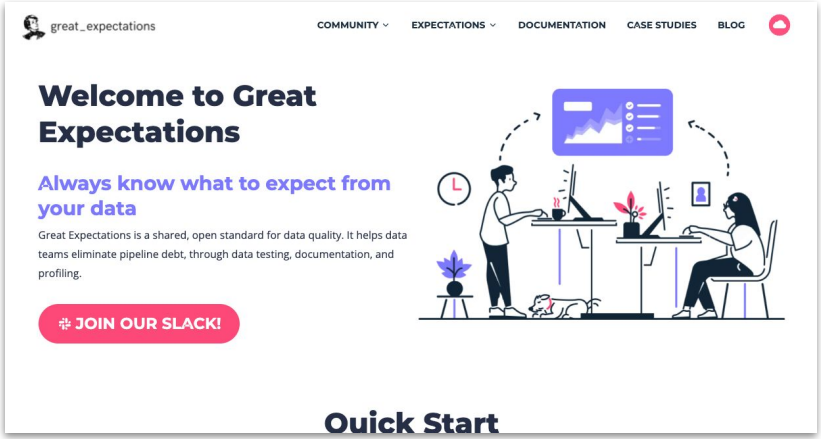
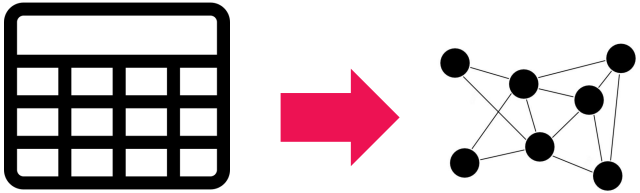
Going Meta #11: Graph data quality with *Graph Expectations*

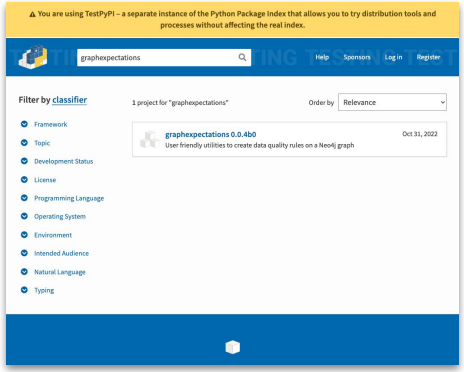
Graph Expectations





graphexpectations package

`https://test.pypi.org/search/?q=graphexpectations`




```
import graphexpectations as ge
import pandas as pd
```

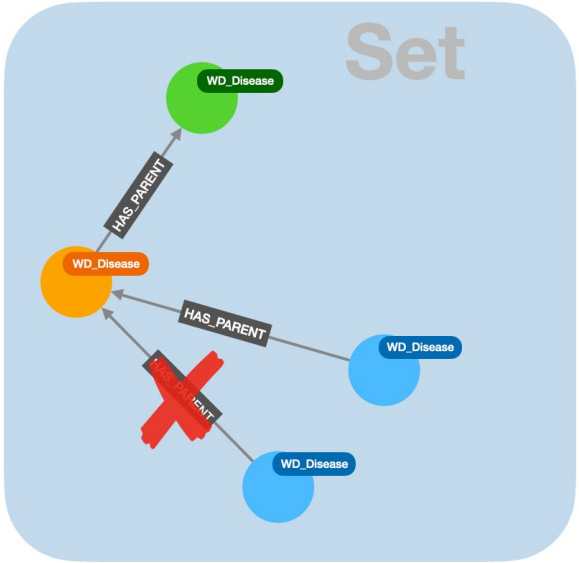
```
companyExpectations = ge.Set(nodeType="Company")
companyExpectations.expect_property_values_to_be_of_type(property="foundingDate", datatype="date")
companyExpectations.expect_number_of_property_values_to_be_between(property="foundingDate", min=1,
companyExpectations.expect_outgoing_relationship_to_connect_to_nodes_of_type(relationship="location
companyExpectations.expect_node_types_to_be_in_list(typeList=["Company"])

personExpectations = ge.Set(nodeType="Person")
personExpectations.expect_node_types_to_be_in_list(typeList=["Person"])
```

```
s = ge.Suite(desc="suite of expectations for my Neo4j Company
s.add_expectations([personExpectations, companyExpectations, ]
```

```
context = s.bind_to_db("bolt://54.243.15.28:7687","neo4j","firefighting-capacitor
```

```
df = pd.DataFrame(context.run())
```

Sets: describing a the shape of a set of nodes



Associated to a node type (Label)

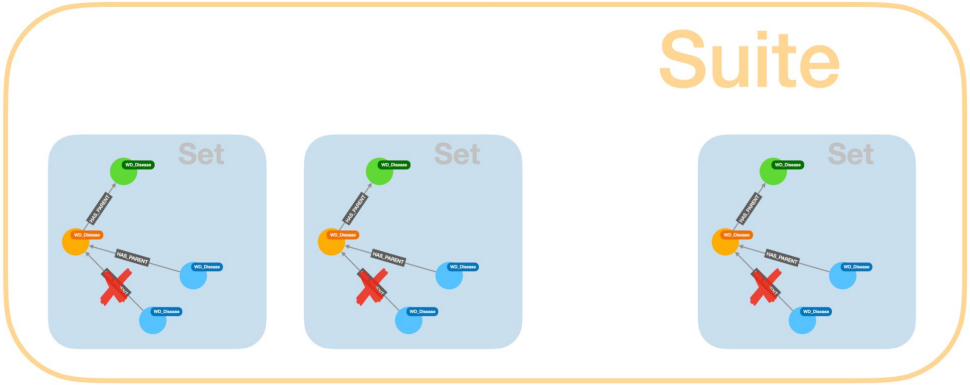
```
companyExpectations = ge.Set(nodeType="Company")
```

Or to a set of nodes defined by an expression (query)

```
locationExpectations = ge.Set(query=
        " (focus)<-[:location|locationCity]-() ")
```

(note for SHACLers: equivalent to a sh:NodeShape)

Suites: a collection of Sets



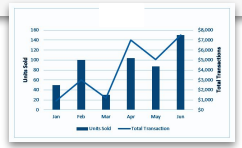
Create and add sets…

```
s = ge.Suite(desc="suite of expectations for my Neo4j Company KG")
s.add_expectations([personExpectations, companyExpectations, locationExpectations])
```

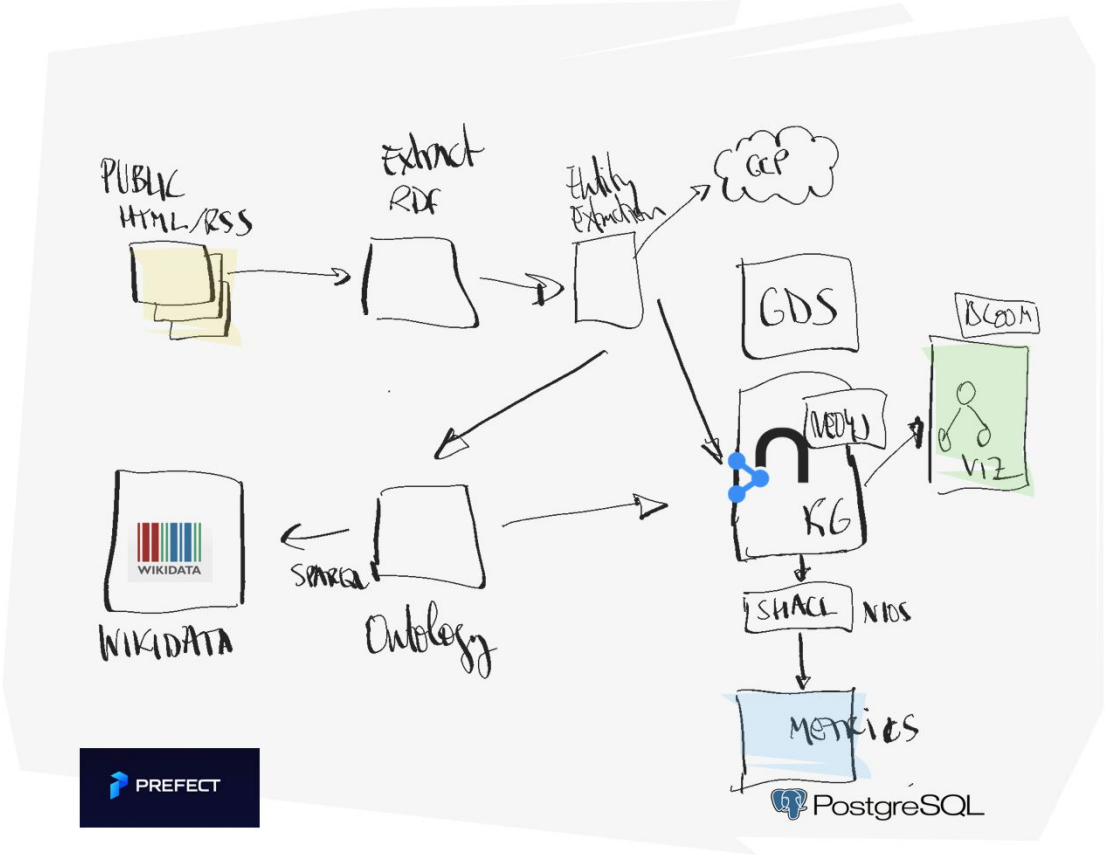
Context: a Suite bound to a DB

Get an execution context by deploying the suite to a DB

```
context = s.bind_to_db("bolt://3.83.175.133:7687","neo4j","pwd")
context.run()
```



Expectations in use…



What next?

- Feedback please! (and contributions)
- Observability = Freshness, Distribution, Volume, Schema, Lineage… so far we have only focused on schema. Looking at distribution/volume.
- Storage/integration of expectation results