Circuit Breakers to safeguard for Garbage in, Garbage out

Sandeep Uttamchandani
Chief Data Architect & Head of Data Platform Engineering, Small Business & Self Employed Group
Intuit
Real?
OR
Data quality issue?
Who we serve

Consumers

Small Businesses

Self-Employed
Our mission

Powering Prosperity Around the World
Data Pipeline: Physical View

Data Sources
- User Entered Data
- Clickstream
- CRM
- Social

Insights
- In-product
- Business Operations
Data Pipeline: Physical View

Data Sources
- User Entered Data
- Clickstream
- CRM
- Social

Collect
- kafka
- ORACLE GoldenGate

Store ↔ Analyze
- Amazon S3
- HIVE
- Apache Spark
- VERTICA
- cassandra
- druid

Serve

Integrations:
- In-product
- Business Operations
Data Pipeline: Logical View
Data Pipeline: Logical View Example

Lineage created for etl for 2018-9-5 with a depth of 4
Key Reasons for Data Quality Issues

Data Source Issues
- Table inconsistencies
  a. Illegitimate values
  b. Missing values
  c. Duplicate Primary keys
- Hard deletes
- Bulk inserts
- Missing updates to CDC column

Data Ingestion Issues
- Uncoordinated upstream changes
  a. Volume of data
  b. Change in schema
  c. Change in meaning of data
  d. Upgrade of platform
- No CDC for large tables leading to delayed availability
- Errors in ETL logic
- Timezone inconsistencies
- Duplicate or null records due to ingestion errors

Referential Integrity Issues
- Data elements have different data types and/or meaning in different sources
- Inconsistent data element enums
- Heuristic ID Correlation
- Uncoordinated schema changes
- Dropped updates across data sources
Circuit Breakers - Avoiding Electric Fires

BEFORE

AFTER
Circuit Breakers - Avoiding Services with High Response Time

BEFORE

Response Time

AFTER

Response Time

Certain services unavailable
Circuit Breaker - Avoiding Insights with Data Quality Issues

BEFORE

Time-to-Reliable-Insights
UNBOUNDED

AFTER

Team alerted to diagnose & backfill (if possible)

Time-to-Reliable-Insights
BOUNDED
### Circuit Breaker - Avoiding Insights with Data Quality Issues

<table>
<thead>
<tr>
<th>Answering the question:</th>
<th>BEFORE</th>
<th>AFTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is it Real?</td>
<td>Hours/Days/Weeks</td>
<td>Minutes</td>
</tr>
<tr>
<td>OR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a Data quality issue?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time-to-Reliable-Insights</th>
<th>BEFORE</th>
<th>AFTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNBOUNDED</td>
<td></td>
<td>BOUNDED</td>
</tr>
</tbody>
</table>
Implementing Circuit Breakers for Data Pipelines
Implementing Circuit Breakers for Data Pipelines

1. Track Lineage
2. Profile Pipeline
3. Control the Circuit
1. Track Lineage

Data Pipeline

Table
Table
Table
Job
Table
Table
Job
Table/Dashboard
1. Track Lineage
1. Track Lineage

Data Pipeline

\[<\text{Job}, \text{Input, Output}>\]

\[<\text{Job}, \text{Input, Output}>\]

\[<\text{Job}, \text{Input, Output}>\]

\[<\text{Job}, \text{Input, Output}>\]
2. Profile Pipeline

**Operational Profiling**

- Job Health
- Data Fabric Health

*Platform Engineers*

**Data Profiling**

- Single Column
- Multi Column
- Cross-DB Dependencies

*Data Engineers*
## 2. Profile Pipeline - Job Health Example

### Execution Stats

<table>
<thead>
<tr>
<th>Run ID</th>
<th>Name</th>
<th>Status</th>
<th>Schedule Date</th>
<th>Start Time</th>
<th>End Time</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>29005648</td>
<td></td>
<td>Scheduled</td>
<td>2018-09-06</td>
<td>12:00:00.0</td>
<td></td>
<td><code>.../jobrunuser/subgoals/subgoal_stable_current/jobrun/...</code></td>
</tr>
<tr>
<td>28970397</td>
<td></td>
<td>Completed</td>
<td>2018-09-05</td>
<td>12:00:00.0</td>
<td>11:27:36</td>
<td><code>.../jobrunuser/subgoals/subgoal_stable_current/jobrun/...</code></td>
</tr>
<tr>
<td>28940343</td>
<td></td>
<td>Completed</td>
<td>2018-09-04</td>
<td>12:00:00.0</td>
<td>08:34:30</td>
<td><code>.../jobrunuser/subgoals/subgoal_stable_current/jobrun/...</code></td>
</tr>
<tr>
<td>28909375</td>
<td></td>
<td>Completed</td>
<td>2018-09-03</td>
<td>12:00:00.0</td>
<td>03:37:14</td>
<td><code>.../jobrunuser/subgoals/subgoal_stable_current/jobrun/...</code></td>
</tr>
<tr>
<td>28877859</td>
<td></td>
<td>Completed</td>
<td>2018-09-02</td>
<td>12:00:00.0</td>
<td>09:22:21</td>
<td><code>.../jobrunuser/subgoals/subgoal_stable_current/jobrun/...</code></td>
</tr>
<tr>
<td>28848111</td>
<td></td>
<td>Completed</td>
<td>2018-09-01</td>
<td>12:00:00.0</td>
<td>08:20:12</td>
<td><code>.../jobrunuser/subgoals/subgoal_stable_current/jobrun/...</code></td>
</tr>
</tbody>
</table>
2. Profile Pipeline - Data Fabric Health

Data Sources
- User Entered Data
- Clickstream
- CRM
- Social

Collect
- kafka

Store ↔ Analyze
- Amazon S3
- HIVE
- ORACLE GoldenGate
- VERTICA
- Spark

Serve
- cassandra
- druid

Insights
- In-product
- Business Operations

Monitoring & Logging API
2. Profile Pipeline - Data Profiling

- Cardinalities
- Patterns & Data types
- Value distributions
- Domain classification

- Correlations
- Association rules
- Clustering
- Outliers
- Summaries & sketches

- Unique column combinations
- Inclusion dependencies
- Functional dependencies

# 2. Profile Pipeline - Data Profiling

## Single Column

- Cardinalities
- Patterns & Data types
- Value distributions
- Domain classification

<table>
<thead>
<tr>
<th>Category</th>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cardinalities</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>num-rows</td>
<td>Number of rows</td>
</tr>
<tr>
<td></td>
<td>value length</td>
<td>Measurements of value lengths (minimum, maximum, median, and average)</td>
</tr>
<tr>
<td></td>
<td>null values</td>
<td>Number or percentage of null values</td>
</tr>
<tr>
<td></td>
<td>distinct</td>
<td>Number of distinct values; sometimes called “cardinality”</td>
</tr>
<tr>
<td></td>
<td>uniqueness</td>
<td>Number of distinct values divided by the number of rows</td>
</tr>
<tr>
<td>Value distributions</td>
<td>histogram</td>
<td>Frequency histograms (equi-width, equi-depth, etc.)</td>
</tr>
<tr>
<td></td>
<td>constancy</td>
<td>Frequency of most frequent value divided by number of rows</td>
</tr>
<tr>
<td></td>
<td>quartiles</td>
<td>Three points that divide the (numeric) values into four equal groups</td>
</tr>
<tr>
<td></td>
<td>first digit</td>
<td>Distribution of first digit in numeric values; to check Benford’s law</td>
</tr>
<tr>
<td>Patterns, data types, and domains</td>
<td>basic type</td>
<td>Generic data type, such as numeric, alphabetic, alphanumeric, date, time</td>
</tr>
<tr>
<td></td>
<td>data type size</td>
<td>Concrete DBMS-specific data type, such as varchar, timestamp, etc.</td>
</tr>
<tr>
<td></td>
<td>decimals</td>
<td>Maximum number of digits in numeric values</td>
</tr>
<tr>
<td></td>
<td>patterns</td>
<td>Maximum number of decimals in numeric values</td>
</tr>
<tr>
<td></td>
<td>data class domain</td>
<td>Histogram of value patterns (Aa9...)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Semantic, generic data type, such as code, indicator, text, date/time, quantity, identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Classification of semantic domain, such as credit card, first name, city, phenotype</td>
</tr>
</tbody>
</table>

2. Profile Pipeline - Data Profiling Example

- Cardinalities
- Patterns & Data types
- Value distributions
- Domain classification
3. Control the Circuit

Detecting Issues

Absolute Threshold Rules

If current_state >= threshold_value
Then <Alert>

History-based Anomaly Detection

If current_state <differs> history_state
Then <Alert>

Example

<table>
<thead>
<tr>
<th>Operational Profiling</th>
<th>If job_retries &gt; 3, then fail job</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Profiling</td>
<td>If col_val &gt; 2σ, then alert</td>
</tr>
</tbody>
</table>
3. Control the Circuit

<table>
<thead>
<tr>
<th>Soft Alerts (Low Confidence)</th>
<th>Hard Alerts (High Confidence)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational Profiling</td>
<td></td>
</tr>
<tr>
<td>Data Profiling</td>
<td></td>
</tr>
</tbody>
</table>
Summary: Trade-off Between Data Quality & Availability

- **Operational Profiling**
  - Soft Events
  - Hard Events

- **Data Profiling**
  - Soft Events
  - Hard Events

- **Time-to-Reliable-Insights**: UNBOUNDED

- **Data Availability**
- **Data Quality**
Summary: Trade-off Between Data Quality & Availability

- **Data Quality**
- **Data Availability**

**Operational Profiling**
- **Soft Events**
- **Hard Events**

**Data Profiling**
- **Soft Events**
- **Hard Events**

**Time-to-Reliable-Insights**
- BOUNDED

**Availability ~ Quality**

**Data Availability**

- **Data Quality**
The Rockstar Team driving this work!

We are hiring! Come join us!

sandeep_uttamchandani@intuit.com