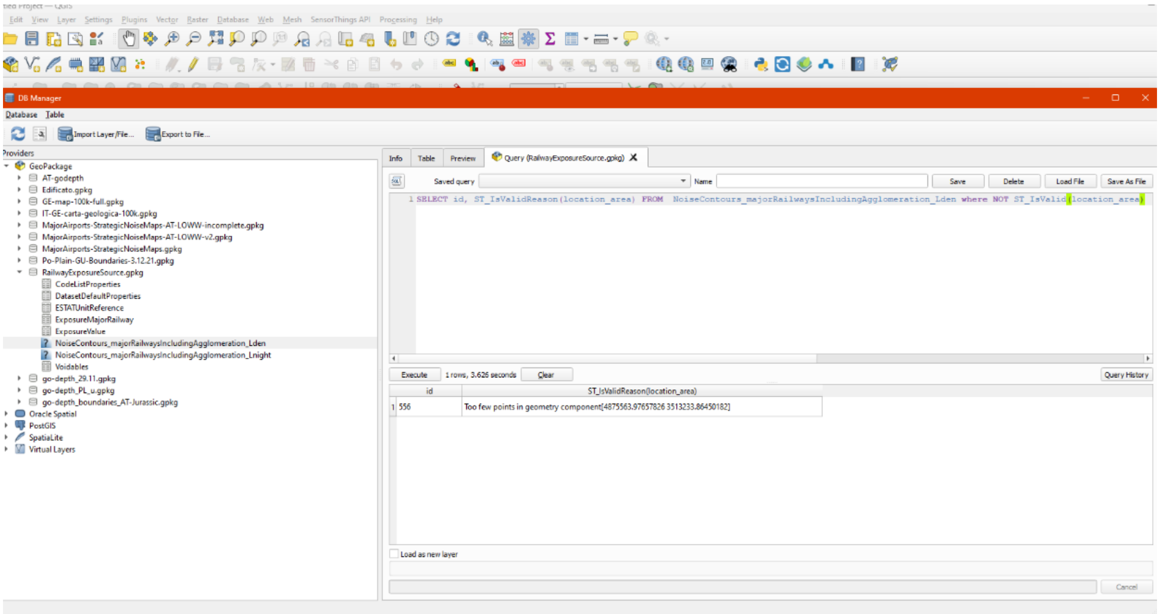
Filename: C:/Users/Stefania Mororne/Downloads/2jkEW6qDeQaCHdi/2jkEW6qDeQaCHdi/RailwayExposureSource.gpkg

Figure 2: Open SQL window

In the SQL window, run sequentially the two queries below to list and possibly fix any errors.

1) Error detection query: lists the ids of records with invalid geometries and provides error details (replace *table* and *geometry* with the name of the table to test and the name of the geometry column, respectively):

```
SELECT id, ST_IsValidReason(geometry) FROM  table where NOT ST_IsValid(geometry)
```



DB Manager
Database Table
Import Layer/File... Export to File...
Providers
GeoPackage
AT-godepth
Edificato.gpkg
GE-map-100k-full.gpkg
IT-GE-carta-geologica-100k.gpkg
MajorAirports-StrategicNoiseMaps-AT-LOWW-incomplete.gpkg
MajorAirports-StrategicNoiseMaps-AT-LOWW-v2.gpkg
MajorAirports-StrategicNoiseMaps.gpkg
Po-Plain-GU-Boundaries-3.12.21.gpkg
RailwayExposureSource.gpkg
CodeListProperties
DatasetDefaultProperties
ESTATUnitReference
ExposureMajorRailway
ExposureValue
NoiseContours_majorRailwaysIncludingAgglomeration_Lden
NoiseContours_majorRailwaysIncludingAgglomeration_Lnight
Voidables
go-depth_29.11.gpkg
go-depth_PL_u.gpkg
go-depth_boundaries_AT-Jurassic.gpkg
Oracle Spatial
PostGIS
SpatiaLite
Virtual Layers
Info Table Preview Query (RailwayExposureSource.gpkg)
Saved query Name Save Delete Load File Save As File
Execute Clear Query History
Load as new layer
Cancel

```
SELECT id, ST_IsValidReason(location_area) FROM  NoiseContours_majorRailwaysIncludingAgglomeration_Lden where NOT ST_IsValid(location_area)
```

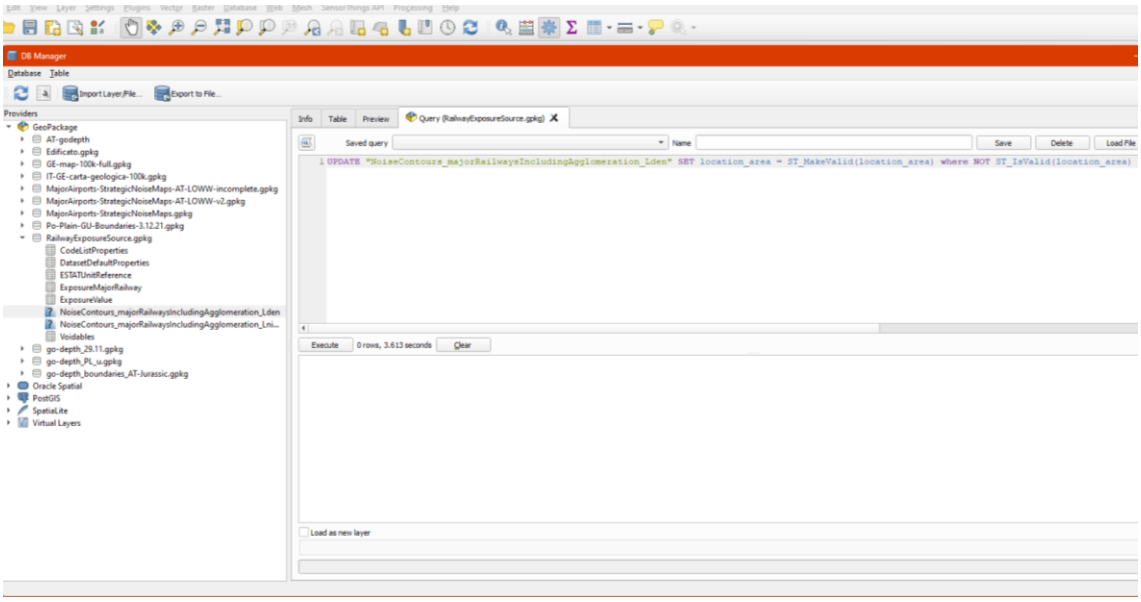
1 rows, 3.626 seconds

	id	ST_IsValidReason(location_area)
1	556	Too few points in geometry component[4875563.97657826 3513233.86450182]

Figure 3: Error detection query in the SQL window

2) Error fixing query: updates the table to fix the errors:

UPDATE *table* SET *geometry* = ST_MakeValid(*geometry*) where NOT ST_IsValid(*geometry*)



DB Manager
Database Table
Import Layer/File... Export to File...
Providers
GeoPackage
AT-godepth
Edificato.gpkg
GE-map-100k-full.gpkg
IT-GE-carta-geologica-100k.gpkg
MajorAirports-StrategicNoiseMaps-AT-LOWW-incomplete.gpkg
MajorAirports-StrategicNoiseMaps-AT-LOWW-v2.gpkg
MajorAirports-StrategicNoiseMaps.gpkg
Po-Plain-GU-Boundaries-3.12.21.gpkg
RailwayExposureSource.gpkg
CodeListProperties
DatasetDefaultProperties
ESTATUnitReference
ExposureMajorRailway
ExposureValue
NoiseContours_majorRailwaysIncludingAgglomeration_Lden
NoiseContours_majorRailwaysIncludingAgglomeration_Lni...
Voidables
go-depth_29.11.gpkg
go-depth_PL_u.gpkg
go-depth_boundaries_AT-Jurassic.gpkg
Oracle Spatial
PostGIS
SpatiaLite
Virtual Layers
Info Table Preview Query (RailwayExposureSource.gpkg)
Saved query Name Save Delete Load File
Execute Clear
Load as new layer

```
UPDATE "NoiseContours_majorRailwaysIncludingAgglomeration_Lden" SET location_area = ST_MakeValid(location_area) where NOT ST_IsValid(location_area)
```

0 rows, 3.613 seconds

Figure 4: Error Fixing query in the SQL window

Note: should the error detection query find null geometry errors, corresponding records must be deleted executing the following statement in the SQL window:

DELETE FROM *table* WHERE *geometry* IS NULL

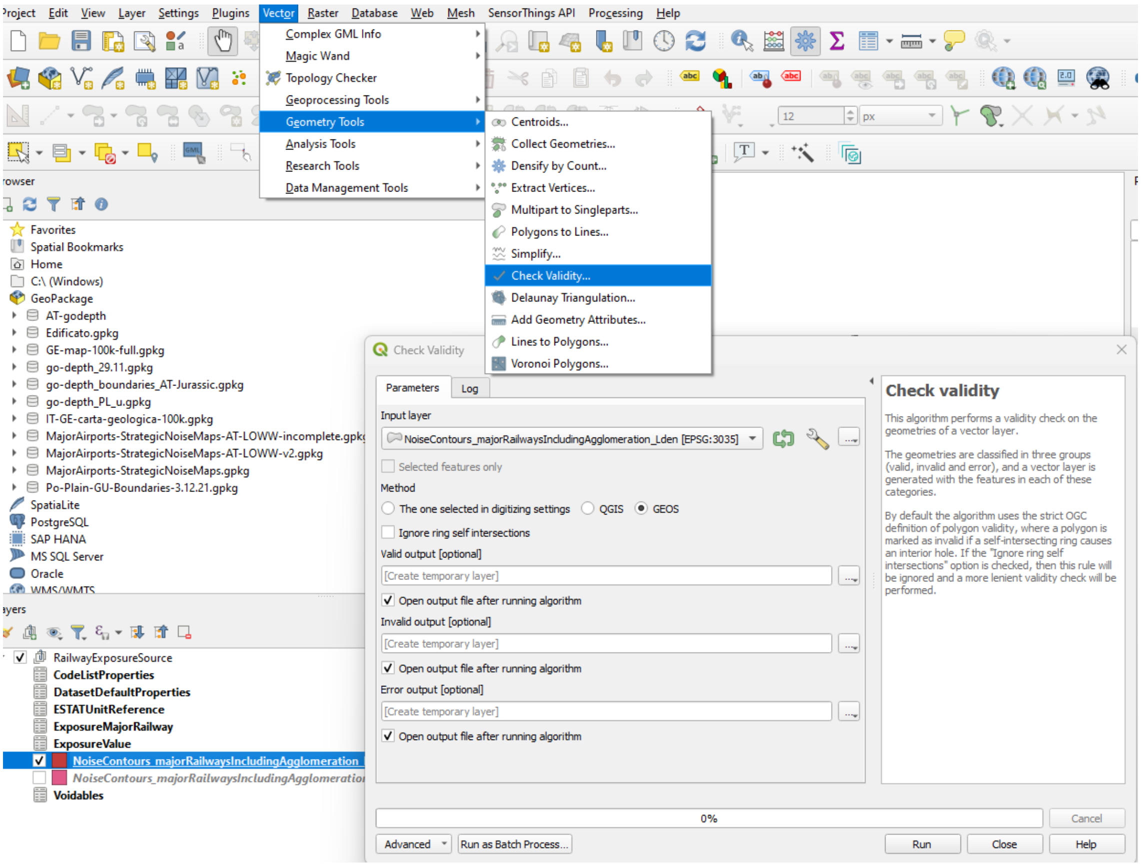
Once the second query (and, if needed, the DELETE statement) has been executed, the error detection query must be run again to assess whether all errors have been corrected.

Should any errors persist, either delete the failing geometries (execute the DELETE statement for failing ids) or operate manually:

1. open the GeoPackage in QGIS.
2. Zoom to the failing geometries (select the related ids in the Attribute Table).
3. Edit geometries as needed and re-test.

Option 2: use geometry plugins

1. Open your GeoPackage in QGIS.
2. Select the spatial layer to test.
3. In the *Vector* menu select '*Geometry Tools*'.
4. Select '*Check Validity*' (keep all parameters in the pop-up window to default) and click '*Run*'.



Check Validity

Parameters | Log

Input layer

NoiseContours_majorRailwaysIncludingAgglomeration_Lden [EPSG:3035]

□ Selected features only

Method

○ The one selected in digitizing settings ○ QGIS ● GEOS

□ Ignore ring self intersections

Valid output [optional]

[Create temporary layer]

☑ Open output file after running algorithm

Invalid output [optional]

[Create temporary layer]

☑ Open output file after running algorithm

Error output [optional]

[Create temporary layer]

☑ Open output file after running algorithm

0%

Check validity

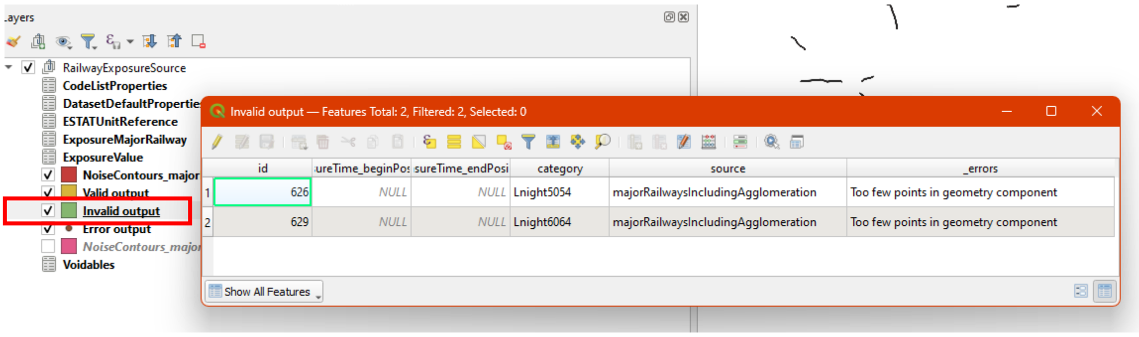
This algorithm performs a validity check on the geometries of a vector layer.

The geometries are classified in three groups (valid, invalid and error), and a vector layer is generated with the features in each of these categories.

By default the algorithm uses the strict OGC definition of polygon validity, where a polygon is marked as invalid if a self-intersecting ring causes an interior hole. If the "Ignore ring self intersections" option is checked, then this rule will be ignored and a more lenient validity check will be performed.

Figure 5: Check Validity

If errors are detected, the record ids as well as the type of errors are listed in the "Invalid output" layer.



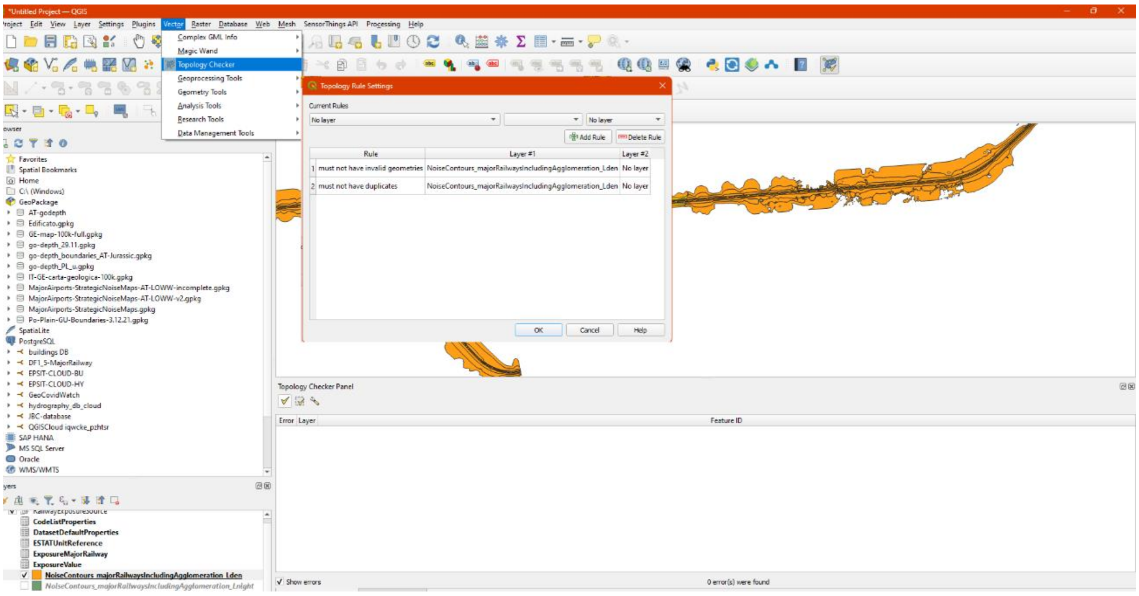
Invalid output — Features Total: 2, Filtered: 2, Selected: 0

	id	ureTime_beginPos	sureTime_endPosi	category	source	_errors
1	626	NULL	NULL	Lnight5054	majorRailwaysIncludingAgglomeration	Too few points in geometry component
2	629	NULL	NULL	Lnight6064	majorRailwaysIncludingAgglomeration	Too few points in geometry component

Figure 6: results of the Check Validity operation

Alternatively, you might perform more in-depth analysis using the *Topology Checker* plugin and specify the different topology rules the selected layer will be validated against e.g., “Must not have invalid geometries”, “Must not overlap with:” etc.

To run the *Topology Checker*, in the *Vector* menu select *Topology Checker* and, in the topology checker panel that will pop up, click on the wrench icon to show the *Topology Rule Settings* window. Choose the layer to test and the select the rules it will be tested against, adding each rule one by one. After defining rules, in topology checker panel click Validate All or Validate Extent. The errors will be shown in topology checker panel list (do not forget to check the *Show errors* box).



Topology Rule Settings

Current Rules

No layer | No layer

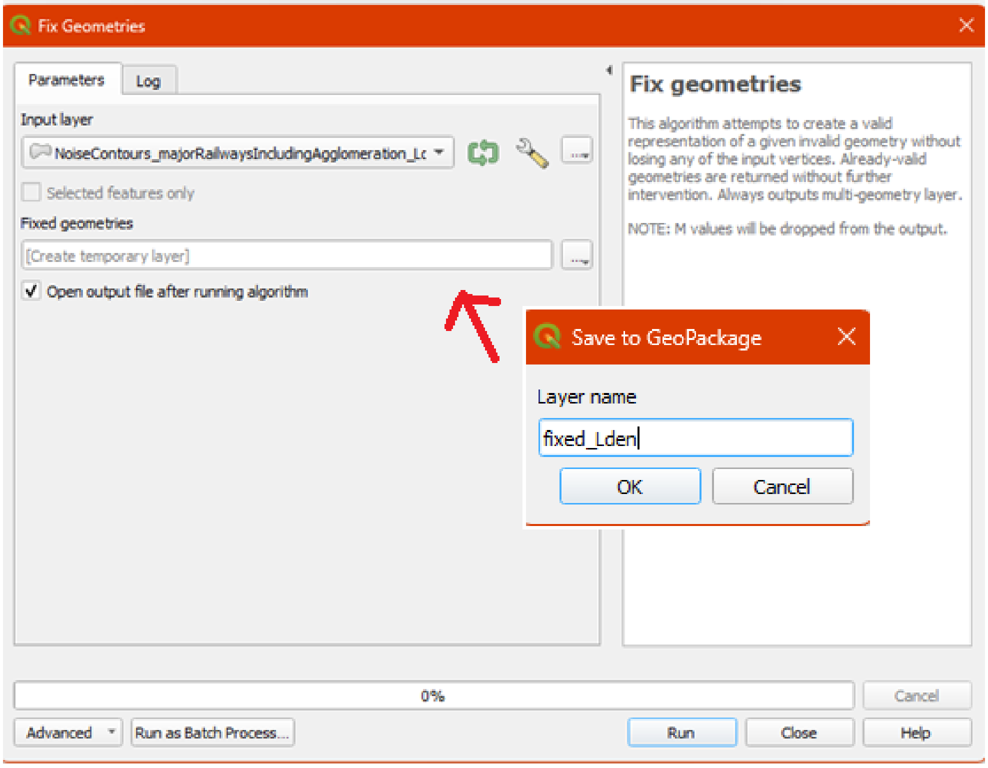
	Rule	Layer #1	Layer #2
1	must not have invalid geometries	NoiseContours_majorRailwaysIncludingAgglomeration_Lden	No layer
2	must not have duplicates	NoiseContours_majorRailwaysIncludingAgglomeration_Lden	No layer

Figure 7 Topology Checker plugin

Regardless of the plugin used to detect the errors, in order to fix them you can run the *Fix Geometries* algorithm (see Figure 8). This can be accessed in the *Processing Toolbox* (Processing menu > *Toolbox*), under *Vector geometry* section.

In the related pop-up window:

1. select the input layer (e.g., the Lden table).
2. Specify the input GeoPackage file as output layer and provide then a layer name (e.g., fixed_Lden).
3. Press *Run*.


Fix Geometries

Parameters | Log

Input layer

NoiseContours_majorRailwaysIncludingAgglomeration_Lc

☐ Selected features only

Fixed geometries

[Create temporary layer]

☑ Open output file after running algorithm

Fix geometries

This algorithm attempts to create a valid representation of a given invalid geometry without losing any of the input vertices. Already-valid geometries are returned without further intervention. Always outputs multi-geometry layer.

NOTE: M values will be dropped from the output.

Save to GeoPackage

Layer name

fixed_Lden

0%

Figure 8: QGIS Fix Geometry

The returned layer with fixed geometries (in our example the ‘fixed_Lden’) should then replace the corresponding failing layer in the original GeoPackage. This can be done using the DB Manager:

1. open the connection to the GeoPackage file (you should first create it if not existent).
2. Delete the layer with failing geometries.
3. Rename the layer with fixed geometries to the same name as the deleted layer.

POLYGON vs MULTIPOLYGON

Error Message by RN3:

- “At least one geometry must be provided”

This error is reported by RN3 when no geometry is provided or when geometry cannot be imported into the system. It has been observed that the latter case occurs:

- when the GeoPackage spatial table has a POLYGON geometry type while (some of) the contained records are MULTIPOLYGON.
- when the GeoPackage spatial table has ‘GEOMETRY’ geometry type (regardless of the geometry type of the contained data).

NOTE: if you use *hale studio* for creating the reporting datasets, be aware that the software would not complain /raise any warning about above circumstances.

How to solve:

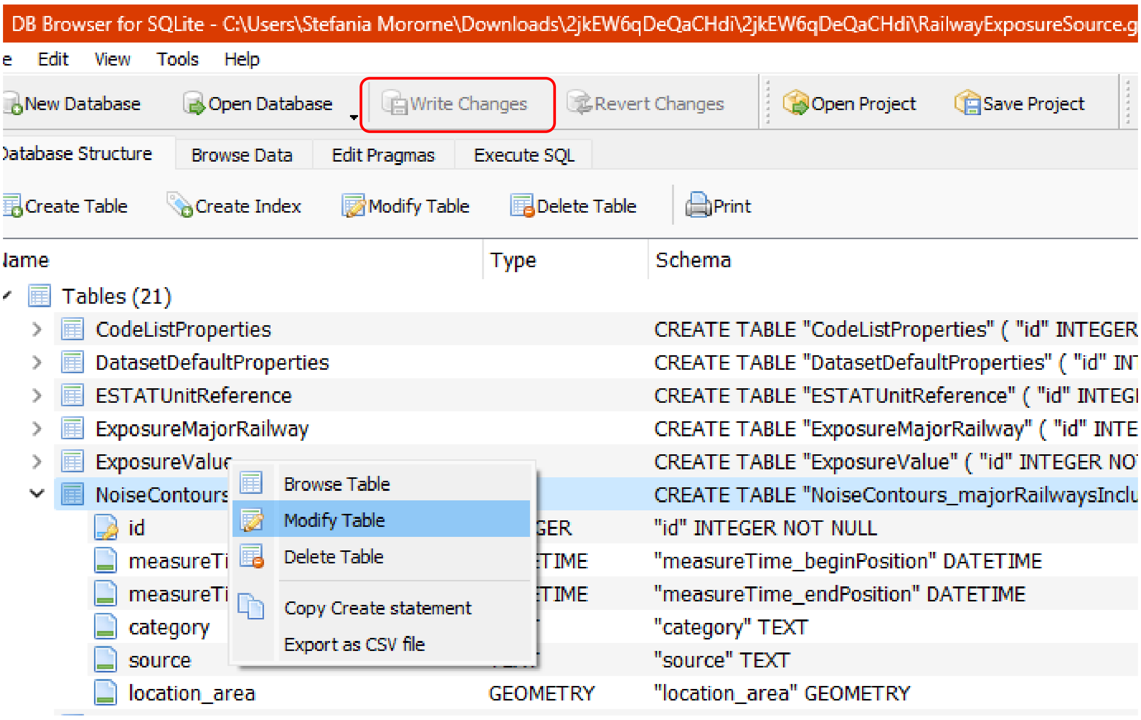
If input data contains MULTIPOLYGON geometries, it is possible to download the corresponding MULTIPOLYGON GeoPackage template now available from the Eionet END page and (re)transform the reporting data using the MULTIPOLYGON template as target schema.

However, if you have already created a delivery with POLYGON template and do not want to recreate it, you can follow one of the two options provided below:

Option 1:

Use a GeoPackage administration tool - e.g. [DB Browser for SQLite]() - and

1. open the GeoPackage database.
2. Modify the geometry column type to MULTIPOLYGON.
3. Write changes.
4. Delete null geometries if present using the SQL window and write changes.



DB Browser for SQLite - C:\Users\Stefania Morome\Downloads\2jkEW6qDeQaCHdi\2jkEW6qDeQaCHdi\RailwayExposureSource.g

lame	Type	Schema
Tables (21)		
CodeListProperties		CREATE TABLE "CodeListProperties" ("id" INTEGER
DatasetDefaultProperties		CREATE TABLE "DatasetDefaultProperties" ("id" IN
ESTATUnitReference		CREATE TABLE "ESTATUnitReference" ("id" INTEGI
ExposureMajorRailway		CREATE TABLE "ExposureMajorRailway" ("id" INTE
ExposureValue		CREATE TABLE "ExposureValue" ("id" INTEGER NO
NoiseContours		CREATE TABLE "NoiseContours_majorRailwaysInclu
id	GER	"id" INTEGER NOT NULL
measureTi	TIME	"measureTime_beginPosition" DATETIME
measureTi	TIME	"measureTime_endPosition" DATETIME
category		"category" TEXT
source		"source" TEXT
location_area	GEOMETRY	"location_area" GEOMETRY

Figure 9: DB Browser for SQLite

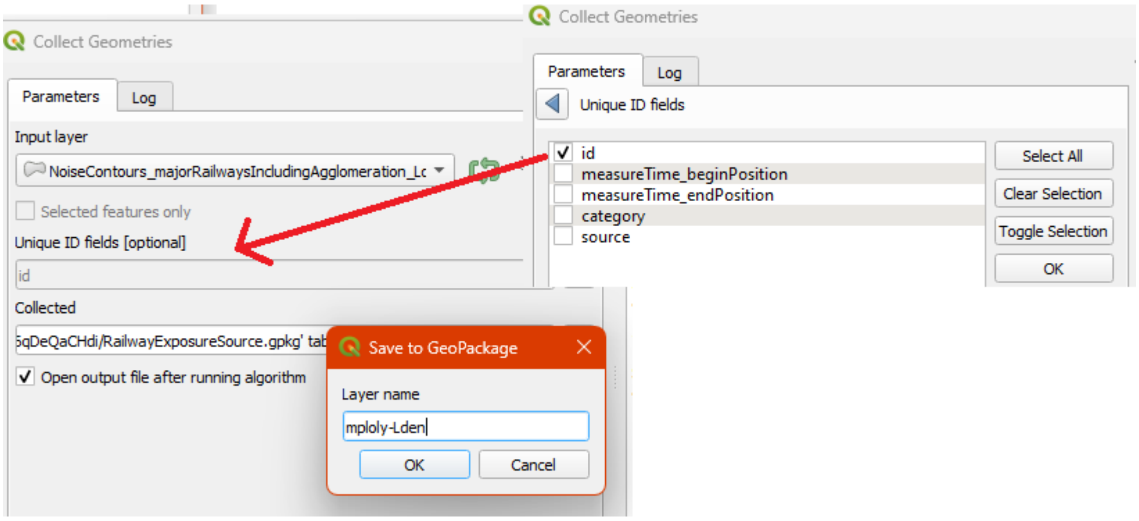
Option 2:

Use QGIS:

1. delete null geometries if present e.g., using the DB Manager and the SQL window (see DELETE example in the above section)
2. Transform POLYGONS into MULTIPOLYGONS using the '*Collect geometries*' algorithm. This can be accessed in the Processing Toolbox (Processing menu > Toolbox), under *Vector geometry* section.

   In the pop-up window:

   a. select the input layer (e.g., the Lden table).
   b. Specify the input GeoPackage file as output layer and then provide an easily recognisable layer name (e.g., mpoly_Lden), then press Run.



Input layer: NoiseContours_majorRailwaysIncludingAgglomeration_Lc

□ Selected features only

Unique ID fields [optional]: id

Collected: 5qDeQaCHdi/RailwayExposureSource.gpkg' tab

☑ Open output file after running algorithm

Unique ID fields

☑ id
□ measureTime_beginPosition
□ measureTime_endPosition
□ category
□ source

Layer name: mploly-Lden

Figure 10 Collect geometry plugin.

The returned MULTIPOLYGON layer (in our example the ‘mpoly_Lden’) should then replace the corresponding failing layer in the input GeoPackage. This can be done using the *DB Manager*:

1. open the connection to the GeoPackage file (you should first create it if not existent).
2. Delete the layer with failing geometries.
3. Rename the layer with fixed geometries to the same name as the deleted layer.

3 Geometry Validation in FME

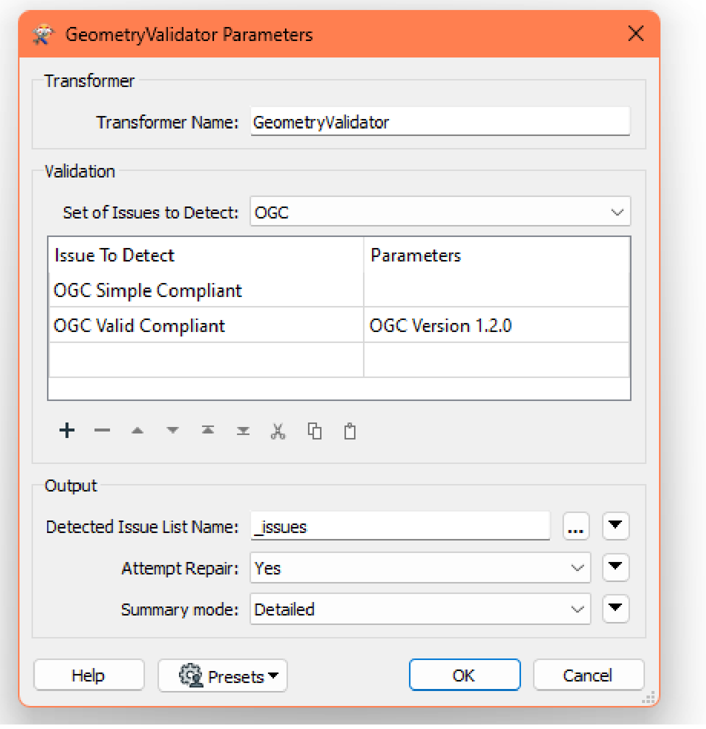
Error Messages by RN3:

- “Self-intersection”
- “Too few points in geometry component”

Geometry validation can be executed running the *GeometryValidator* transformer against the spatial tables of the geopackage file. The transformer detects and describes issues in input features, and optionally repairs them.

In the parameters pop-up window:

- Select validation against OGC requirements.
- Set Attempt Repair =’Yes’ if you’d like FME try to fix errors



GeometryValidator Parameters

Transformer

Transformer Name: GeometryValidator

Validation

Set of Issues to Detect: OGC

Issue To Detect	Parameters
OGC Simple Compliant	
OGC Valid Compliant	OGC Version 1.2.0

Output

Detected Issue List Name: _issues

Attempt Repair: Yes

Summary mode: Detailed

Figure 11 GeometryValidator Parameters pop-up window

The screenshots below respectively show a typical geometry validation for a geopackage spatial table (Figure 12) and the analysis of the validation results in the Data Inspector visual tool (Figure 13).

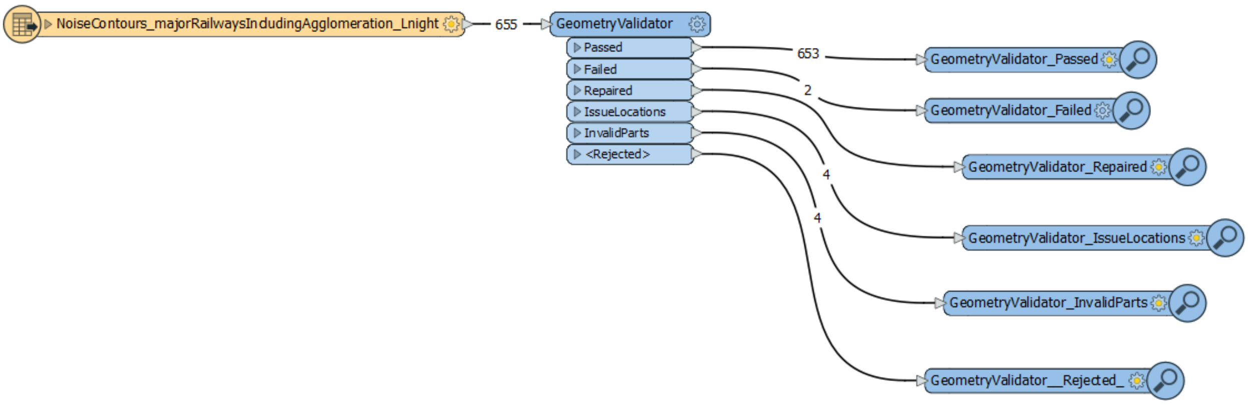
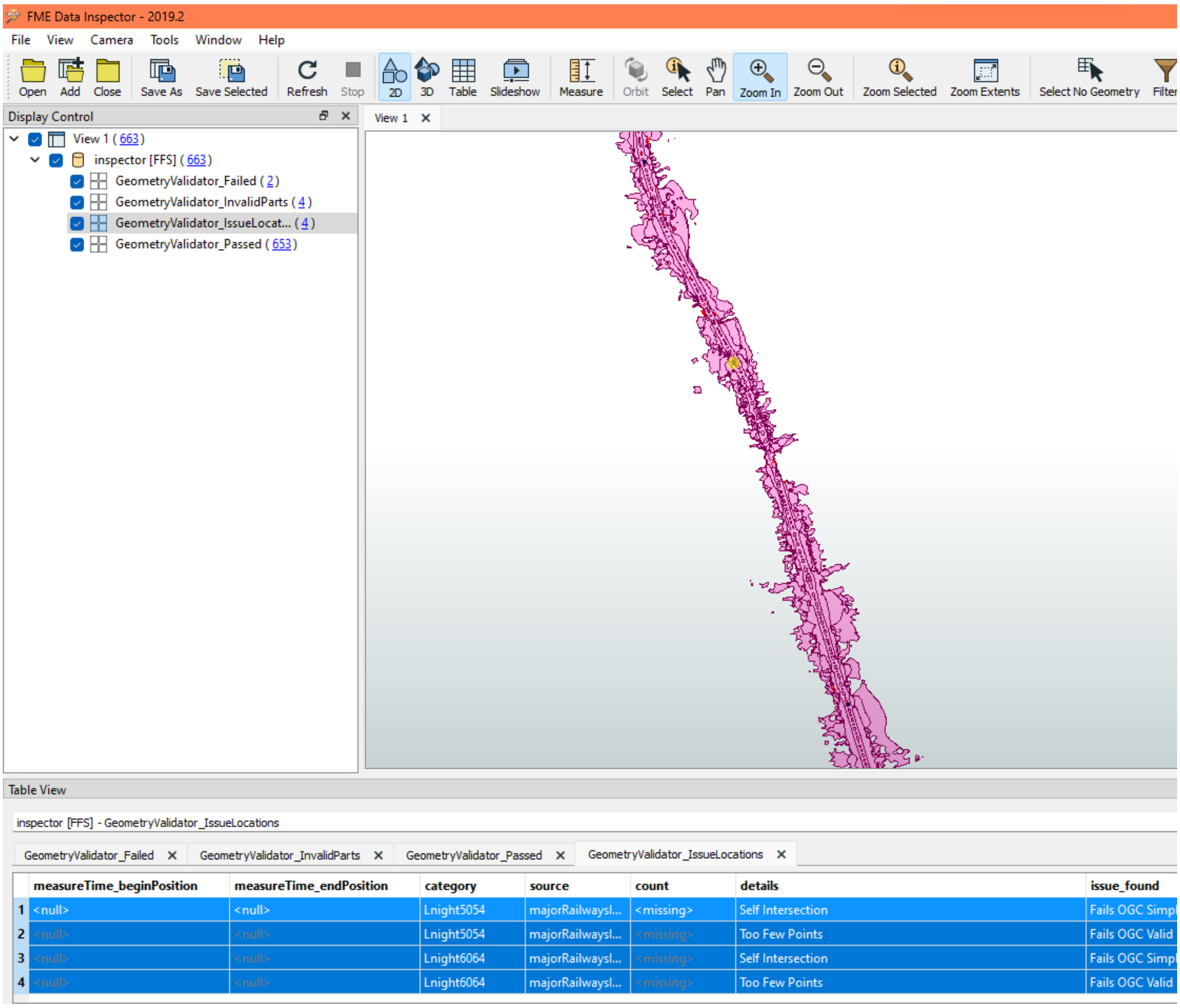

NoiseContours_majorRailwaysIncludingAgglomeration_Lnight
655
GeometryValidator
Passed
Failed
Repaired
IssueLocations
InvalidParts
<Rejected>
653
2
4
4
GeometryValidator_Passed
GeometryValidator_Failed
GeometryValidator_Repaired
GeometryValidator_IssueLocations
GeometryValidator_InvalidParts
GeometryValidator__Rejected_

Figure 12 Connect GeometryValidator to spatial table



inspector [FFS] - GeometryValidator_IssueLocations

	measureTime_beginPosition	measureTime_endPosition	category	source	count	details	issue_found
1	<null>	<null>	Lnight5054	majorRailwaysl...	<missing>	Self Intersection	Fails OGC Simpl
2	<null>	<null>	Lnight5054	majorRailwaysl...	<missing>	Too Few Points	Fails OGC Valid
3	<null>	<null>	Lnight6064	majorRailwaysl...	<missing>	Self Intersection	Fails OGC Simpl
4	<null>	<null>	Lnight6064	majorRailwaysl...	<missing>	Too Few Points	Fails OGC Valid

Figure 13 Analyse errors with the FME Data Inspector

4 Record duplication

It has been noted that in cases where invalid geometries are present, the number of records imported into RN3 may not match the number of records in the input data (either over or under). This will be investigated further, but it is worth highlighting here that by correcting the invalid records the problem disappears.

5 Compact GeoPackage database

Once the corrections on the GeoPackage files are completed, it is highly recommended to perform a ‘vacuum’ operation to reduce the file size before it is submitted to reporting.

Find below explanations of how to perform a vacuum operation respectively in QGIS and in DB Browser for SQLite tools.

Option 1: QGIS

In the Browser panel on the left, select the connection to your GeoPackage file, press the right button of the mouse and select “Compact Database (VACUUM)” option.

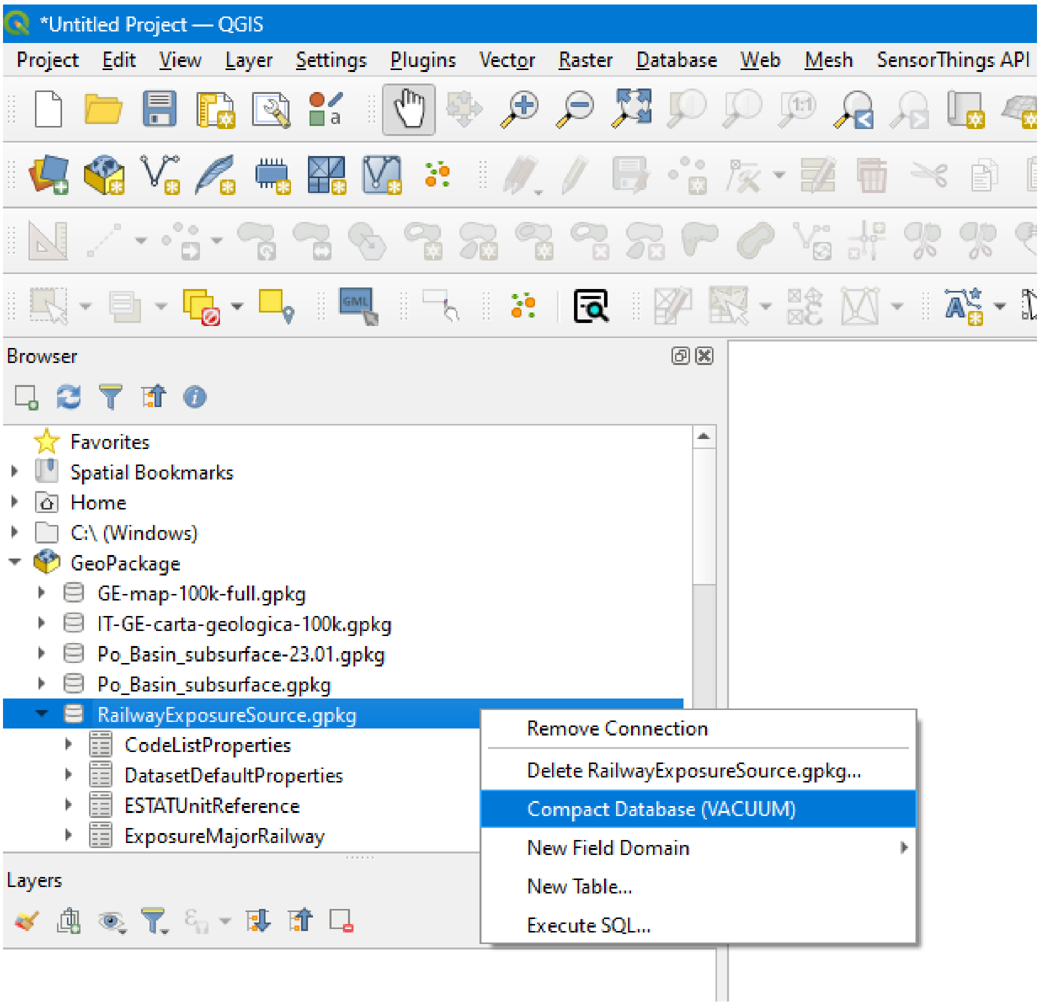
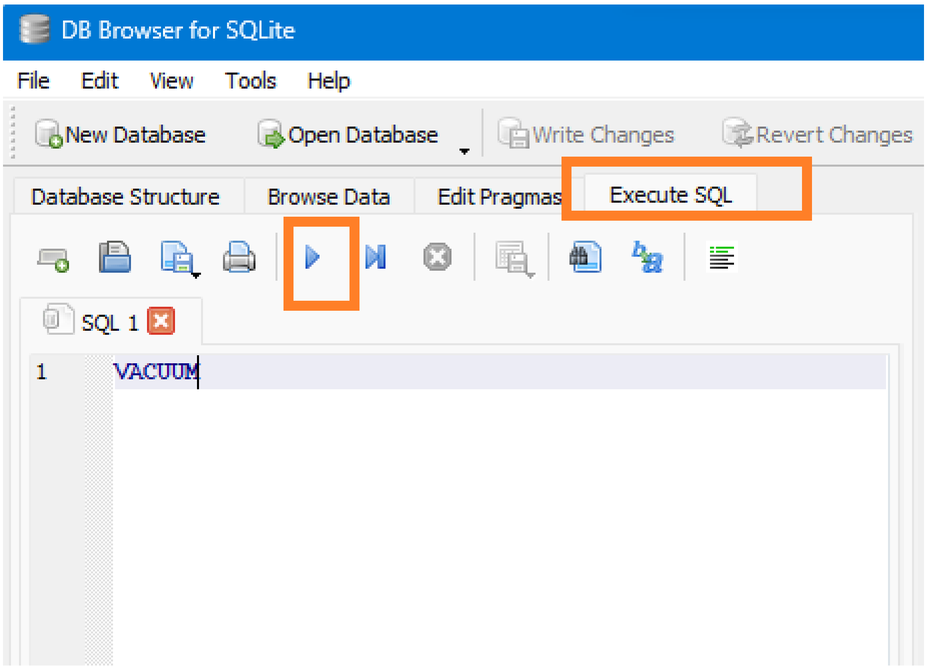


Figure 14: Vacuum operation in QGIS

Option 2: DB Browser for SQLite

1. Open the GeoPackage file (‘Open Database’ menu)
2. In the ‘Execute SQL’ panel, run a “vacuum “command (small arrow).



```
VACUUM
```

Figure 15: vacuum command in DB Browser