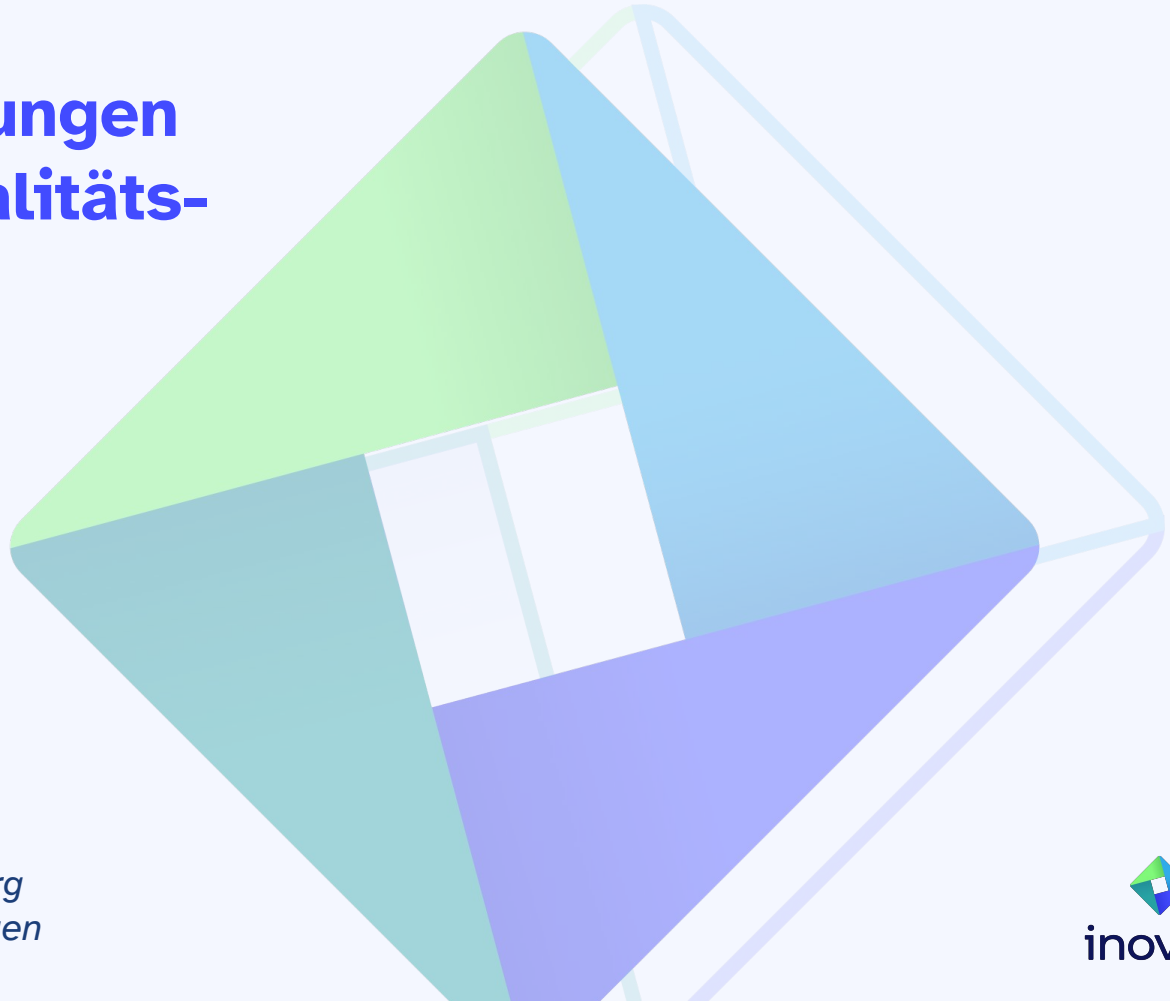
Praxisnahe Erfahrungen aus dem Datenqualitäts-Dschungel

Florian Gräbe, Marcel Spitzer



Team inovex
Karlsruhe · Köln · München · Hamburg
Berlin · Stuttgart · Pforzheim · Erlangen

inovex

Marcel Spitzer

Marcel Spitzer

Data & ML Engineer (2016)

“Struggles his way through the data wilderness”

Florian Gräbe

Florian Gräbe

Data & ML Engineer (2022)

“Driven by good Data, ML and Dancing”

inovex

inovex

Wir freuen uns auf Ihren Besuch an **Stand 2** im Foyer!

WE ARE INOVEX – INNOVATE. INTEGRATE. EXCEED.

Ihr Partner für die digitale Transformation mit dem Leistungsspektrum

- **Data & AI**
- **Application Development**
- **Skalierbare IT-Infrastrukturen**
- **Training & Coaching**

Ausschließlich festangestellte Mitarbeitende

Wachstum durch Innovationen und erfolgreiche Projekte

inovex

inovex.de

SOFTWARE · DATA & AI · INFRASTRUCTURE

Begeben wir uns in die Wildnis

1. Sind wir im Daten-Dschungel?
2. Risiken und Gefahren
3. Sicheres Fortbewegen
4. Survival Ausrüstung

inovex

Sind wir im Daten-Dschungel?

Verwachsen

Undurchsichtig

Unterschiedliche Blickwinkel

Chaotisch / Wild

inovex

Sind wir im Daten-Dschungel?

Verwachsen

Undurchsichtig

Unterschiedliche Blickwinkel

Welche Regeln und Gesetze gelten (GDPR, ...)?

Woher kommen die Daten?

Chaotisch / Wild

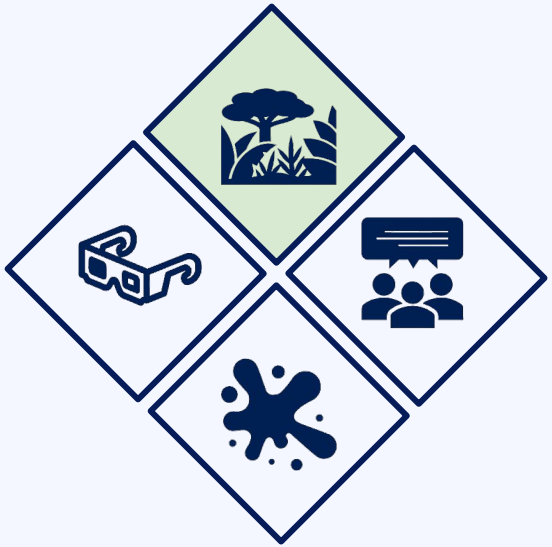
inovex

Sind wir im Daten-Dschungel?

Verwachsen

Wozu sind die Daten nutzbar?

Kann ich den Daten vertrauen?



Undurchsichtig

Unterschiedliche Blickwinkel

Welche Regeln und Gesetze gelten (GDPR, ...)?

Woher kommen die Daten?

Chaotisch / Wild

inovex

Sind wir im Daten-Dschungel?

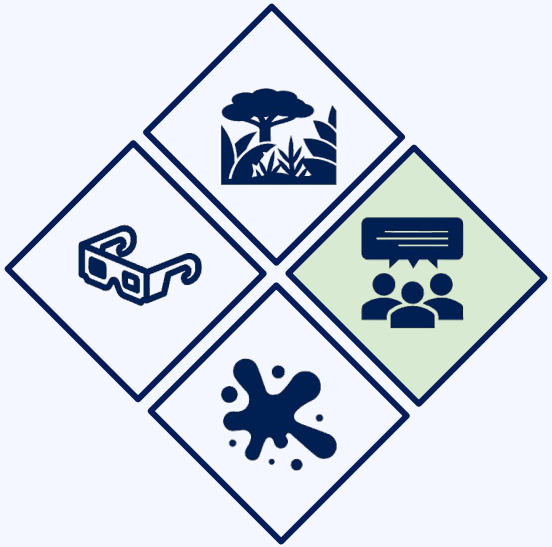
Verwachsen

Wozu sind die Daten nutzbar?

Wie kann ich auf die Daten zugreifen?

Kann ich den Daten vertrauen?

Wer besitzt welche Daten?



Undurchsichtig

Unterschiedliche Blickwinkel

Welche Regeln und Gesetze gelten (GDPR, ...)?

Woher kommen die Daten?

Chaotisch / Wild

inovex

Sind wir im Daten-Dschungel?

Wozu sind die Daten nutzbar?

Verwachsen

Wie kann ich auf die Daten zugreifen?

Kann ich den Daten vertrauen?

Wer besitzt welche Daten?

Undurchsichtig

Unterschiedliche Blickwinkel

Welche Regeln und Gesetze gelten (GDPR, ...)?

In welcher Qualität und Formaten liegen die Daten vor?

Woher kommen die Daten?

Chaotisch / Wild

inovex

Was verstehen wir unter Datenqualität?

- Allgemein: der Grad der **Übereinstimmung von Daten mit der Realität**
- in der Praxis jedoch oft schwierig bis unmöglich auf diese Weise zu messen
- Deshalb: **Abweichung von zuvor festgelegten Annahmen**
- Annahmen ergeben sich aus der jeweiligen **Domäne**, deshalb regelmäßiger Austausch mit Experten erforderlich

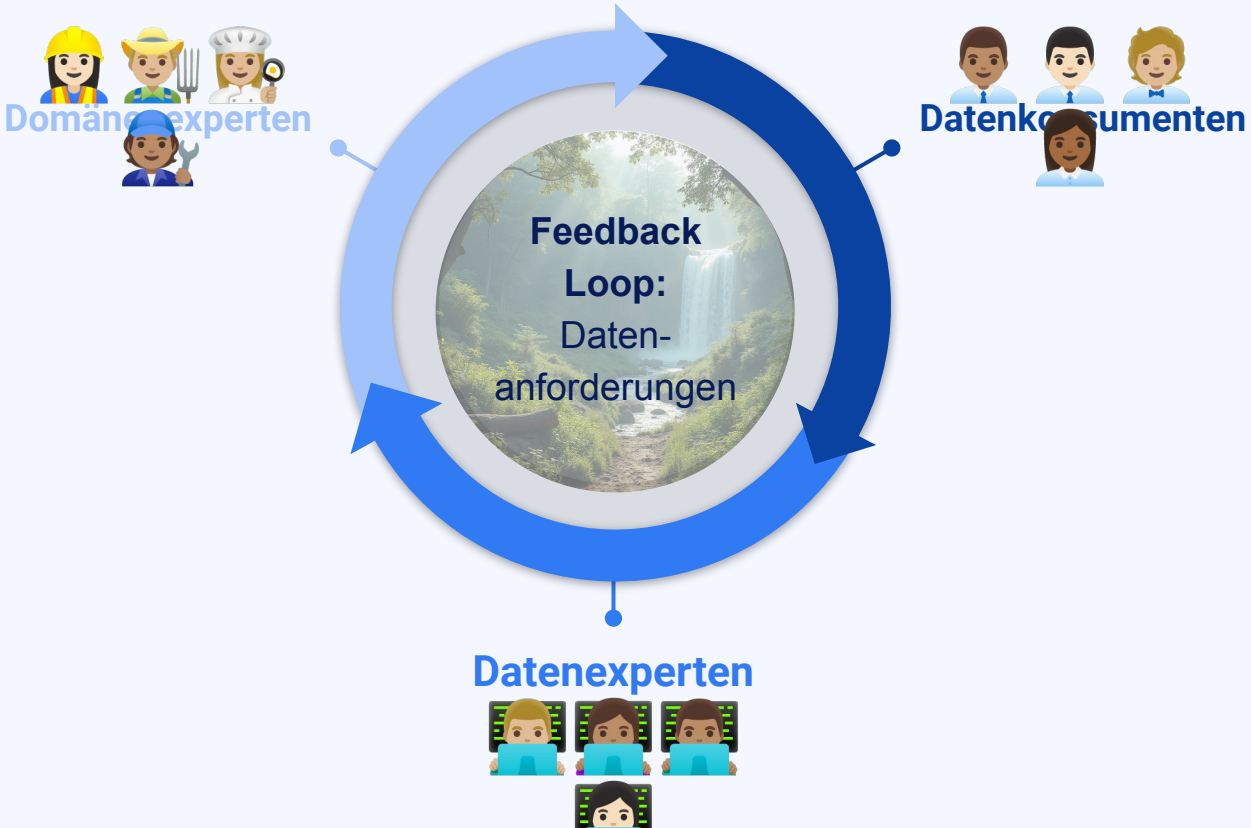
Gefahren im Daten-Dschungel

- Auf Basis von fehlerhaften Daten können **falsche Schlussfolgerungen gezogen** und **suboptimale Entscheidungen** getroffen werden
- Die Qualität eines AI- oder ML-Modells hängt maßgeblich von den Trainingsdaten ab (**Garbage-in-Garbage-out**)
- Beispielszenarien, in denen fehlerhafte Daten **großen Schaden** anrichten können:
  - *Empfehlungssysteme*
  - *Absatzprognose*
  - *Investitionsentscheidungen*

Mit Plan sicher durch den Daten-Dschungel bewegen

1. Kooperatives Erarbeiten von **Annahmen**
2. Definition von **Strategien** für den Umgang mit Auffälligkeiten und Fehlern
3. **Integration** von Datenvalidierung in Datenpipelines
4. **Überwachen** der Validierungsergebnisse und aktualisieren der Annahmen

Herausforderung: Anforderung sammeln & konsolidieren



inovex

Datenqualität umfasst viele Dimensionen

Regex Pattern

Vollständigkeit

Datentypen

Nullability

Schwellen-
werte

semantische Korrektheit

syntaktische Korrektheit

Dates/
Timestamps

Wertebereiche

Korrelationen

Aktualität

Relevanz

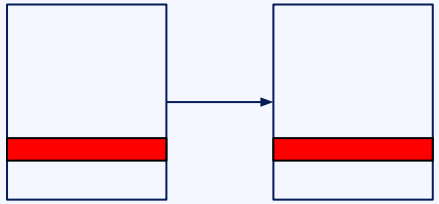
Stakeholder
Bedeutung

inovex

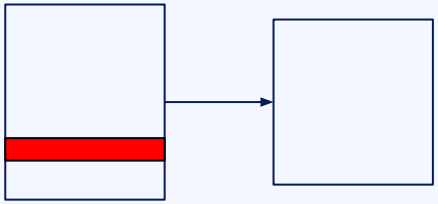
Kritikalität bestimmt den Umgang mit fehlerhaften Daten

- Je nach **Datensatz** und **Annahme** sind unterschiedliche Reaktionen erforderlich
- Unabhängig von der Reaktion sind **Benachrichtigungen** in jedem Falle sinnvoll
- Wurden **fehlerhafte Daten identifiziert** gibt es verschiedene Möglichkeiten, darauf zu **reagieren**:

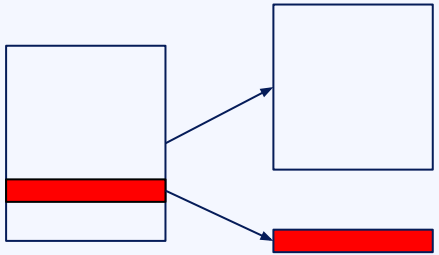
Kritikalität bestimmt den Umgang mit fehlerhaften Daten



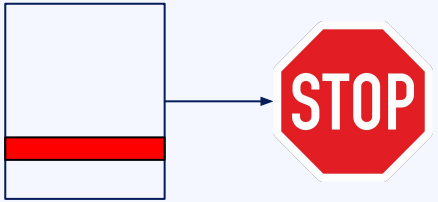
(a) weiterverarbeiten



(b) ignorieren

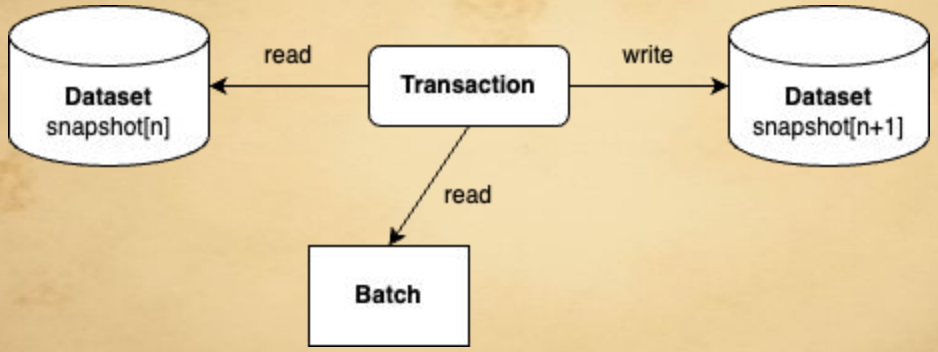


(c) aussortieren

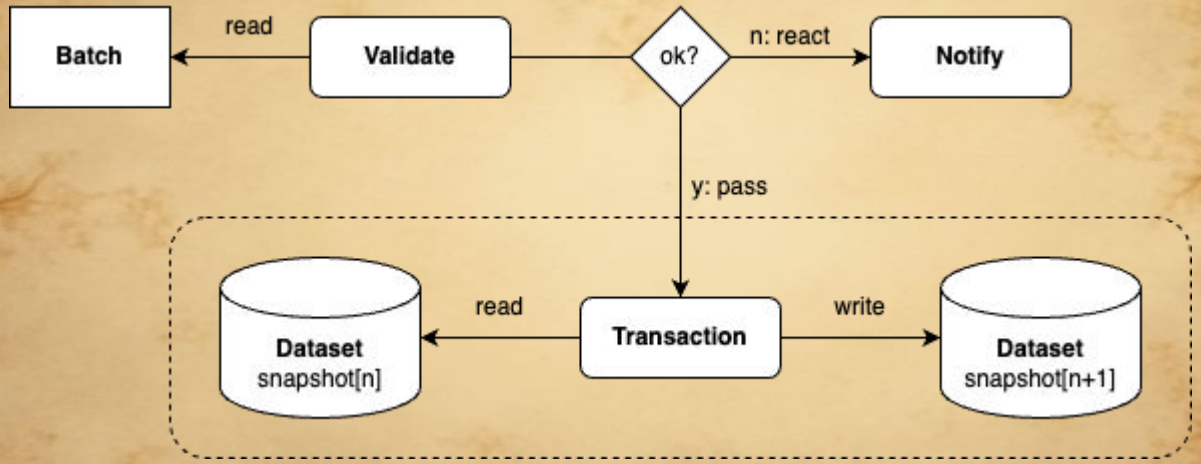


(d) stoppen

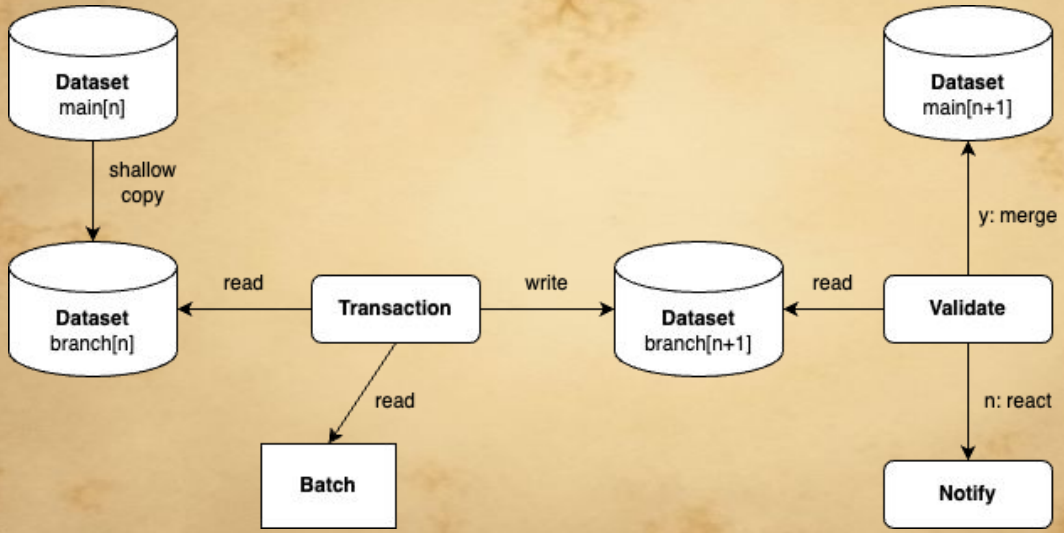
Transaktion ohne Validierung der Datenqualität



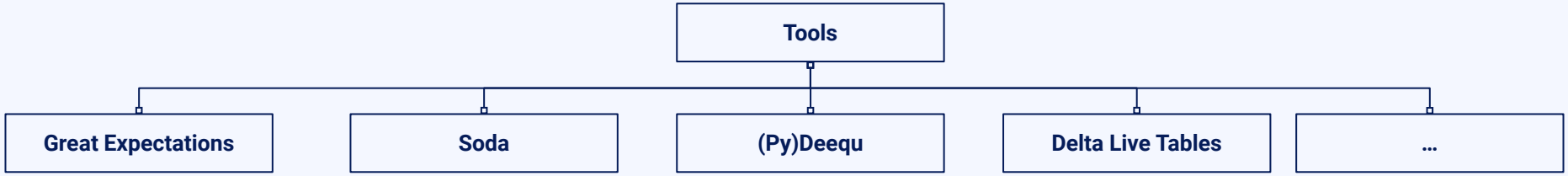
Validierung neuer Daten vor Integration



Validierung neuer Daten nach Integration



Ein breites Angebot an Werkzeugen



inovex

Great Expectations: Die etablierte Toolbox für den Ernstfall

“Great Expectations is the leading tool for validating, documenting, and profiling your data to maintain quality and improve communication between teams”

- **“Eierlegende Wollmilchsau”** im Datenqualitäts-Dschungel
- Breites Einsatzspektrum, aber **steile Lernkurve**
- Umfangreiches Set an **Expectation Optionen**
- Große **Community** und Verbreitung

inovex

Great Expectations: Context Setup

Data Context	Data Source	expectation.json	Checkpoint	Data Docs

```
from great_expectations.data_context import EphemeralDataContext,
FileDataContext


# Fluechtiger in Memory Context
context = EphemeralDataContext(project_config=project_config)


# Context mit Filesystem Backend
path_to_folder = "./"
context = FileDataContext.create(project_root_dir=path_to_folder)
```

<0.18.x

Great Expectations: Datenquellen

Data Context	**Data Source**	expectation.json	Checkpoint	Data Docs

```
# Nutzen einer Postgres als Datenquelle
pg_datasource = context.sources.add_postgres(
    name="pg_datasource", connection_string=PG_CONNECTION_STRING
)

# Nutzen einer Datei in S3 als Datenquelle
data_source = context.sources.add_pandas_s3(
   name=source_name, bucket=bucket_name
)


data_asset = data_source.add_csv_asset(
   name=asset_name, batching_regex=batching_regex, s3_prefix=s3_prefix
)


data_asset.build_batch_request()
```

<0.18.x

Great Expectations: Expectation Suite

Data Context	Data Source	expectation.json	Checkpoint	Data Docs

```
{
 "expectation_suite_name": "d2d_expectations",
 "ge_cloud_id": null,
 "expectations": [

{
 "expectation_type": "expect_column_values_to_be_between",
 "kwargs": {
   "column": "listener_count",
   "max_value": 1000000,
   "min_value": 1,
 },
 "meta": {}
},

...

}
```

<0.18.x

Great Expectations: Checkpoint

Data Context	Data Source	expectation.json	Checkpoint	Data Docs

```
validations = [
   {
       "batch_request": batch_request,
       "expectation_suite_name": expectation_suite,
       "action_list": action_list,
   }
]


checkpoint = context.add_or_update_checkpoint(
   name="Checkpoint",
   validations=validations,
   runtime_configuration={}},
)


checkpoint.run(run_name=run_name)
```

<0.18.x

Data Docs: Ergebnisse als HTML

Data Context	Data Source	expectation.json	Checkpoint	Data Docs

great expectations

Home / Validations / demo_expectation_suite / postgres_busbreakdown_data / 20240615-153040-my-run-name-template / 2024-06-15T17:30:40Z

Expectation Validation Result

Evaluates whether a batch of data matches expectations.

Actions

Validation Filter:

Show All | Failed Only

How to Edit This Suite

Show Walkthrough

Table of Contents

- Overview
- Table-Level Expectations
- bus_no
- busbreakdown_id
- has_contractor_notified_schools
- number_of_students_on_the_bus

Overview

Expectation Suite: demo_expectation_suite

Data asset: postgres_busbreakdown_data

Status: ✗ Failed

Statistics

Evaluated Expectations	5
Successful Expectations	4
Unsuccessful Expectations	1
Success Percent	80%

Show more info...

Table-Level Expectations

Search

Status	Expectation	Observed Value
✓	Must have exactly 199998 rows.	199998

bus_no

Search

Status	Expectation	Observed Value
✓	is a required field.	--

busbreakdown_id

Stay current on everything GX with our newsletter Subscribe

Der Challenger im DQ Dickicht

“Build reliable data products and pipelines with Soda, the GenAI-first platform for data quality. Embed tests into your workflows and monitor data quality health any way you like–through out-of-the-box observability or declarative testing.”

- Breites Angebot an **Checks** und **Datenquellen**
- Möglichkeit zur Erstellung **individueller Checks**
- **Intuitives** Nutzungskonzept
- aktuell **weniger verbreitet**

inovex

Datasets und Checks werden in Soda in YAML definiert

checks.yml

```
checks for actor:
# check for missing columns
  - missing_count(first_name) = 0
  - missing_count(last_name) = 0
```

config.yml

```
data_source my_local_db:
  type: postgres
  connection:
    host: localhost
    port: '5432'
    username: user
    password: myPassword
  database: dvdrental
  schema: public
```

CLI

```
soda scan checks.yml -d my_local_db -c config.yml
```

Python

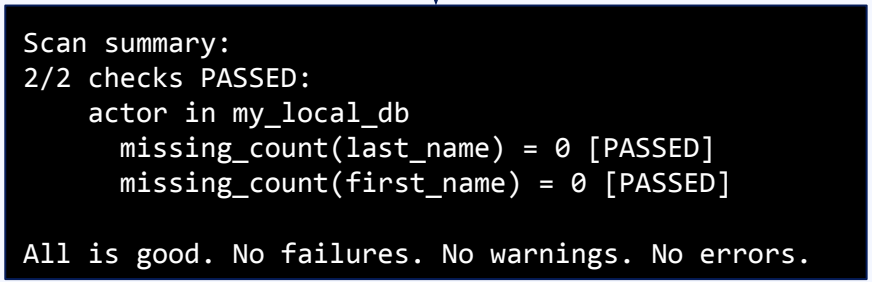
```
from soda.scan import Scan

scan = Scan()
scan.set_data_source_name("my_local_db")
scan.add_configuration_yaml_file("./config.yml")
scan.add_sodacl_yaml_file("./checks.yml")

scan.execute()
```

Datasets und Checks werden in Soda in YAML definiert

SODA

CLI

```
soda scan checks.yml -d my_local_db -c config.yml
```



inovex

Soda ist ein starker Konkurrent zum etablierten GX

	SODA	Great Expectations
Integrationsmöglichkeiten	+	+
Funktionsumfang	+	+
Nutzungskonzept	+	-
Dokumentation	+	-
Verbreitung/Community	-	+

inovex

Fazit: Überleben im Daten-Dschungel

Risiken und Gefahren kennen:

- Konsequenzen fehlerhafter Daten
- Vorsorge ist besser als Nachsorge

Verbündete mit einbeziehen:

- Domänenexperten, Nutzer, Stakeholder, ...

Stets den Kompass im Blick behalten:

- Iterative Herangehensweise, inkrementelle Verbesserung

Geschulter Umgang mit Werkzeugen:

- Integrierte Datenvalidierung
- Strategien für den Umgang mit Auffälligkeiten

Vielen Dank!

Marcel Spitzer
Data & ML Engineer

marcel.spitzer@inovex.de

Florian Gräbe
Data & ML Engineer

florian.graebe@inovex.de

Bildgenerierung: fal.ai/flux

inovex

Vortragsfolien zum Download

Besuchen Sie uns gerne auch an unserem Stand im Foyer!

inovex