



www.datalineo.com

Automate your Testing & Validation in Power BI

Ben Watt

dataMindsConnect
11 October 2022



Ben Watt



At your Service!

- Managing Director at Datalineo
- Microsoft MVP Data Platform
- Power BI, SQL/Azure/Power Platform
- Events & User Group
 - Dublin Power BI UG Leader
 - Data Ceili (eta: Summer 2023)



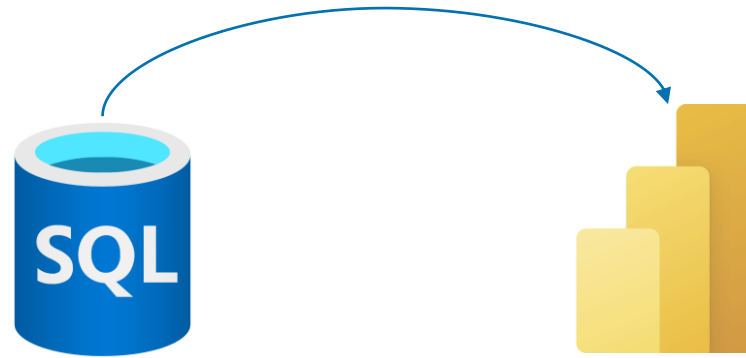
@benrebooted



Agenda

- Intro on the solution: AdventureWorks
 - Power BI dataset
 - Source SQL Database
- The testing process, as a diagram (1 slide)
- Configuration, Code walkthrough, Demo
- Considerations and Limitations (3 slides)

The solution we are testing



Fields

Search

✓ Calculations

- ☐ Customer Count
- ☐ Product Count
- ☐ Promotion Count
- ☐ Sales Amount
- ☐ Sales Line Volume
- ☐ Sales Volume
- ☐ Σ Value

> Calendar

> Customer

> Period Selector

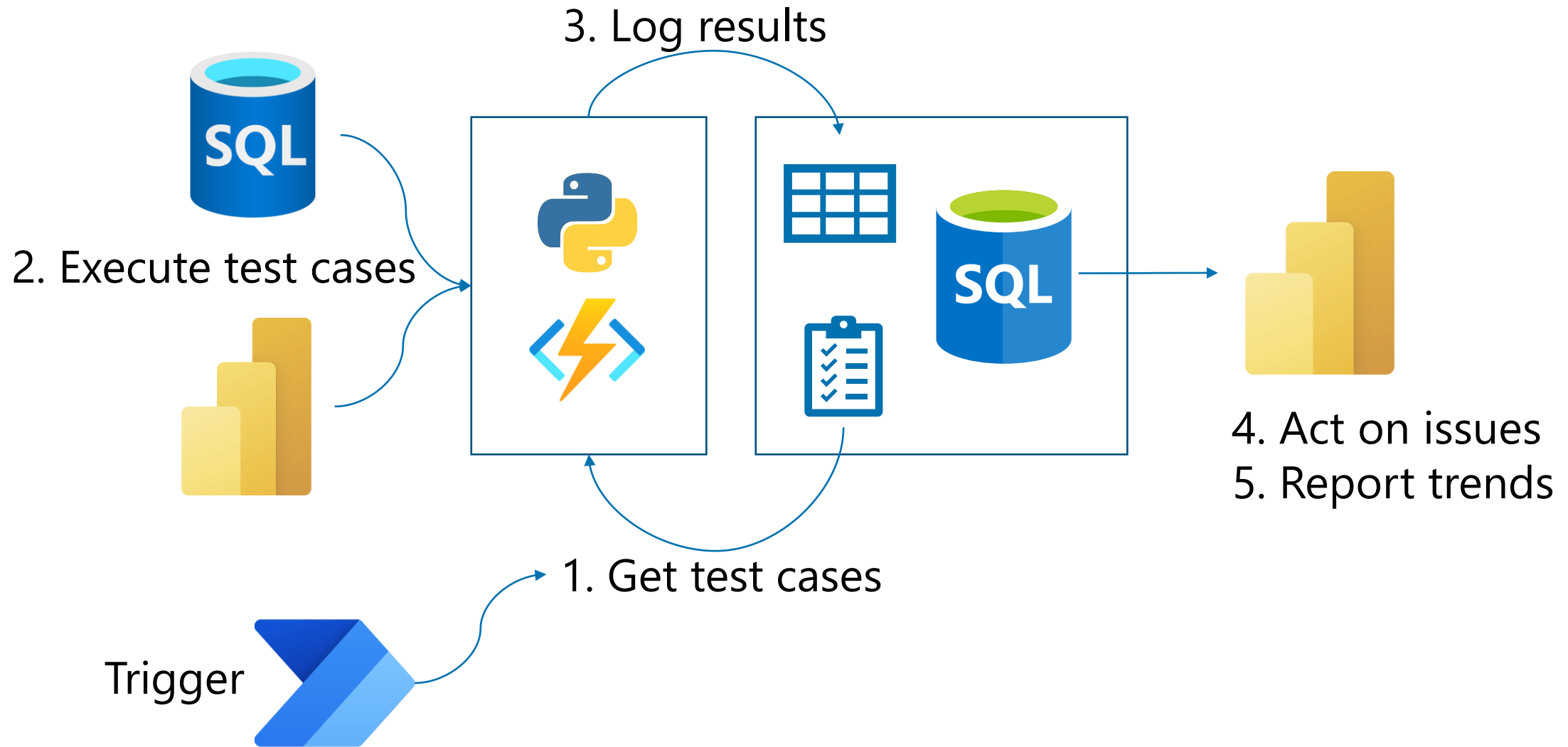
> Product

> Promotion

> Sales

> Sales Territory

The moving parts and the process

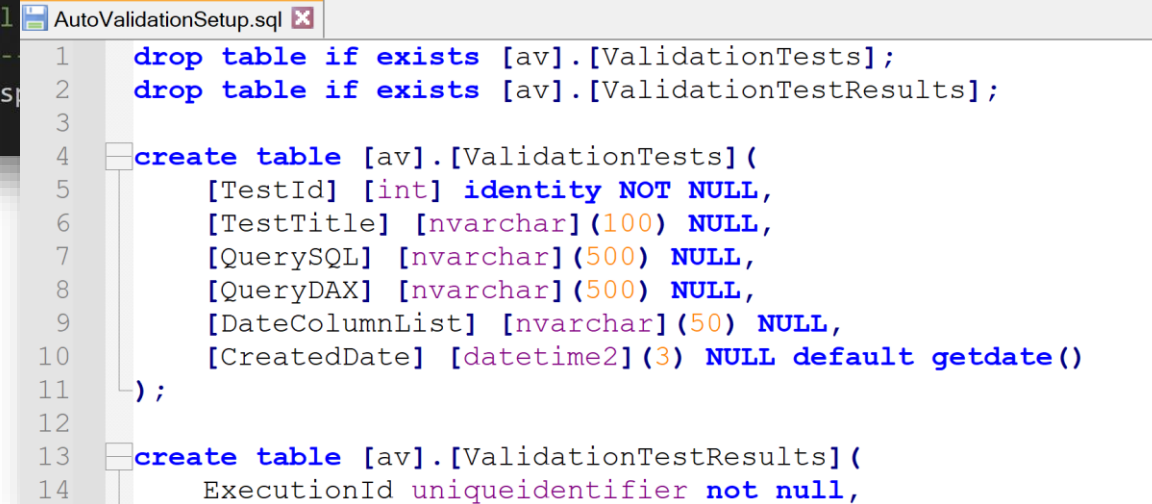


- Configuration Setup
- Code walk through
- Execution Demo

```
for x in test_list:
    test_id = x['TestId']
    query_sql = x['QuerySql']
    query_dax = x['QueryDax']
    #date_column_list_text = x['DateColumnList']
    date_column_list = []
    if x['DateColumnList']: date_column_list = json.loads(x['DateColumnList'])

    df_sql = pd.read_sql_query(sql=query_sql, con=source_engine)
    hash_sql = hash_dataframe(df_sql,date_column_list,1)

# -----
# Call
# -----
dax_res
```



```
1 drop table if exists [av].[ValidationTests];
2 drop table if exists [av].[ValidationTestResults];
3
4 create table [av].[ValidationTests](
5     [TestId] [int] identity NOT NULL,
6     [TestTitle] [nvarchar](100) NULL,
7     [QuerySQL] [nvarchar](500) NULL,
8     [QueryDAX] [nvarchar](500) NULL,
9     [DateColumnList] [nvarchar](50) NULL,
10    [CreatedDate] [datetime2](3) NULL default getdate()
11 );
12
13 create table [av].[ValidationTestResults](
14     ExecutionId uniqueidentifier not null,
```

Approach

- Focus is on raw data, across all necessary datapoints (facts & dims)
 - i.e. if the flour/sugar/butter/eggs are OK, then the cake will be fine!
- The query results have to be re-producible on both sides. Therefore, your SQL query has to synthesize what is in your Data Model, or conversely, your DAX query has to synthesize source data
- If the raw data is fine but validation issues are still popping up, then testing needs to move within the data model (DAX Measures/Columns, Filters, etc)

Limitations

- 100,000 row limit in Power BI executeQueries API
- Version 1 based on comparing Power BI dataset to one SQL source
 - other data sources could be implemented, include connection strings in meta data
- Testing a "live" database against a "import-mode" model may produce differences due to refresh timing.
 - Can be handled with date-sensitive queries (last week, year to date)
- Need to specifically call out date fields, due to type-casting variation loading JSON and SQL results to a dataframe

Comparing results: Python pandas dataframe.compare

- Column header names not important (it overrides to 1,2,3...)
- Everything else is important:
 - Column order
 - Results set order
 - Case sensitivity
 - Number rounding
- Consider higher aggregation queries for better performance
 - Don't compare 2 x 1 million row datasets
 - Aggregate will still surface issues at detailed level

Demo content and slides here:

<https://www.github.com/datalineo/sessions>

Contact



@benrebooted



Please make my day
and leave feedback!