FME Database Mastery

Take control of your data with FME and Databases

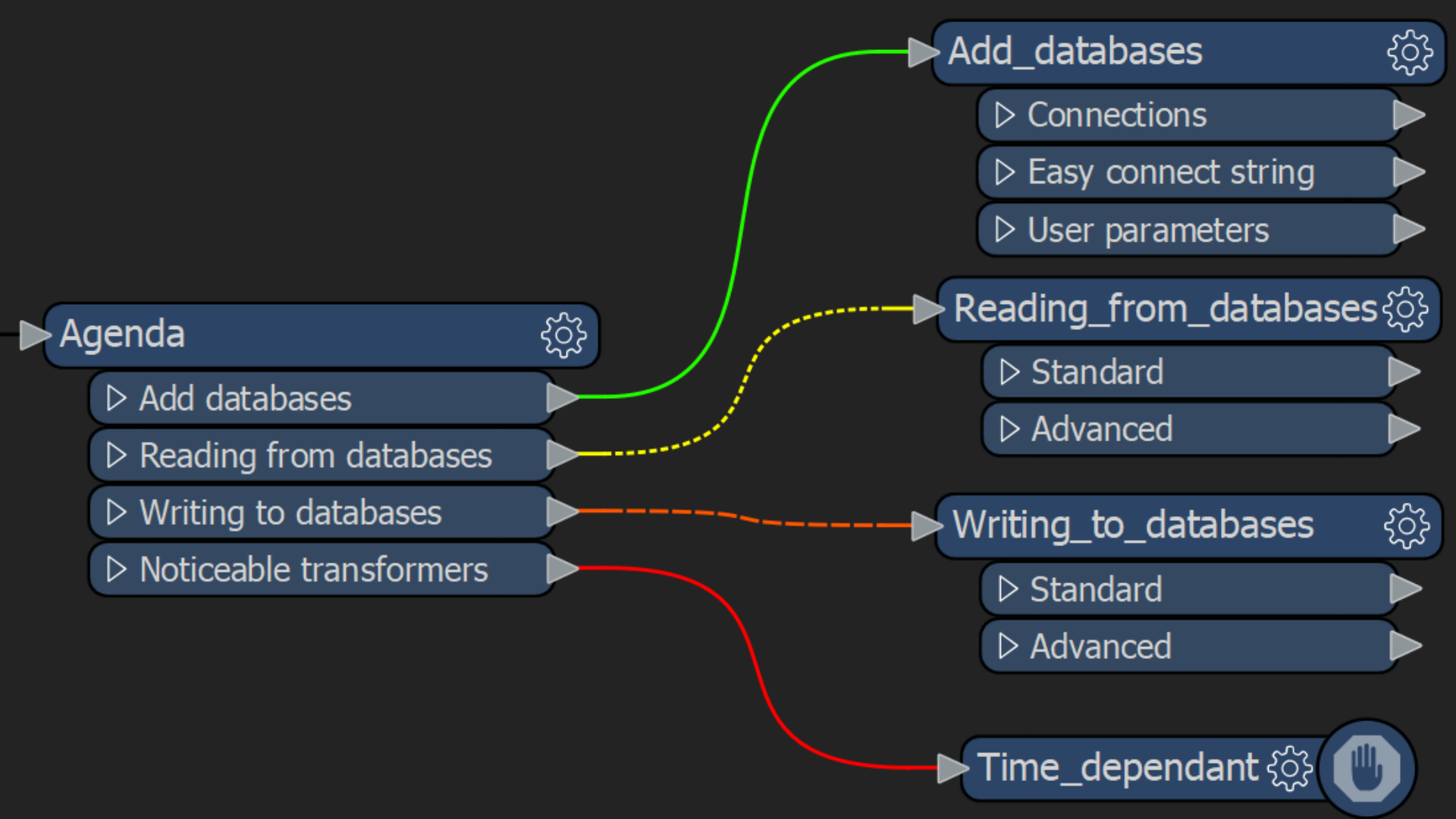
SAFE SOFTWARE | ★★★ PLATINUM PARTNER

SWECO

QUESTIONS

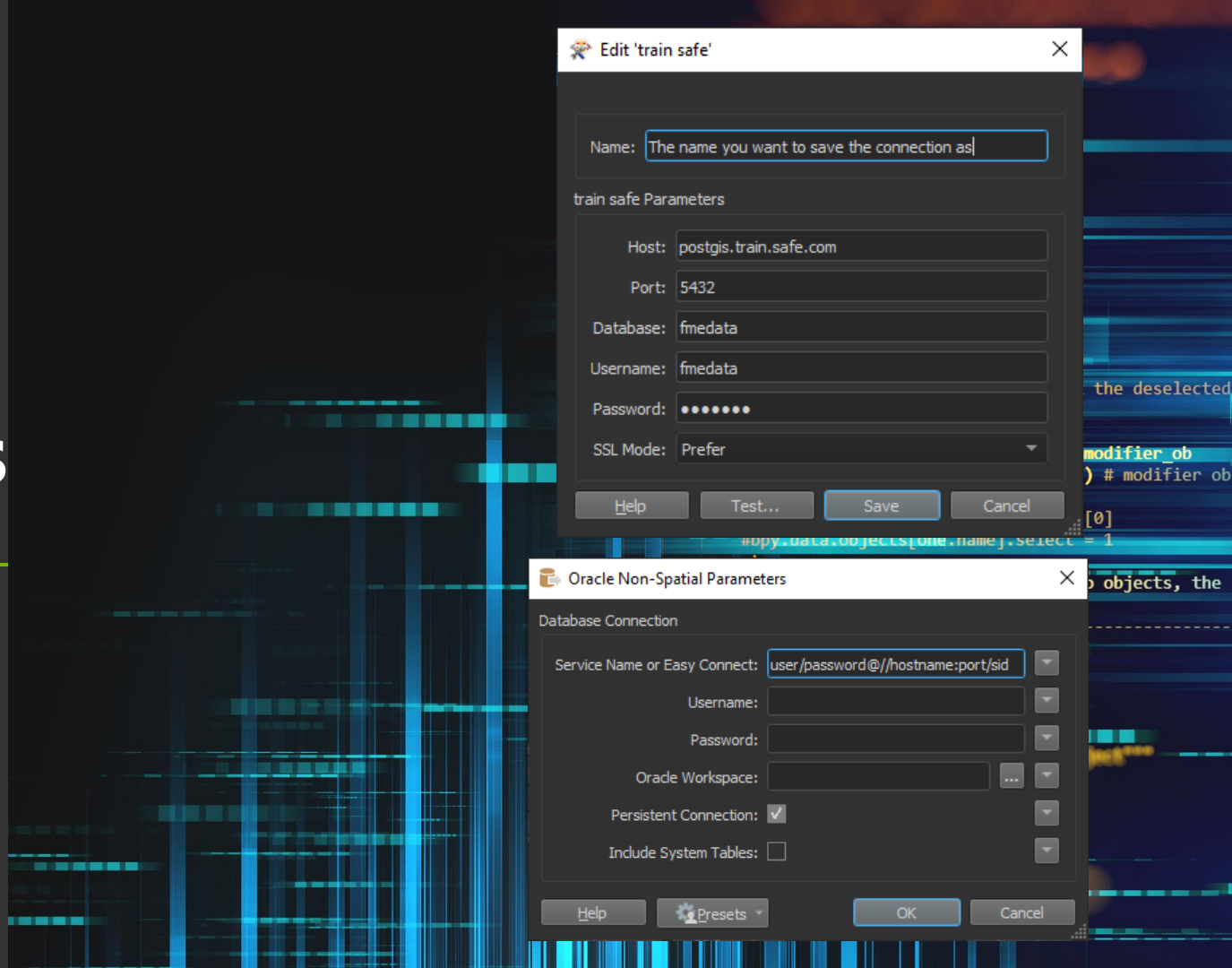
Jurgen
van Tiggelen

Mikael
Månsson


Agenda
Add databases
Reading from databases
Writing to databases
Noticeable transformers
Add_databases
Connections
Easy connect string
User parameters
Reading_from_databases
Standard
Advanced
Writing_to_databases
Standard
Advanced
Time_dependant

POLL x2

Add Databases



Connections
Easy connect string
User parameter

DEMO

Reading from databases

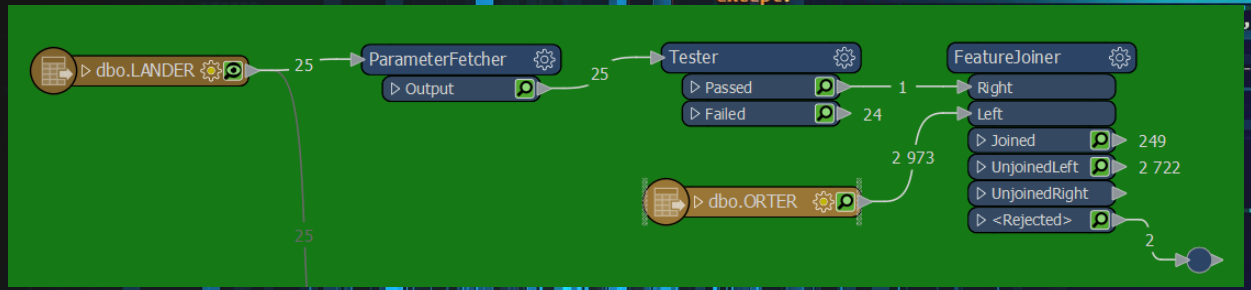
Classic readers.
~~Combinations.~~
~~Where clauses.~~
~~SQL code.~~

Pro’s
- Easy and clear to use.

Con’s
- Slower performances.
- Memory intensive.

```
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
elif _operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif _operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

    #selection at the end -add back the deselected
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob
    #mirror_ob.select = 0
#one = bpy.context.selected_objects[0]
#bpy.data.objects[one.name].select = 1
except:
```

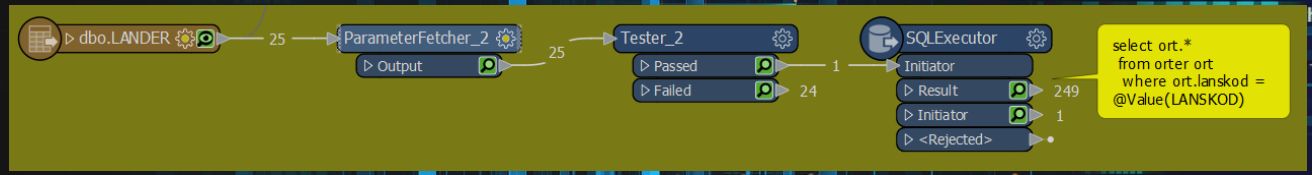


Reading from databases

Pro’s
- Easy and clear to use.
- Reading less data.
- Better performance.

Con’s
- More complicated

*This method can also be done with the DatabaseJoiner which doesn’t require SQL code knowledge



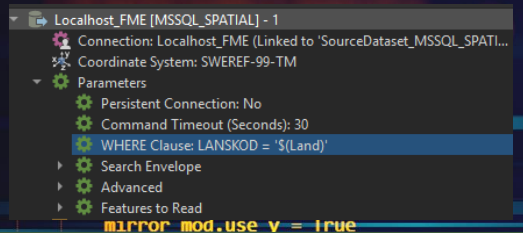
~~Classic readers.~~
Combinations.
~~Where clauses.~~
~~SQL code.~~

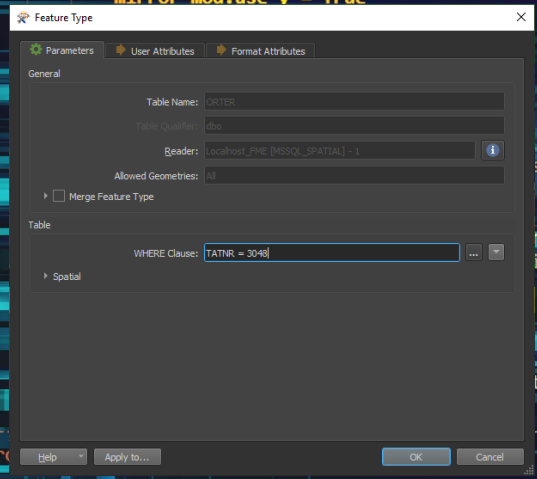
Reading from databases

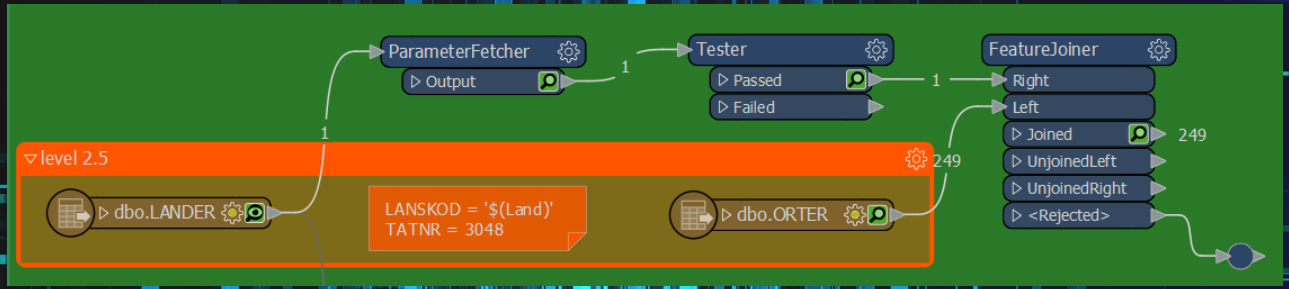
~~Classic readers.~~
~~Combinations.~~
Where clauses.
~~SQL code.~~

Pro’s
- Relative easy to use.
- Reading less data.
- Better performance.

Con’s
- Where clauses are spread over tables and harder to see.





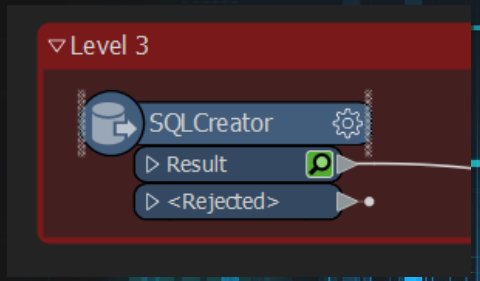


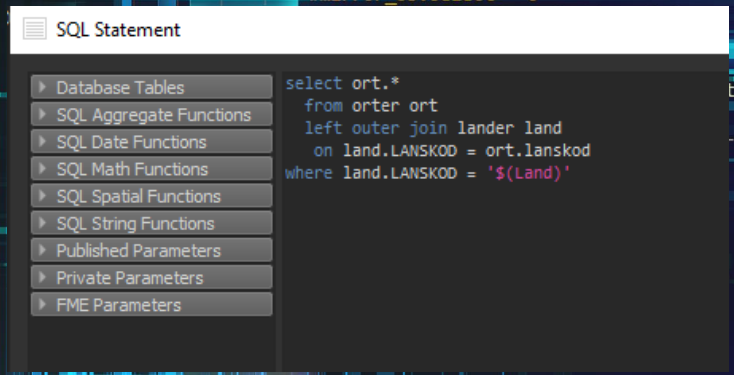
Reading from databases

~~Classic readers.~~
~~Combinations.~~
~~Where clauses~~.
SQL code.

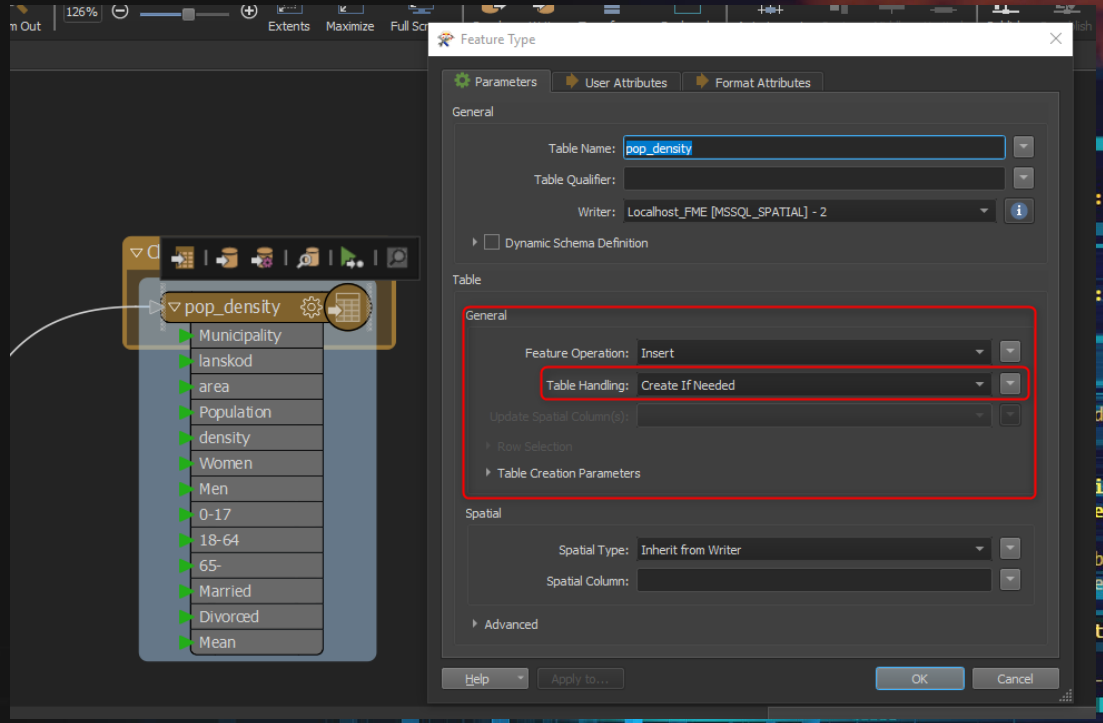
Pro's
- Best performance when query is well formed.
- Best memory usage.

Con's
- Most complicated.
- Requires SQL knowledge.





Writing to databases



Classic writers.
~~Insert Statements.~~
~~Complicated statements.~~

- Always look at ”Table Handling” before writing to a database.
- Use the Import featuretype from dataset function if the table already exists in the database to ensure correct datatypes.

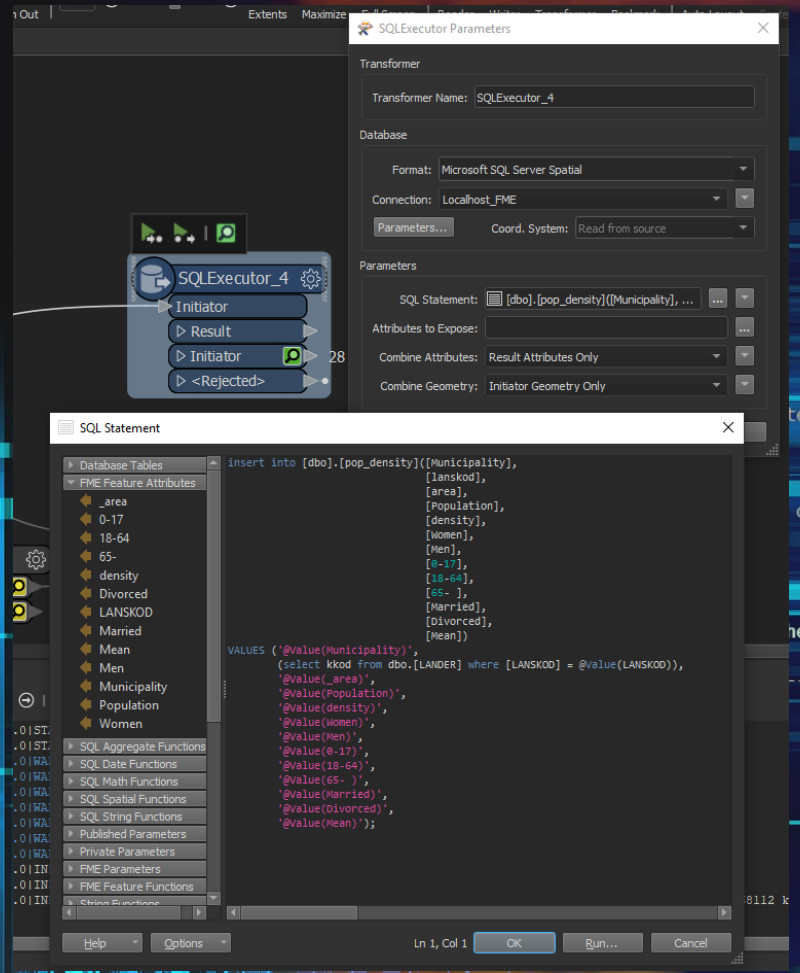
Writing to databases

~~Classic writers.~~

Insert Statements.

~~Complicated statements.~~

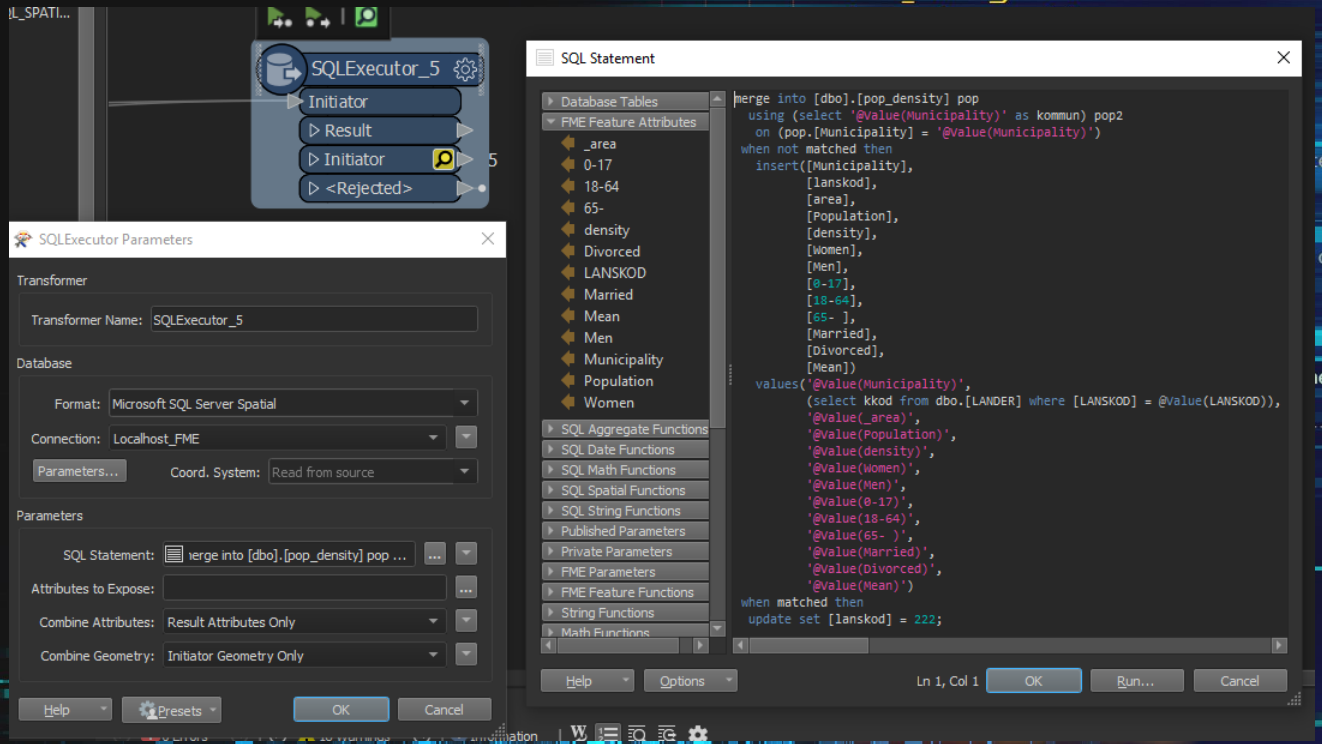
This type of insert is by far harder then a classic writer but it allows you to call functions and sub-select statements for example.



Writing to databases

~~Classic writers.~~
~~Insert Statements.~~
Complicated statements.

The SQLExecutor allows you to use more advanced sql statements.

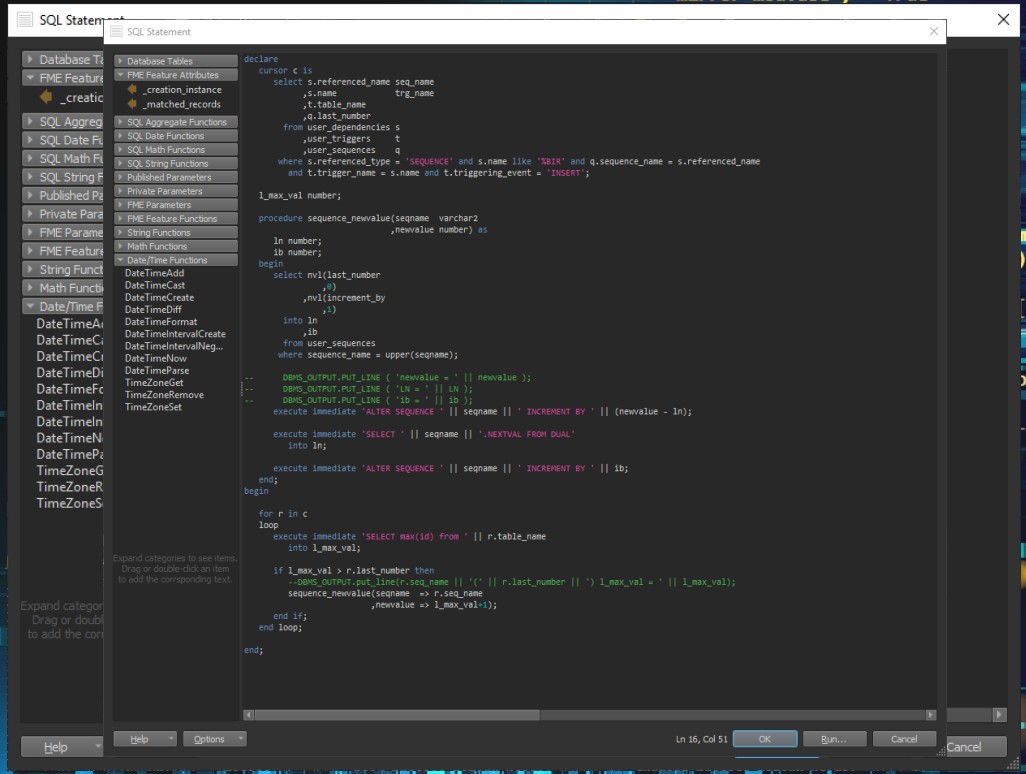


Writing to databases

~~Classic writers.~~
~~Insert Statements.~~
Complicated statements.

The SQLExecutor allows you to use more advanced sql statements and even create tables, functions, etc.

**Make sure that you know what you are doing with these kind of scripts. Talk with your databasemanager!*



DEMO

Online Training

23-25/6 FME Desktop Basic Training

https://dataflow.center/training-events/

Upcoming webinars

17/6 Special guest: Dmitri Bach from Safe Software

Lets´s go gaming with FME!

25/6 FME Server Apps

https://dataflow.center/training-events/

Contact

fmesupport@sweco.se
jurgen.van.tiggelen@sweco.se
mikael.mansson@sweco.se

https://dataflow.center/contact