Data At Rest
Granular Quality in Flowing Pipelines

Mauricio De Diana
C.J. Jameson
2023-09-20
Data Quality: The Data itself, and how it flows

- Data itself
  - Field-level metrics: null rate, uniqueness, etc
  - Freshness and volume
  - Schema

- Data flow / processing
  - DAGs and tasks
  - dbt pipelines
  - Databricks jobs
Phases of a data incident

- Prevention
- Detection
- Resolution
Airflow operators

Detection/Prevention - field level

- PythonOperator with an error
- SQLColumnCheckOperator: evaluates fields in isolation
- SQLTableCheckOperator: allows evaluation of multiple fields
- SQLCheckOperator: evaluates single row returned
- ShortCircuitOperator: skips but do not stop/alert

Skips silently, email on failure

Show code example / DAG image, talk about detection vs prevention
Great Expectations / dbt tests

Detection/Prevention, field-level

- More descriptive
- Integrates with Slack, Email
- TODO: Can be used as CB?

Show GE snippet example

Show dbt DAG or snippet or output
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Details</th>
<th>Status</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Monte Carlo monitors</strong></td>
<td>Detection/Resolution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Field Health</strong></td>
<td>[Snowflake] demo_env:staging.zuora_invoice Field: invoice_amount</td>
<td><a href="mailto:epost+demo@montecarlodata.com">epost+demo@montecarlodata.com</a></td>
<td>20 hours ago / in a day</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Segmented by: status</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Volume Rule</strong></td>
<td>Ensure that the number of new offers does not dip below the rolling 7 day average</td>
<td>[Snowflake] demo_env:raw:offer</td>
<td>a day ago / in an hour</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="mailto:mcdemo@montecarlodata.com">mcdemo@montecarlodata.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SQL Rule</strong></td>
<td>SQL rule to ensure all records from SFDC Opportunity get loaded downstream</td>
<td>[Snowflake] demo_env:raw:salesforce_opportunity,</td>
<td>6 days ago / in 19 hours</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[Snowflake] demo_env:reporting.d_opportunity,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[Snowflake] demo_env:staging.salesforce_opportunity</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Variables:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>agg (2 values)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>field (1 values)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>table_1 (2 values)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>table_2 (2 values)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>category (3 values)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="mailto:mcdemo@montecarlodata.com">mcdemo@montecarlodata.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Field Health</strong></td>
<td>[Snowflake] demo_env:raw:salesforce_account</td>
<td><a href="mailto:mcdemo@montecarlodata.com">mcdemo@montecarlodata.com</a></td>
<td>Triggered dynamically</td>
<td>0</td>
</tr>
<tr>
<td><strong>Freshness Rule</strong></td>
<td>Test to ensure that plan data is being updated throughout the day</td>
<td>[Snowflake] demo_env:raw:plan</td>
<td>2 days ago / in 7 hours</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="mailto:mcdemo@montecarlodata.com">mcdemo@montecarlodata.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Dimension Tracking</strong></td>
<td>[Snowflake] demo_env:raw:subscription Field: status</td>
<td><a href="mailto:mcdemo@montecarlodata.com">mcdemo@montecarlodata.com</a></td>
<td>10 hours ago / in 2 hours</td>
<td>0</td>
</tr>
</tbody>
</table>
Monte Carlo monitors Detection/Resolution

Viewing field quality rule: % zero > 0% for invoice_quantity field in [Snowflake] demo_env:raw.zuora_invoice table

<table>
<thead>
<tr>
<th>Monitor Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule type: Scheduled</td>
</tr>
<tr>
<td>Test interval: This monitor runs every 12 hour(s) starting at May 12th 2023 14:47 GMT-3</td>
</tr>
<tr>
<td>Status:</td>
</tr>
<tr>
<td>Run history: Monitor ran successfully at Sep 13th 2023 02:47 GMT-3</td>
</tr>
<tr>
<td>Model training status: Fully trained</td>
</tr>
<tr>
<td>Misconfigurations: None</td>
</tr>
<tr>
<td>Created at: May 12th 2023 14:35 GMT-3</td>
</tr>
<tr>
<td>Created by: <a href="mailto:eposh-demo@montecarlodata.com">eposh-demo@montecarlodata.com</a></td>
</tr>
<tr>
<td>Last modified at: May 12th 2023 14:35 GMT-3</td>
</tr>
<tr>
<td>Last modified by: <a href="mailto:eposh-demo@montecarlodata.com">eposh-demo@montecarlodata.com</a></td>
</tr>
<tr>
<td>Incidents (7d): 2</td>
</tr>
</tbody>
</table>

Incident Trend (30d)
Monte Carlo monitors Detection/Resolution

Field Quality Rule run results

<table>
<thead>
<tr>
<th>Date</th>
<th>Result</th>
<th>Rule Breach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sep 12th 2023 02:47 GMT-3</td>
<td>0.06%</td>
<td>View breach</td>
</tr>
<tr>
<td>Sep 12th 2023 02:47 GMT-3</td>
<td>0%</td>
<td>No breach</td>
</tr>
<tr>
<td>Sep 12th 2023 14:48 GMT-3</td>
<td>0.08%</td>
<td>View breach</td>
</tr>
</tbody>
</table>

Summary

**[Snowflake] SQL rule breached: Circuit Breaker Condition**

Opened Wednesday Sep 13th 2023 10:00 GMT-3 (Now)

**Summary**

- **Breached rows**
  - Breached Rows Returned
    - BREAKER_RECORD
      - 2023-02-09 17:41:17.433 Z

Breached rows are stored in the data collector. If you would like to disable such collection, please contact your customer success manager.
Monte Carlo circuit breaker Detection/Prevention

https://pypi.org/project/airflow-mcd/
Lineage

Resolution - Data at rest + Data flows

- Which tables are affected by a data quality problem upstream?
- Is there something in common among apparently unrelated data issues?
- Are there reports affected downstream by this data issue? Which ones?
- Which job(s) populates this table?
OpenLineage

- Marquez
- Astronomer

Show Marquez screenshot
**Airflow failures Resolution**

**Airflow DAG failure: subscription_load, Task: load_incremental, Exception: 002003 (02000): 01ae060a-0603...**

Opened: Tuesday Aug 1st 2023 16:28 GMT-3 (1 month, 12 days ago)

---

### Summary

**DAG**
- subscription_load

**Task**
- load_incremental

**Logs**
- View logs

**Owner**
- mcdemo@montecarlold...

**Severity**
- No severity

**Jira issues**
- CDI-23

**Notification Channels**
- montecarlo.alationcatalog.com
- https://montecarlodatainc.webhook.office.com...
- OpsGenie
- private
- private
- private
- https://hooks.zapier.com***frk/

---

**002003 (02000): 01ae05d0-0603-a63c-0010-a8830cbbcc2b2: SQL compilation error: Task 'DEMO_ENV.RAW.SUBSCRIPTION_LOAD' does not exist or not authorized.**

Start: Aug 01, 2023 16:27:55 GMT-3 | Execution: Aug 01, 2023 16:27:54 GMT-3 | End: Aug 01, 2023 16:28:06 GMT-3 | Duration: 00:00:11 | Skipped Tasks: -

---

**002003 (02000): 01ae060a-0603-a63b-0010-a8830cbd726e: SQL compilation error: Task 'DEMO_ENV.RAW.SUBSCRIPTION_LOAD' does not exist or not authorized.**

Start: Aug 01, 2023 17:26:18 GMT-3 | Execution: Aug 01, 2023 17:26:18 GMT-3 | End: Aug 01, 2023 17:26:23 GMT-3 | Duration: 00:00:04 | Skipped Tasks: -
### Airflow failures Resolution

A screenshot showing the Airflow dashboard with a table of runs. The table includes columns for `DAG Run ID`, `DAG`, `Task`, `State`, `Start`, `End`, and `Duration`. The data shows various runs that have been completed successfully.

Example entries in the table:
- **DAG Run ID**: scheduled_2023-06-05T16:58:09.519221+00:00
- **DAG**: internal_bi
- **Task**: internal-bi-dbft-models
- **State**: Success
- **Start**: 2023-06-05 04:19:44 PDT
- **End**: 2023-06-05 04:20:14 PDT
- **Duration**: 00:00:30

The image also includes a header indicating freshness anomalies in Snowflake analytics, opened on Wednesday May 31st 2023 at 04:20 PDT (6 days, 9 hours ago).
### Airflow failures Resolution

<table>
<thead>
<tr>
<th>Summary</th>
<th>General Information</th>
<th>Past Incidents (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>DAG</strong></td>
<td><strong>Last run</strong></td>
</tr>
<tr>
<td></td>
<td>subscription_load</td>
<td>Sep 12, 2023 05:00:00 GMT-3</td>
</tr>
</tbody>
</table>

### Runtime

<table>
<thead>
<tr>
<th>Status</th>
<th>All</th>
</tr>
</thead>
</table>

### Graph

- **Run ID:** manual__2023-08-01T20:26:18.463943+00:00
- **Start time:** Aug 01, 2023 17:26:18 GMT-3
- **Duration:** 00:00:04
Airflow failures  Resolution

# explicit, per callback type

dag = DAG(
    'dag_name',
    on_success_callback=mcd_callbacks.mcd_dag_success_callback,
    on_failure_callback=mcd_callbacks.mcd_dag_failure_callback,
    sla_miss_callback=mcd_callbacks.mcd_sla_miss_callback,
)

# explicit, per callback type

task = BashOperator(
    task_id='task_name',
    bash_command='command',
    dag=dag,
    on_execute_callback=mcd_callbacks.mcd_task_execute_callback,
    on_success_callback=mcd_callbacks.mcd_task_success_callback,
    on_failure_callback=mcd_callbacks.mcd_task_failure_callback,
    on_retry_callback=mcd_callbacks.mcd_task_retry_callback,
)

# broad, all callbacks

dag = DAG(
    'dag_name',
    **mcd_callbacks.dag_callbacks,
)

# broad, all callbacks

task = BashOperator(
    task_id='task_name',
    bash_command='command',
    dag=dag,
    **mcd_callbacks.task_callbacks,
)
Questions?

mdediana@montecarloidata.com
@cjcjameson on Twitter
www.montecarlodata.com
What do you do with DAG/Task failures?

- do you get an email? A slack? Some other notification?

- is it one data engineer? The whole team? An on-call rotation?

- which DAGs are configured to alert to which channels? Some are more important than others; some are owned by different teams; who will respond?
Monte Carlo: Airflow failures with tons of context
Managing a legacy, failing Airflow setup
You can review Airflow executions with Databand
Airflow-over-Databricks: review runtime duration