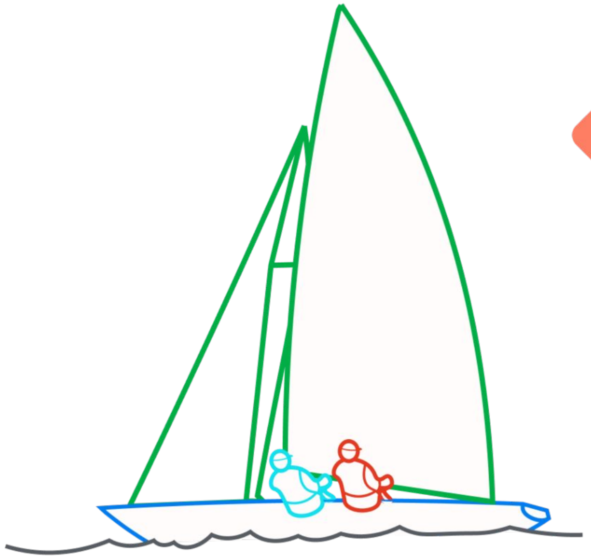
Eat, Sleep, Test, Repeat

Data Quality Testing with Airflow and Soda

Nathan Hadfield - King



Airflow Summit

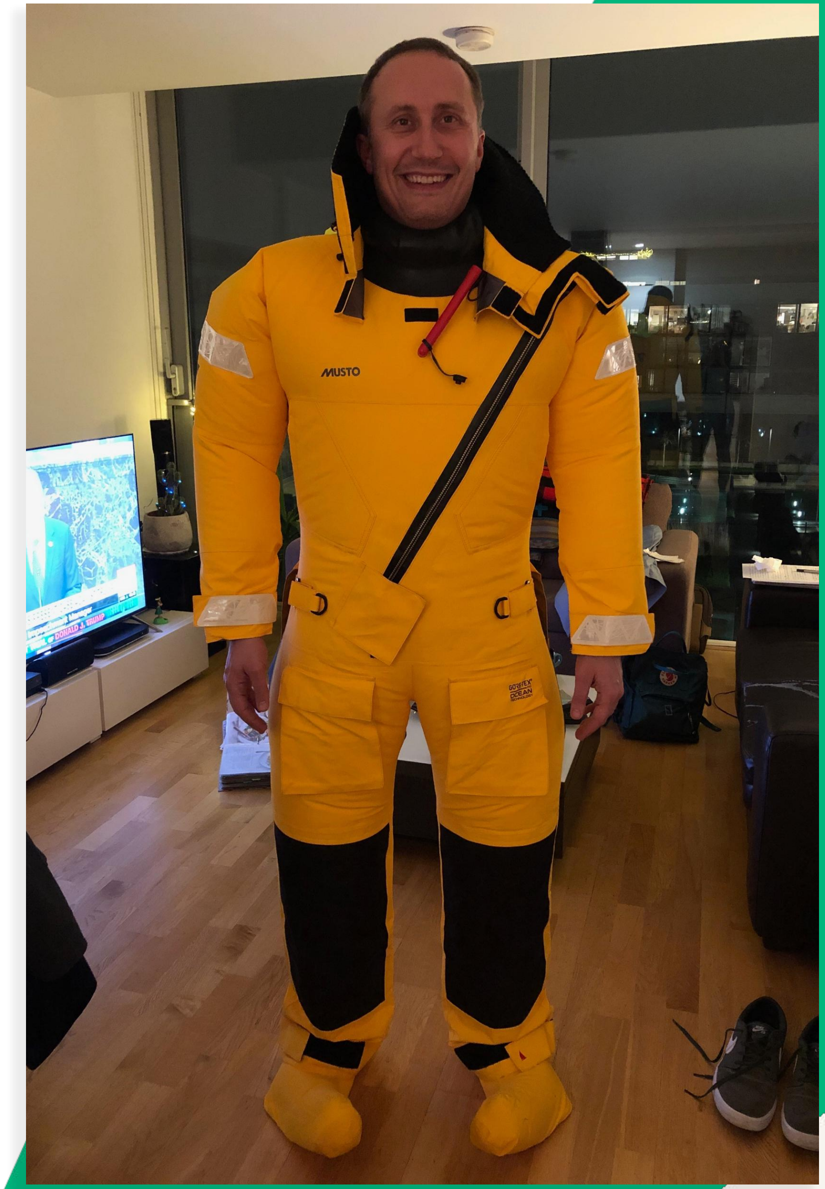
Let's flow together

September 19-21, 2023,
Toronto, Canada

Nathan Hadfield

Principal Data Engineer

EA is a pending trade mark of Electronic Arts, Inc



About King

King Facts

- World leading mobile interactive entertainment company founded in 2003
- Launched Candy Crush Saga on app stores in November 2012
- Downloaded over **3B** times (so far)!
- Played by over **200M MAU**
- Over **14,000 levels!**
- Top-grossing game franchise in the U.S. app stores for the 23rd quarter in a row

2016

ACTIVISION BLIZZARD

King
Making the World Playful

Data @ King

Who doesn't like big numbers?

Truly BIG Data

- Multi-Petabyte scale data warehouse
- Individual tables in excess of 500TB / 1T rows
- >100B events per day

Google Cloud

Data @ King

How do we use it all?

KPIs

Testing

Optimisation

Troubleshooting

Data @ King

Why does it matter?

- Data informed decisions can have a significant impact
- Important that data pipelines are dependable and produce good output
- Data downtime must be kept to a minimum
- Success is partly measured on data SLAs
- Part of the solution is Data Quality Testing

Making the World Playful

Data Quality Testing

What do we mean?

- Testing specific and well-known problems
- Stopping bad data from propagating downstream and reaching data users
- Facilitating investigations
- Tracking dataset health over time

Data Quality Testing

Types of tests

Volume
Did we actually load/create something?

Uniqueness
Are there duplicates?

Reference
Are there unexpected values?

Freshness
How up to date is it?

Validity
Does it conform to expected patterns?

Distribution
How much has the data changed?

Data Quality Testing Tools

A very quick (non-exhaustive) summary

Great Expectations

- ✔ Most established OS tool
- ✔ Large community
- ✔ Lots of pre-defined checks

- ✗ Obtuse terminology
- ✗ Overblown configuration

(Personal opinion, don't hate me)

DBT

- ✔ Testing built in
- ✔ No extra tooling/libraries

- ✗ Not a DBT user/customer

Custom SQL

- ✔ No extra tooling/libraries

- ✗ No standardization
- ✗ Still need to check the result

Soda

www.soda.io

Soda Core

- Free OS Python library and CLI tool

Soda Cloud*

- Data observability web application
- Visualise test results & historical measurements

Soda Library*

- Commercial extension of Soda Core

SodaGPT*

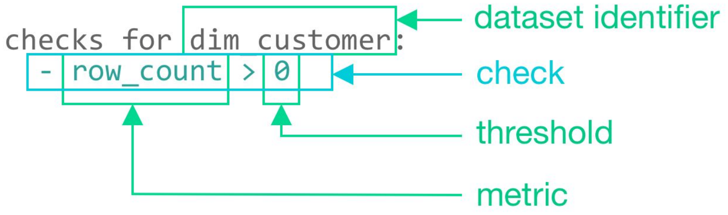
- Generative AI powered tool for DQT
- Translates natural language into data checks

SodaCL (Soda Check Language)

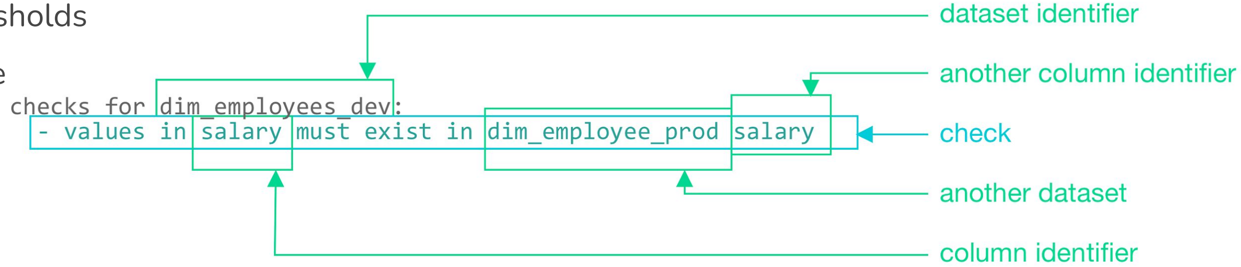
- Provides the foundation for all of the above

SodaCL

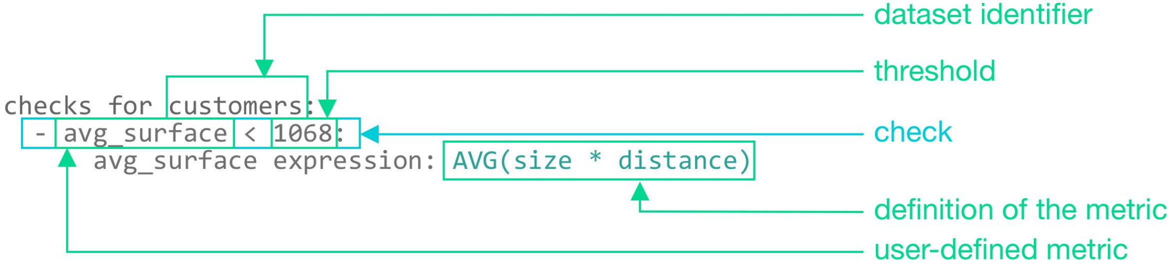
- YAML-based domain specific language for expressing data checks
  - Checks are transportable across different data sources
- Checks are performed by running a scan via Soda Core
  - A single Soda scan can perform multiple checks against one or more datasets
  - Each check results in one of three default states
    - **pass**: the values in the dataset match/fall within the thresholds
    - **fail**: the values in the dataset do not match/fall within the thresholds
    - **error**: the syntax of the check is invalid
  - Currently contains 29 pre-built metrics
- Types of checks:
  - **Standard** – Uses a metric and a threshold
  - **Unique** – Follow unique patterns relevant to the DQ parameters
  - **User-defined** – Uses CTEs or SQL queries



```
checks for dim_customer:
 - row_count > 0
```



```
checks for dim_employees_dev:
 - values in salary must exist in dim_employee_prod salary
```

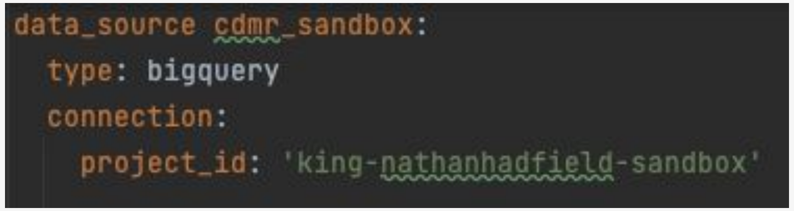


```
checks for customers:
  - avg_surface < 1068:
      avg_surface expression: AVG(size * distance)
```

Running Soda Checks

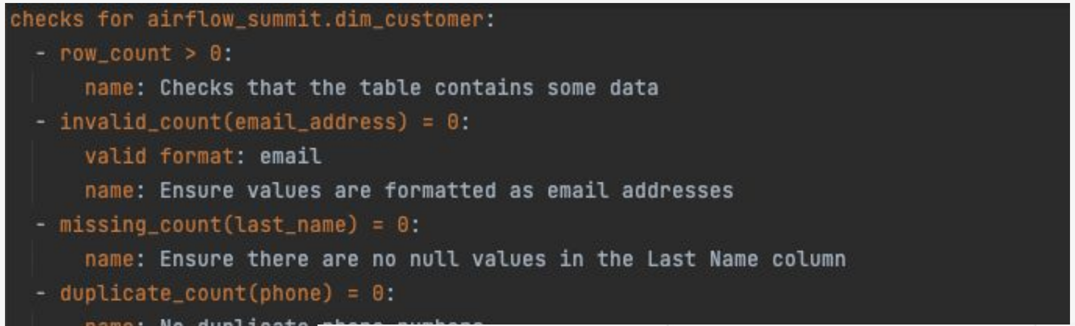
Taking a sip

Configuration file



```
data_source cdmr_sandbox:
  type: bigquery
  connection:
    project_id: 'king-nathanhadfield-sandbox'
```

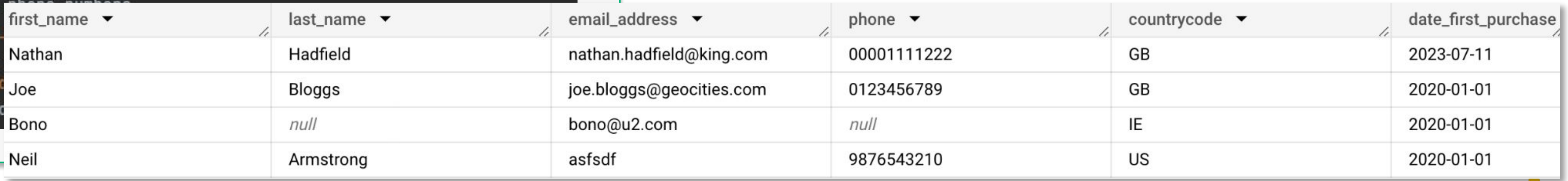
Checks file



```
checks for airflow_summit.dim_customer:
  - row_count > 0:
      name: Checks that the table contains some data
  - invalid_count(email_address) = 0:
      valid format: email
      name: Ensure values are formatted as email addresses
  - missing_count(last_name) = 0:
      name: Ensure there are no null values in the Last Name column
  - duplicate_count(phone) = 0:
      name: No duplicate
  - freshness(date_first_
      name: Data in this
  - values in (countrycod
      name: No invalid co
```

Soda Scan

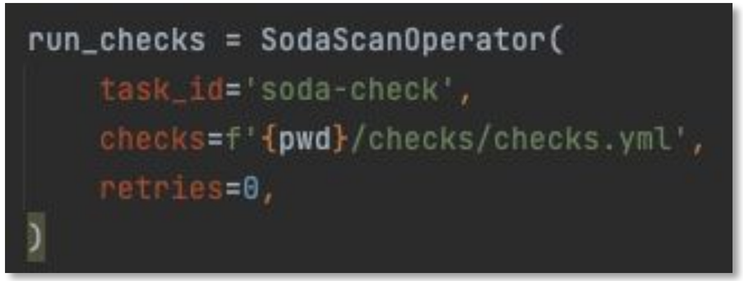
```
✘ □ ~/Documents/airflow_summit □ soda scan -d cdmr_sandbox -c configuration.yml
checks.yml
[10:43:34] Soda Core 3.0.44
[10:43:41] Scan summary:
[10:43:41] 4/6 checks PASSED:
[10:43:41]     airflow_summit.dim_customer in cdmr_sandbox
[10:43:41]       Checks that the table contains some data [PASSED]
[10:43:41]       No invalid country codes [PASSED]
[10:43:41]       No duplicate phone numbers [PASSED]
[10:43:41]       Data in this dataset is less than 7 days old [PASSED]
[10:43:41] 2/6 checks FAILED:
[10:43:41]     airflow_summit.dim_customer in cdmr_sandbox
[10:43:41]       Ensure values are formatted as email addresses [FAILED]
[10:43:41]         check_value: 1
[10:43:41]       Ensure there are no null values in the Last Name column [FAILED]
[10:43:41]         check_value: 1
[10:43:41] Oops! 2 failures. 0 warnings. 0 errors. 4 pass.
```



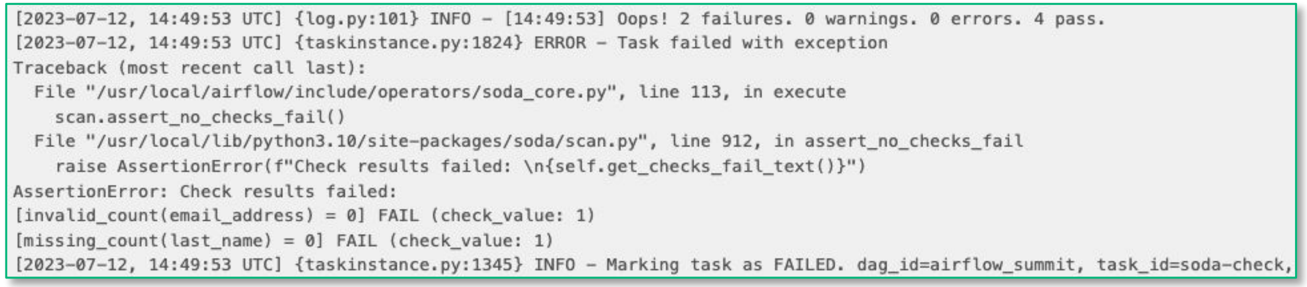
first_name	last_name	email_address	phone	countrycode	date_first_purchase
Nathan	Hadfield	nathan.hadfield@king.com	00001111222	GB	2023-07-11
Joe	Bloggs	joe.bloggs@geocities.com	0123456789	GB	2020-01-01
Bono	*null*	bono@u2.com	*null*	IE	2020-01-01
Neil	Armstrong	asfsdf	9876543210	US	2020-01-01

X SODA

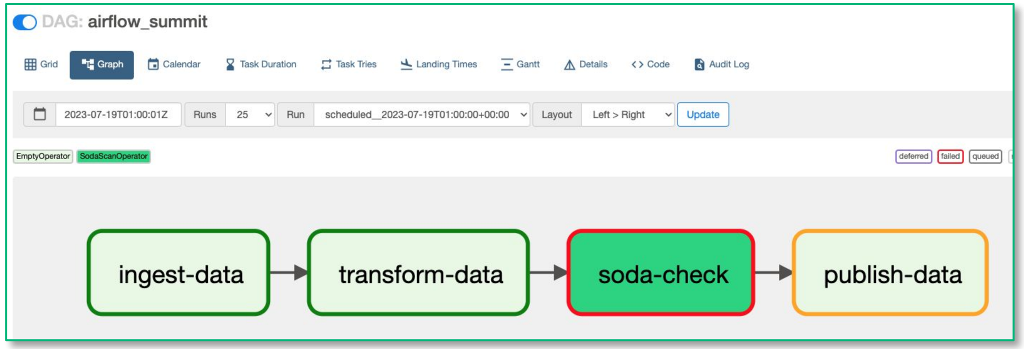
- No Soda provider exists (currently).
  - Example operator code is available on their site
  - Developed a custom **SodaScanOperator**



```
run_checks = SodaScanOperator(
    task_id='soda-check',
    checks=f'{pwd}/checks/checks.yml',
    retries=0,
)
```

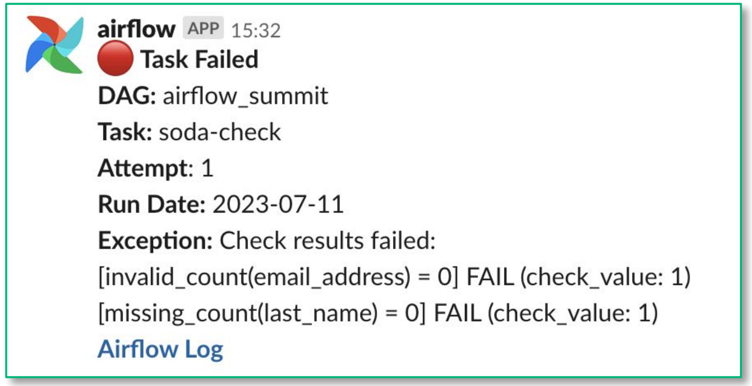


```
[2023-07-12, 14:49:53 UTC] {log.py:101} INFO - [14:49:53] Oops! 2 failures. 0 warnings. 0 errors. 4 pass.
[2023-07-12, 14:49:53 UTC] {taskinstance.py:1824} ERROR - Task failed with exception
Traceback (most recent call last):
  File "/usr/local/airflow/include/operators/soda_core.py", line 113, in execute
    scan.assert_no_checks_fail()
  File "/usr/local/lib/python3.10/site-packages/soda/scan.py", line 912, in assert_no_checks_fail
    raise AssertionError(f"Check results failed: \n{self.get_checks_fail_text()}")
AssertionError: Check results failed:
[invalid_count(email_address) = 0] FAIL (check_value: 1)
[missing_count(last_name) = 0] FAIL (check_value: 1)
[2023-07-12, 14:49:53 UTC] {taskinstance.py:1345} INFO - Marking task as FAILED. dag_id=airflow_summit, task_id=soda-check,
```



DAG: airflow_summit

2023-07-19T01:00:01Z Runs 25 Run scheduled__2023-07-19T01:00:00+00:00 Layout Left > Right



airflow APP 15:32

🔴 Task Failed

DAG: airflow_summit

Task: soda-check

Attempt: 1

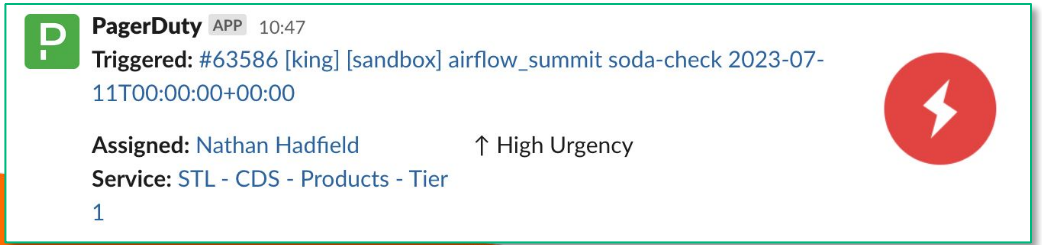
Run Date: 2023-07-11

Exception: Check results failed:

[invalid_count(email_address) = 0] FAIL (check_value: 1)

[missing_count(last_name) = 0] FAIL (check_value: 1)

Airflow Log



PagerDuty APP 10:47

Triggered: #63586 [king] [sandbox] airflow_summit soda-check 2023-07-11T00:00:00+00:00

Assigned: Nathan Hadfield ↑ High Urgency

Service: STL - CDS - Products - Tier 1

Airflow x Soda

SodaScanOperator

set_verbose

- Outputs the check SQL to the log

assert_no_error_logs

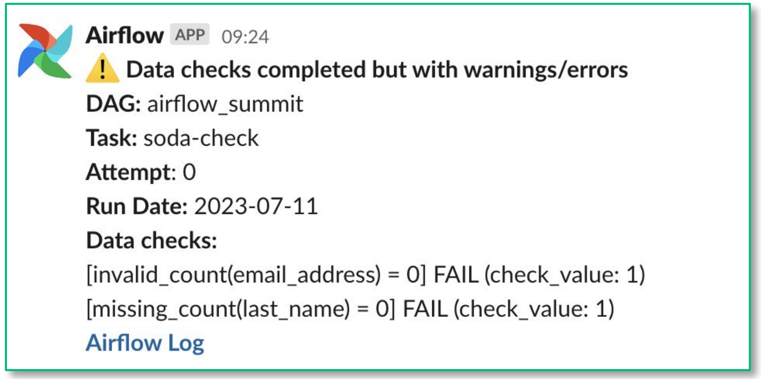
- Checks that there were no SQL errors

assert_no_checks_fail

- Raises an exception if any check failed

has_check_warns_or_fails

- Return a bool if any check fails/warns
- If TRUE, return the check output to an XCOM
- Enables non-critical checks to not cause task failures



Airflow APP 09:24

⚠ **Data checks completed but with warnings/errors**

DAG: airflow_summit

Task: soda-check

Attempt: 0

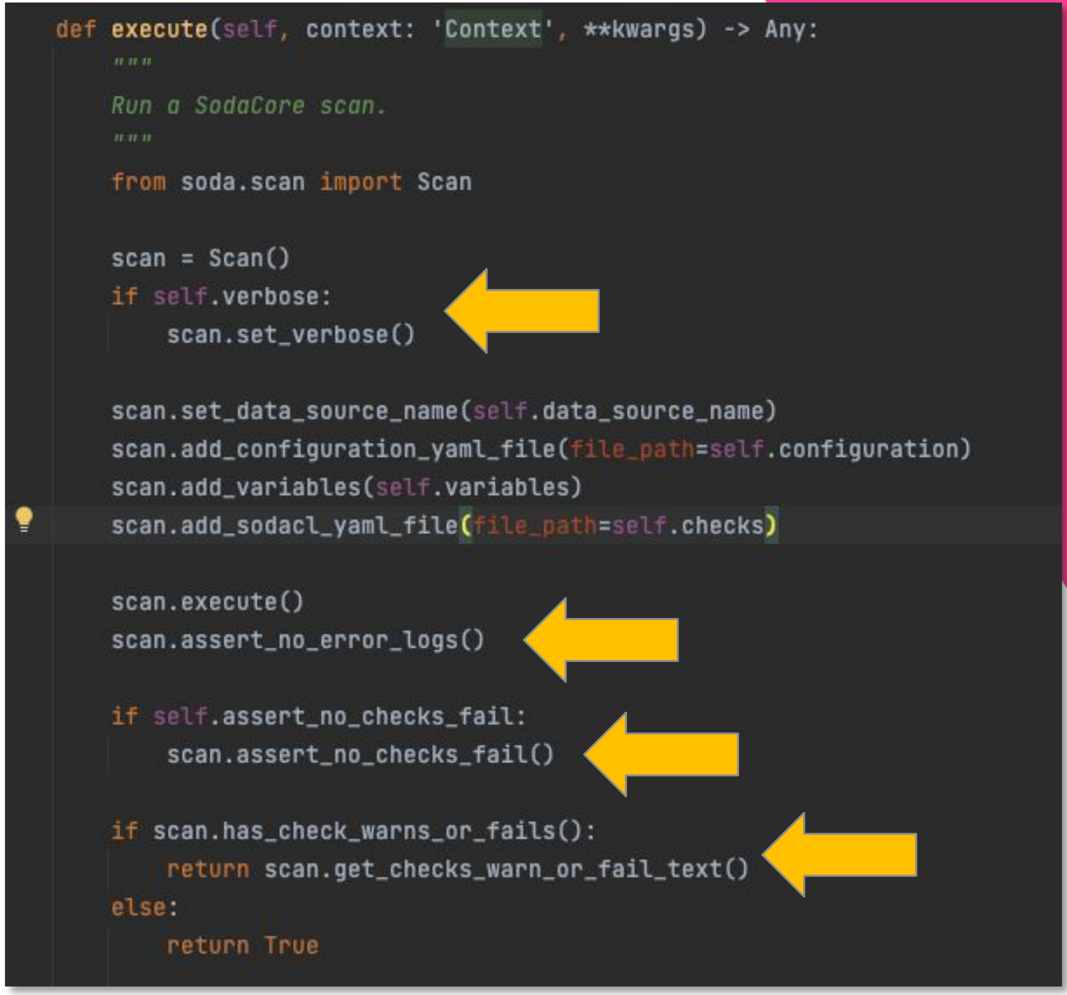
Run Date: 2023-07-11

Data checks:

[invalid_count(email_address) = 0] FAIL (check_value: 1)

[missing_count(last_name) = 0] FAIL (check_value: 1)

Airflow Log

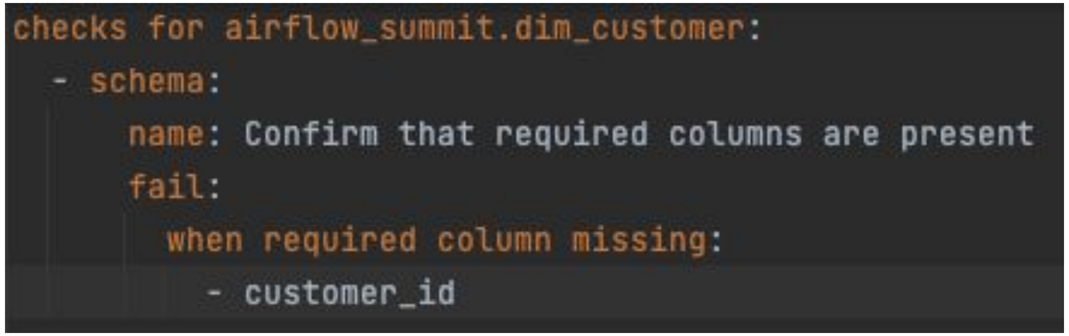


```python
def execute(self, context: 'Context', **kwargs) -> Any:
    """
    Run a SodaCore scan.
    """
    from soda.scan import Scan

    scan = Scan()
    if self.verbose:
        scan.set_verbose()

    scan.set_data_source_name(self.data_source_name)
    scan.add_configuration_yaml_file(file_path=self.configuration)
    scan.add_variables(self.variables)
    scan.add_sodacl_yaml_file(file_path=self.checks)

    scan.execute()
    scan.assert_no_error_logs()

    if self.assert_no_checks_fail:
        scan.assert_no_checks_fail()

    if scan.has_check_warns_or_fails():
        return scan.get_checks_warn_or_fail_text()
    else:
        return True
```

Other Soda Capabilities

Schema checks

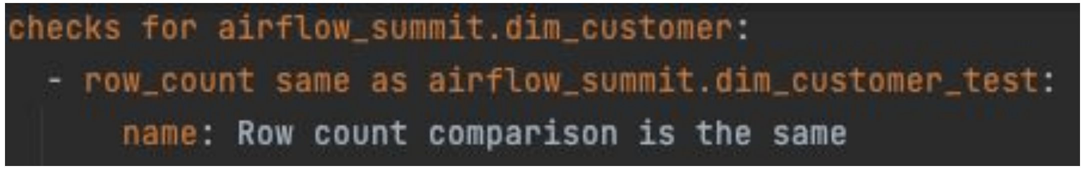
- Validate the presence, absence, position or type of columns
- Employ alert configurations to specify fail conditions



```
checks for airflow_summit.dim_customer:
  - schema:
      name: Confirm that required columns are present
      fail:
        when required column missing:
          - customer_id
```

Cross checks

- Compare row counts between datasets within the same, or different, data sources

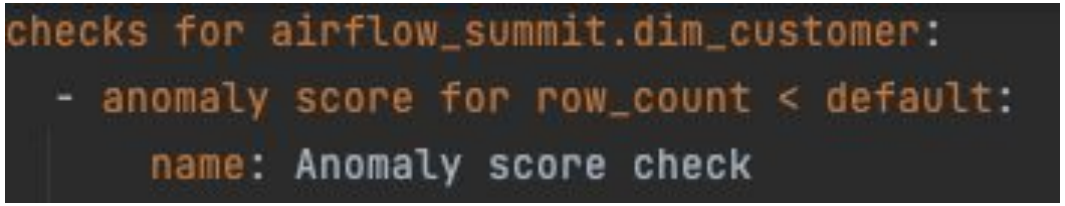


```
checks for airflow_summit.dim_customer:
  - row_count same as airflow_summit.dim_customer_test:
      name: Row count comparison is the same
```

Other Soda Capabilities

Anomaly score

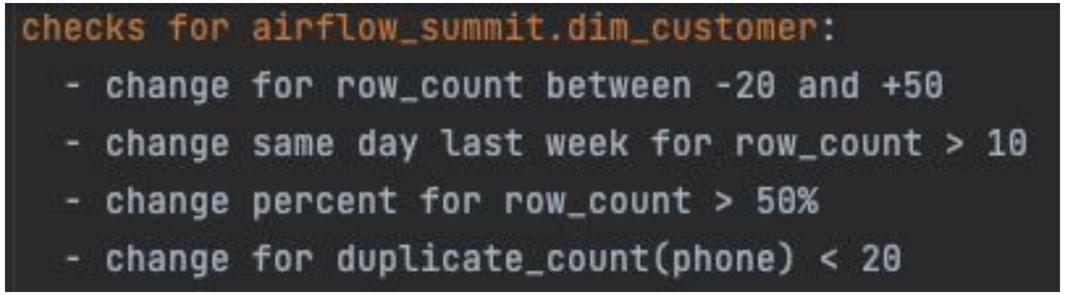
- Machine learning algorithm that detects anomalies based on learned patterns
- Identified and flags anomalies in time series data
- Can use numeric, missing and validity metrics
- **Requires Soda Cloud**



```
checks for airflow_summit.dim_customer:
  - anomaly score for row_count < default:
      name: Anomaly score check
```

Change over time thresholds

- Compares metrics relative to a previously measured value
- **Requires Soda Cloud**



```
checks for airflow_summit.dim_customer:
  - change for row_count between -20 and +50
  - change same day last week for row_count > 10
  - change percent for row_count > 50%
  - change for duplicate_count(phone) < 20
```

Summary

- Data Quality Testing is something you **should** be doing
- Soda provides an easy to configure, data source agnostic and human-readable way of defining common types of data checks
- Integrating Soda with Airflow enables data engineers to perform tests at any point in a pipeline
- Combining with other Airflow integrations (Slack, PagerDuty) accelerates discovery and reduces downtime

- More advanced Soda capabilities are behind their commercial products
- Writing checks requires domain knowledge and knowing what to check for
- DQT is just part of a multi-layered observability strategy
  - Automated testing/monitoring
  - Automated root-cause analysis
  - Data lineage

Thank you!

King

Making the World Playful

Questions?

- Nathan Hadfield
  - nathan.hadfield@king.com
  - https://www.linkedin.com/in/nathanhadfield/
  - https://github.com/nathadfield

- Careers @ King
  - https://careers.king.com/