Test Automation in the Cloud

with Databricks

syncwork

Agenda

1	Testing Strategy Evaluation
2	Syncwork Test Automation Framework (TAF)
3	Key Features, Tests & results
4	Dashboard
5	Architecture Overview
6	Live Demo

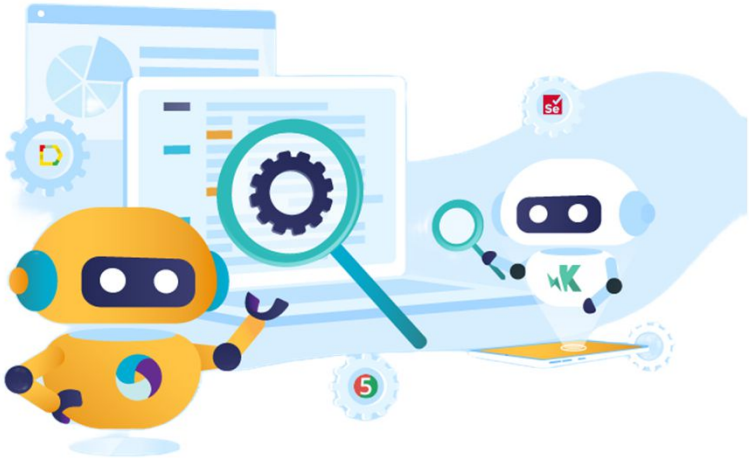
sync work

Testing Strategy Evaluation

Risks and Painpoints

Some disadvantages in Development process due to lack of Test Strategy Implementation:

- Higher System Maintenance Costs.
- Inadequate Quality Assurance.
- Limited Automation Opportunities.
- Difficulty in Debugging.



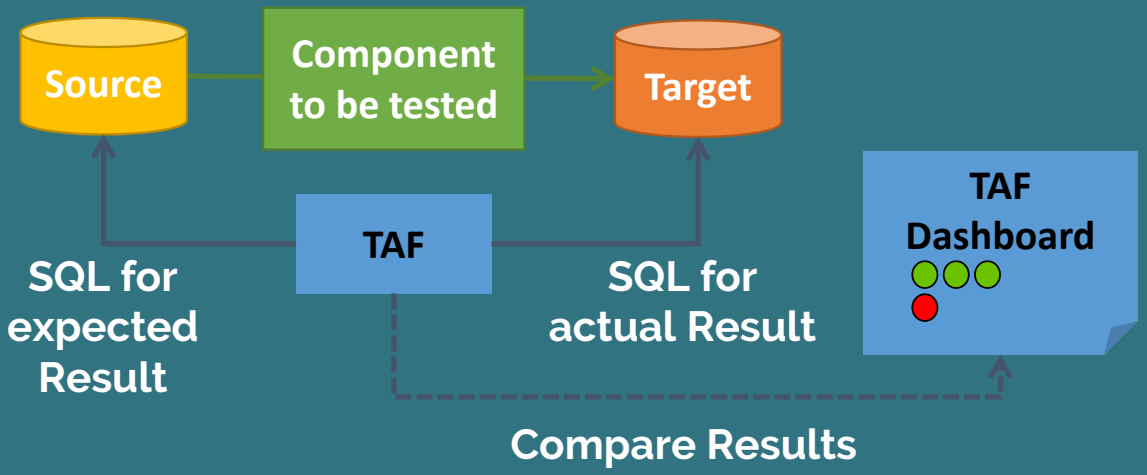
Opportunities

Areas and activities affected possitively by a Test Strategy:

- Rising Demand for Data Quality Assurance.
- Growing Complexity of ETL Processes.
- Compliance Requirements.
- DevOps Automation Pipeline.
- Automated Data Comparison.
- Regression Testing Efficiency.

TAF – **T**est **A**utomation **F**ramework

- **What is TAF?**
  - **T**est **A**utomation **F**ramework, developed for data driven tests
  - First Implementation in regulated environments
  - Supports parallel ETL approach
  - Performs end-to-end tests based on custom SQLs
  - Databricks



syncwork

Key features

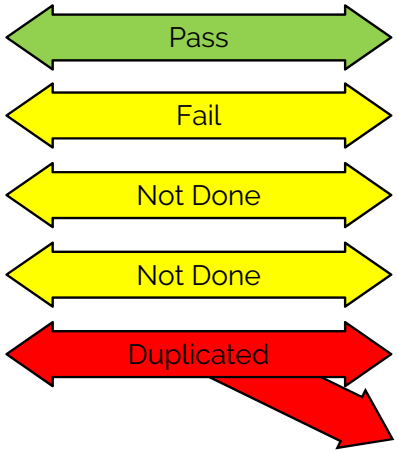
- Automated / scheduled execution of multiple test sets
- Automated result comparison
- Capable for mass testing (>1 mio records)
- Customization of test cases
- Independence of data source and target
- Identification of failure fields and values
- Result cleanup
- Dashboard with Result

syncwork

Tests results

Comparison of test results

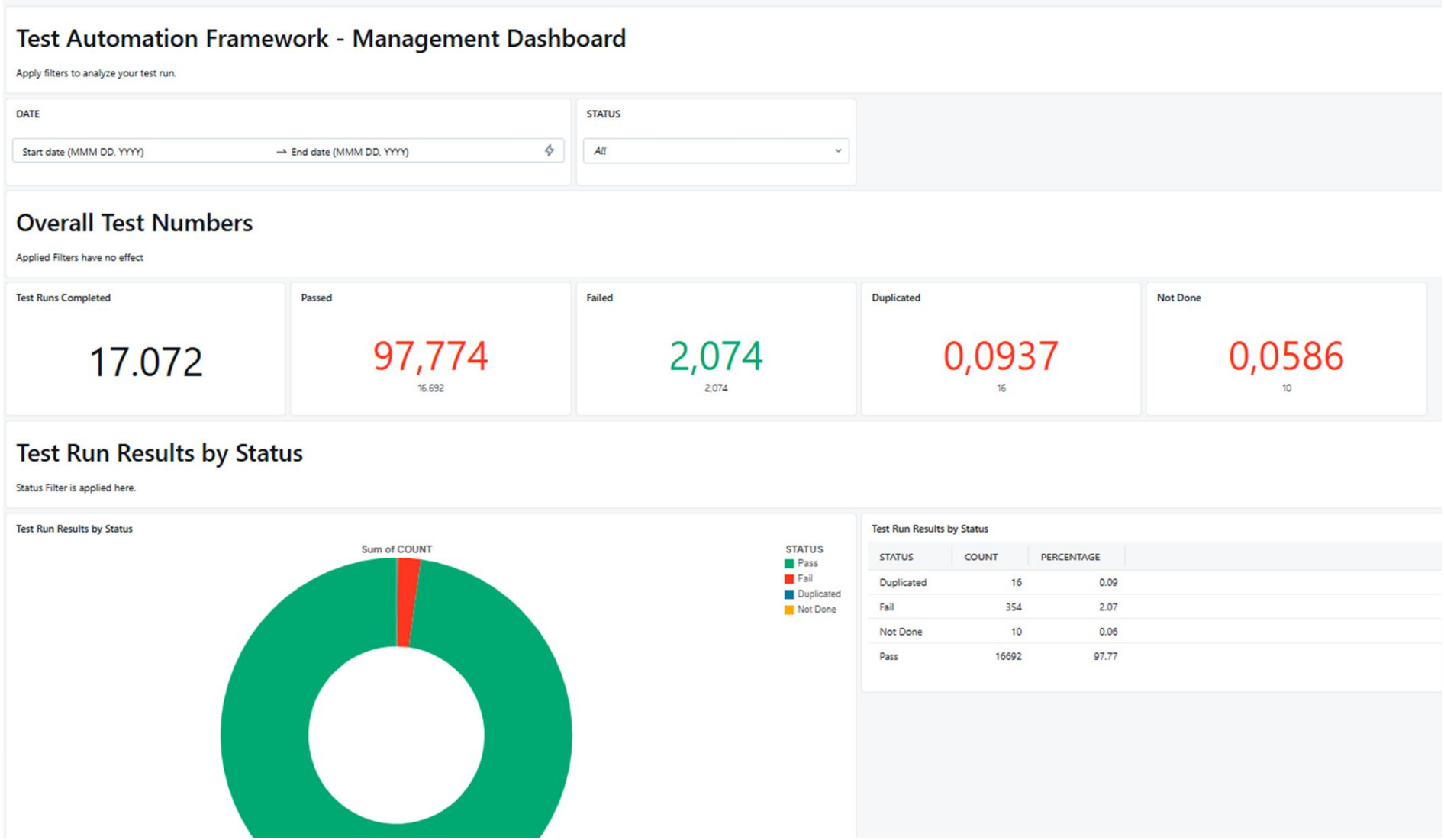
TEST_SOURCE	
TS_PRIMARY_KEY	TS_RESULT
4711	(1,'Data1', 0)
4712	(2,'Data2', 1)
4714	(1,'Data4', 1)
4715	(1,'Data5',0)



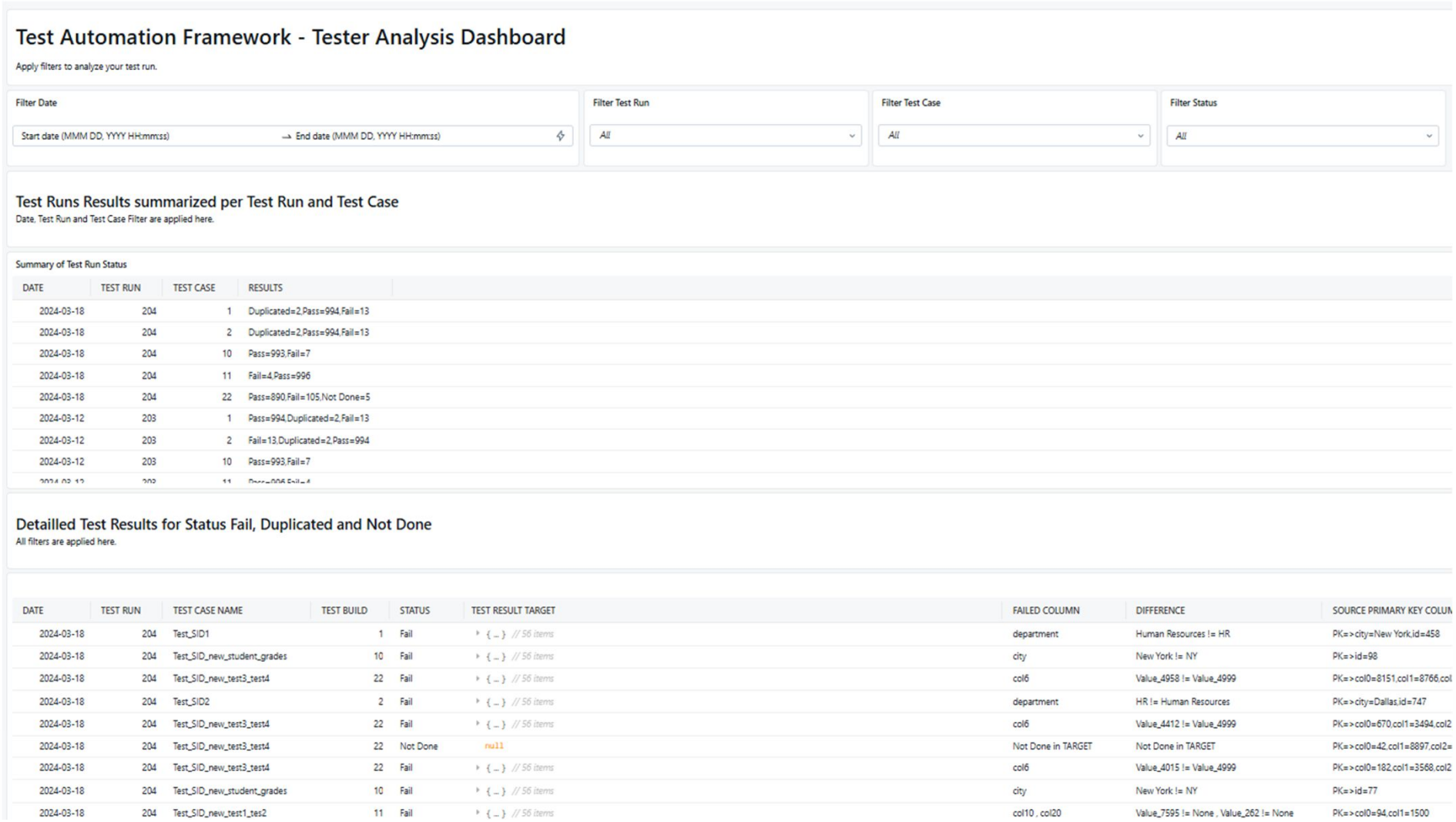
TEST_TARGET	
TT_PRIMARY_KEY	TT_RESULT
4711	(1,'Data1', 0)
4712	(2,'Datax', 1)
4713	(1,'Data3', 0)
4715	(1,'Data5',0)
4715	(1,'Data5',1)

syncwork

TAF Dashboard I



TAF Dashboard II



Test Automation Framework - Tester Analysis Dashboard

Apply filters to analyze your test run.

Filter Date: Start date (MMM DD, YYYY HH:mm:ss) → End date (MMM DD, YYYY HH:mm:ss)

Filter Test Run: All

Filter Test Case: All

Filter Status: All

Test Runs Results summarized per Test Run and Test Case

Date, Test Run and Test Case Filter are applied here.

Summary of Test Run Status

DATE	TEST RUN	TEST CASE	RESULTS
2024-03-18	204	1	Duplicated=2,Pass=994,Fail=13
2024-03-18	204	2	Duplicated=2,Pass=994,Fail=13
2024-03-18	204	10	Pass=993,Fail=7
2024-03-18	204	11	Fail=4,Pass=996
2024-03-18	204	22	Pass=890,Fail=105,Not Done=5
2024-03-12	203	1	Pass=994,Duplicated=2,Fail=13
2024-03-12	203	2	Fail=13,Duplicated=2,Pass=994
2024-03-12	203	10	Pass=993,Fail=7

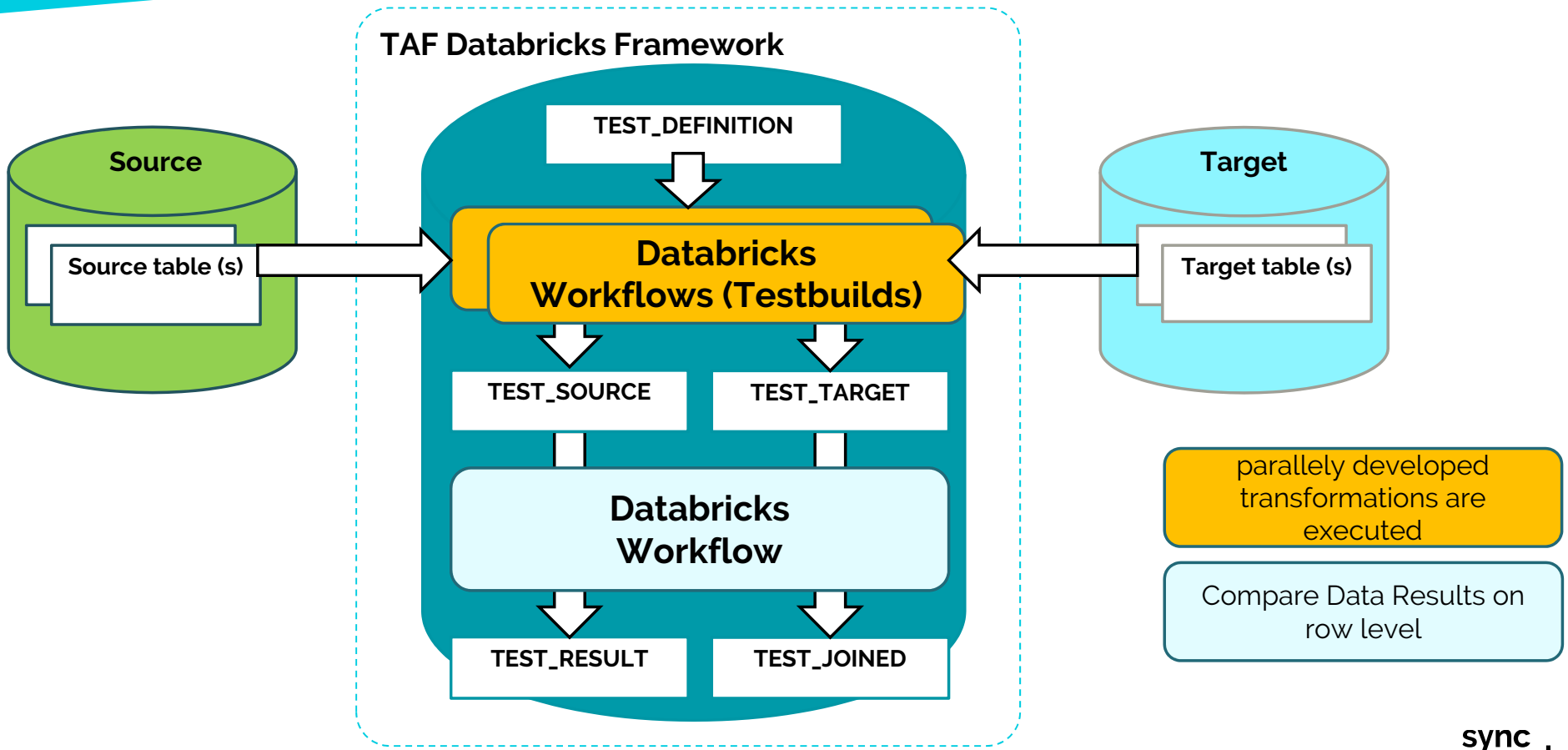
Detailled Test Results for Status Fail, Duplicated and Not Done

All filters are applied here.

DATE	TEST RUN	TEST CASE NAME	TEST BUILD	STATUS	TEST RESULT TARGET	FAILED COLUMN	DIFFERENCE	SOURCE PRIMARY KEY COLUM
2024-03-18	204	Test_SID1	1	Fail	{ … } // 56 items	department	Human Resources != HR	PK=>city=New York,id=458
2024-03-18	204	Test_SID_new_student_grades	10	Fail	{ … } // 56 items	city	New York != NY	PK=>id=98
2024-03-18	204	Test_SID_new_test3_test4	22	Fail	{ … } // 56 items	col6	Value_4958 != Value_4999	PK=>col0=8151,col1=8766,col
2024-03-18	204	Test_SID2	2	Fail	{ … } // 56 items	department	HR != Human Resources	PK=>city=Dallas,id=747
2024-03-18	204	Test_SID_new_test3_test4	22	Fail	{ … } // 56 items	col6	Value_4412 != Value_4999	PK=>col0=670,col1=3494,col2
2024-03-18	204	Test_SID_new_test3_test4	22	Not Done	null	Not Done in TARGET	Not Done in TARGET	PK=>col0=42,col1=8897,col2=
2024-03-18	204	Test_SID_new_test3_test4	22	Fail	{ … } // 56 items	col6	Value_4015 != Value_4999	PK=>col0=182,col1=3568,col2
2024-03-18	204	Test_SID_new_student_grades	10	Fail	{ … } // 56 items	city	New York != NY	PK=>id=77
2024-03-18	204	Test_SID_new_test1_tes2	11	Fail	{ … } // 56 items	col10 , col20	Value_7595 != None , Value_262 != None	PK=>col0=94,col1=1500

Architecture Overview

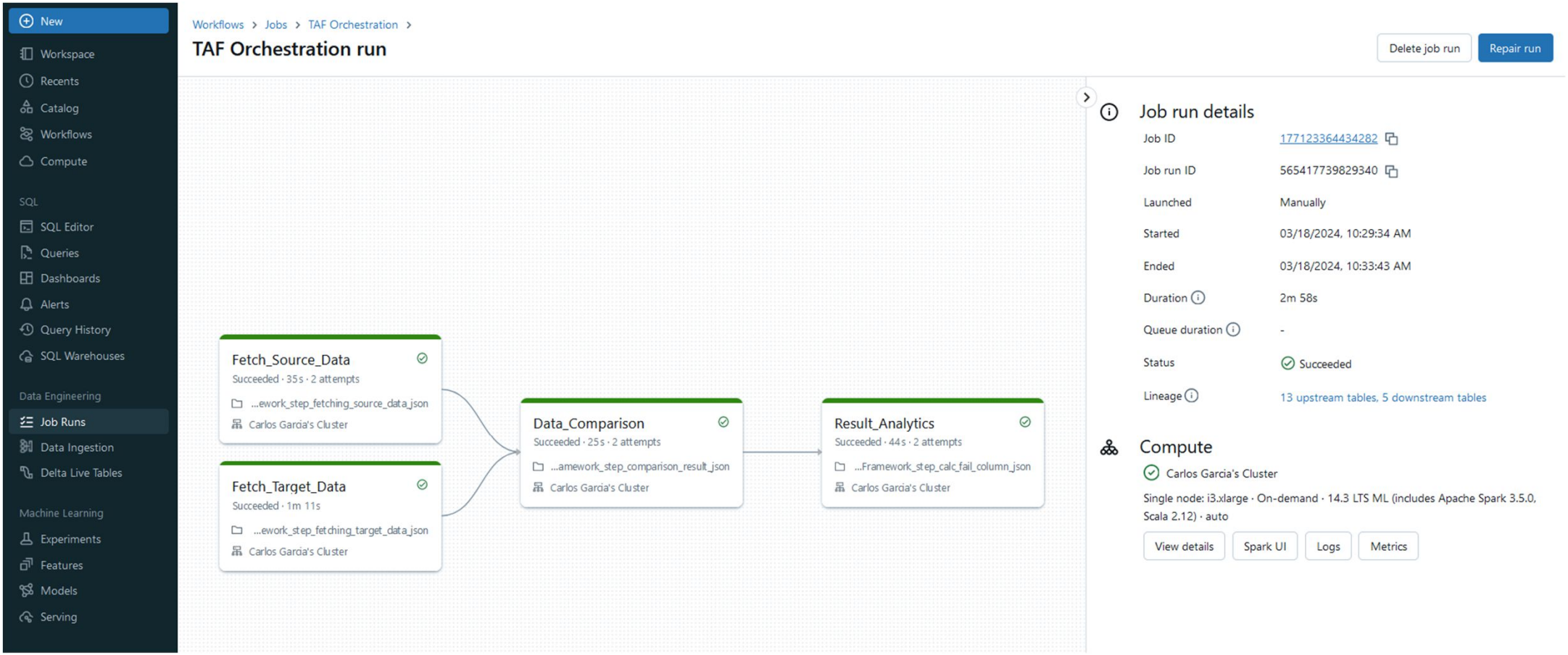
Execution and comparison



syncwork

Architecture Overview

TAF Workflow Orchestration



Architecture Overview

TAF Preparation – Test Configuration

Raw results +

	TEST_BUILD	TEST_SID	TABLE_NAME1	COLUMN_NAME1	TABLE_NAME2	COLUMN_NAME2	COLUMN_ID	IS_PK
1	23	Test_SID_new_test5_test6	test5	col0	test6	col0	0	PK
2	23	Test_SID_new_test5_test6	test5	col1	test6	col1_example	1	PK
3	23	Test_SID_new_test5_test6	test5	col2	test6	col2	2	null
4	23	Test_SID_new_test5_test6	test5	col3	test6	col3	3	null
5	23	Test_SID_new_test5_test6	test5	col4	test6	col4	4	null
6	23	Test_SID_new_test5_test6	test5	col5	test6	col5	5	null
7	23	Test_SID_new_test5_test6	test5	col6	test6	col6	6	null
8	23	Test_SID_new_test5_test6	test5	col7	test6	col7	7	null

Defines and maps the Source and Target table name and the column name.

Generates Source SQL Script:
Select ... from test5

Generates Target SQL Script:
Select ... from test6

syncwork

Architecture Overview

TAF Preparation – Primary Key Rename

Raw results +

	TEST_BUILD	TEST_SID	TABLE_NAME1	COLUMN_NAME1	TABLE_NAME2	COLUMN_NAME2	COLUMN_ID	IS_PK
1	23	Test_SID_new_test5_test6	test5	col0	test6	col0	0	PK
2	23	Test_SID_new_test5_test6	test5	col1	test6	col1_example	1	PK
3	23	Test_SID_new_test5_test6	test5	col2	test6	col2	2	null

```
SELECT pk_json.*, data_json.* FROM (
  --Dummy structure based on record of target table
  SELECT
  --All columns marked in column IS_PK with 'PK'
  from_json(pk, schema_of_json('{"pk":{"col0":"7196","col1":"485"}}')) as pk_json,
  --All columns marked in test_mapping_columns in IS_PK = NULL
  from_json(data, schema_of_json('{"data":{"col2":"1164", ... "col39":"Value_6181"}}')) as data_json FROM (
  --Target SQL Automatically created based on test_mapping_columns.
    SELECT
    to_json(named_struct('pk', struct(col0, col1_example as col1))) as pk,
    to_json(named_struct('data', struct(col2, , ... , col39))) as data FROM tat_samples.ba_xxx_prod.W_AE))
```

syncwork

Agenda

1	Testing Strategy Evaluation
2	Syncwork Test Automation Framework (TAF)
3	Key Features, Tests & results
4	Dashboard
5	Architecture Overview
6	Live Demo

sync work

Fragen / Diskussion

28-03-2024
Carlos Garcia
Consultant – Syncwork AG

Ihr Ansprechpartner:
Syncwork AG
Jörg Krempien

Lietzenburger Straße 69, 10719 Berlin
M +49 175 585 19 16
krempien@syncwork.de

sync work