Agenda

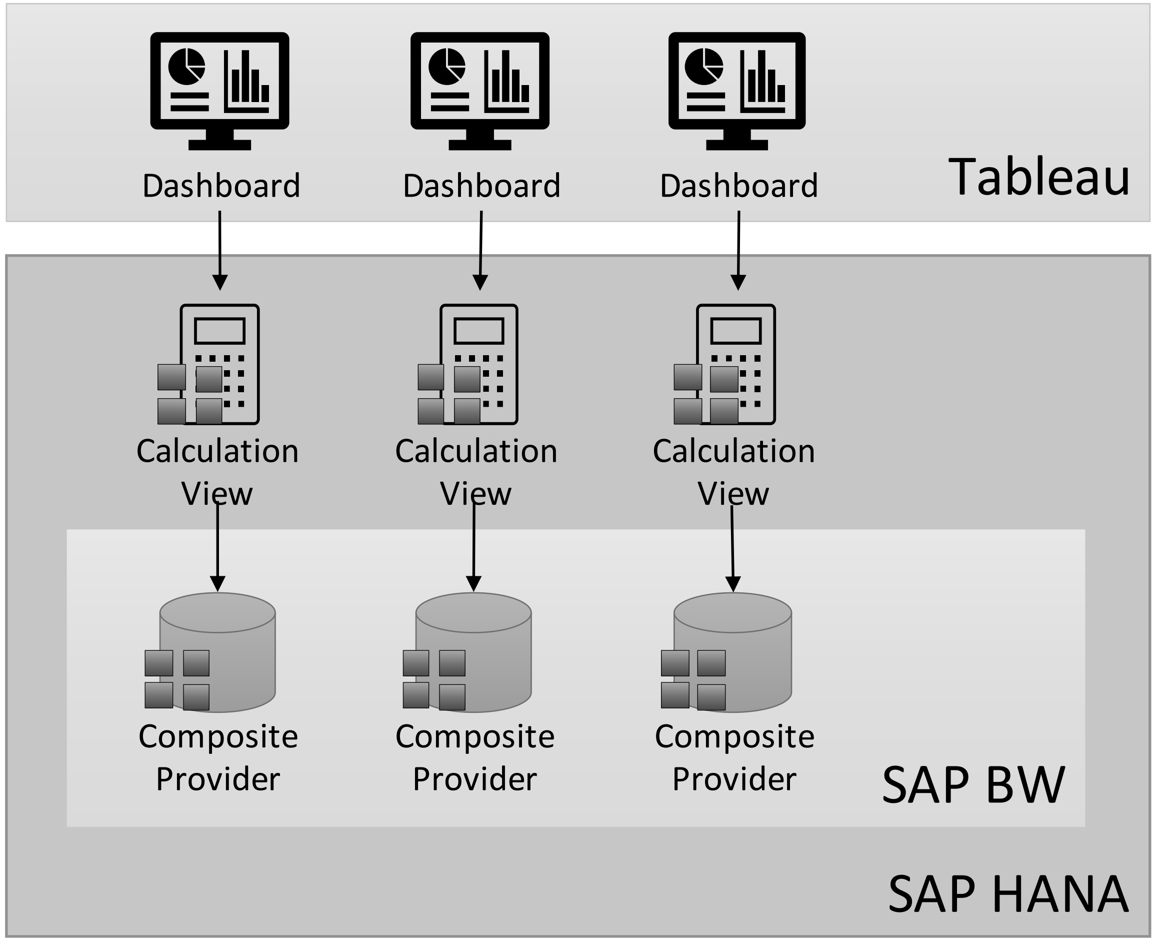
- Results and Learnings
- Live Demo
- Performance Best Practices
- Technical Deep Dive – Performance Analysis
- Q&A

Overview of Architecture

- Fast access with Tableau on Calculation Views
- No need for BW Connector
- Calculation Views as an abstraction layer for data models in BW

Advantages

- Best possible performance
- Best usability for the end user
- Extensibility possible
- Integration of BW analysis authorizations

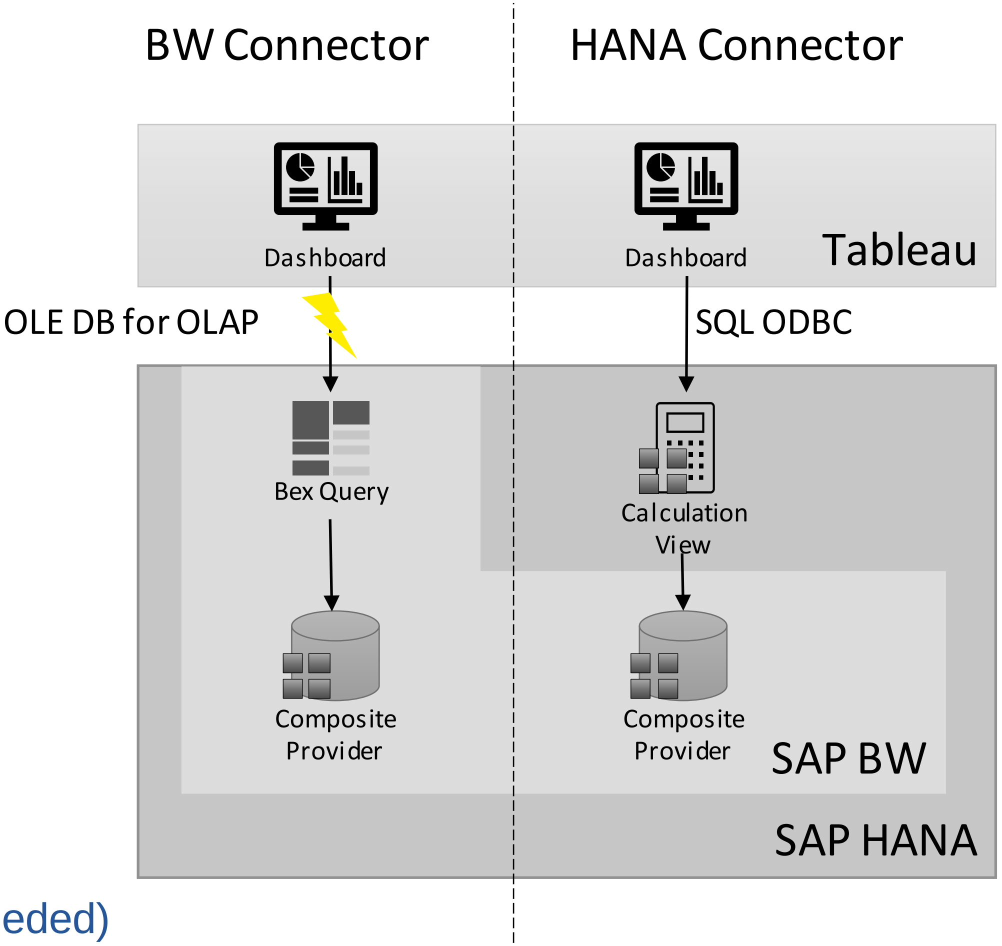


Performance

- OLE DB for OLAP interface
  - Long runtimes
  - Many Query-Features not supported
  - General restrictions of quantity (Cells, Dimensions)
  - Details in SAP Notes 323779 and 1048320

- BW as application server is often the bottleneck

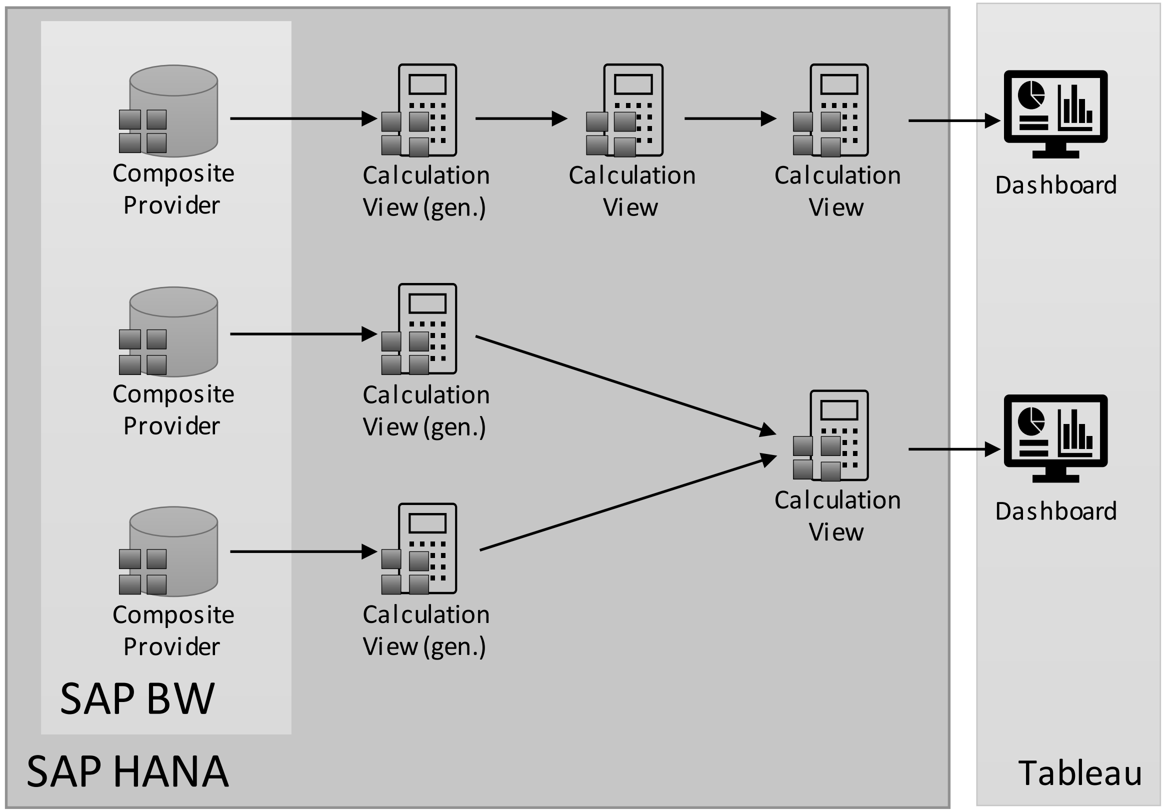
- Access is carried out on data base level
  - Usage of standard ODBC driver
  - Relational access of data structures
  - Parallelization of requests
  - Set limit of memory for requests (use workload groups if needed)



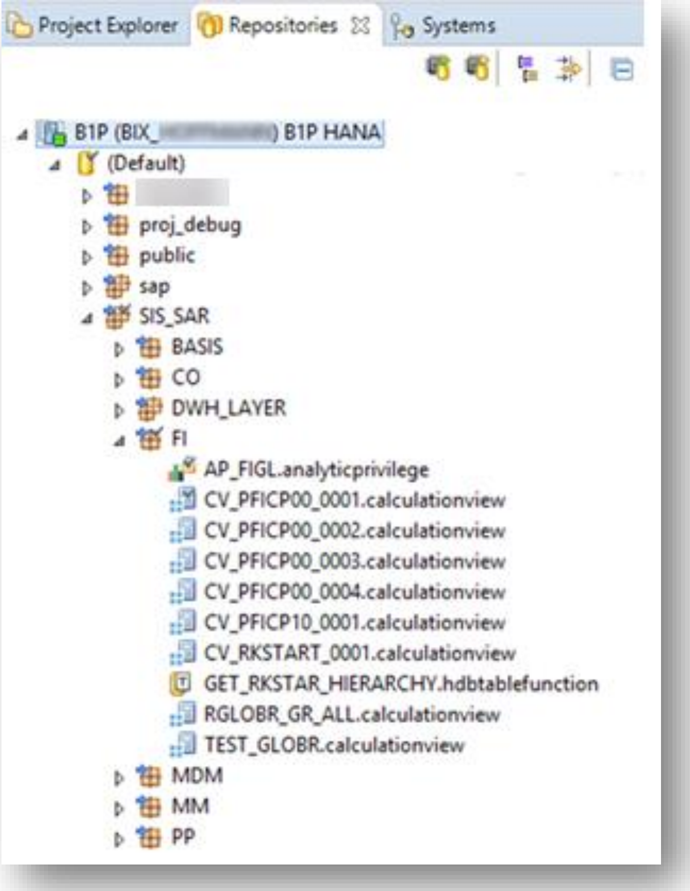
Usability

- Abstraction level offered
  - Simplification of partially extensive BW data models
  - From content model to key figure models
  - Modelling of easy to use key figures
  - Always right data independent of way of navigation
  - Additional persistence for performance issues

- Relational access offered
  - Data Blending
  - Aggregation functions
  - Restrictions regarding Cube-datasources omitted (https://help.tableau.com/current/pro/desktop/de-de/cubes.htm)



Usability



Project Explorer | Repositories | Systems

- B1P (BIX_[redacted]) B1P HANA
  - (Default)
    - [redacted]
    - proj_debug
    - public
    - sap
    - SIS_SAR
      - BASIS
      - CO
      - DWH_LAYER
      - FI
        - AP_FIGL.analyticprivilege
        - CV_PFICP00_0001.calculationview
        - CV_PFICP00_0002.calculationview
        - CV_PFICP00_0003.calculationview
        - CV_PFICP00_0004.calculationview
        - CV_PFICP10_0001.calculationview
        - CV_RKSTART_0001.calculationview
        - GET_RKSTAR_HIERARCHY.hdbtablefunction
        - RGLOBR_GR_ALL.calculationview
        - TEST_GLOBR.calculationview
      - MDM
      - MM
      - PP

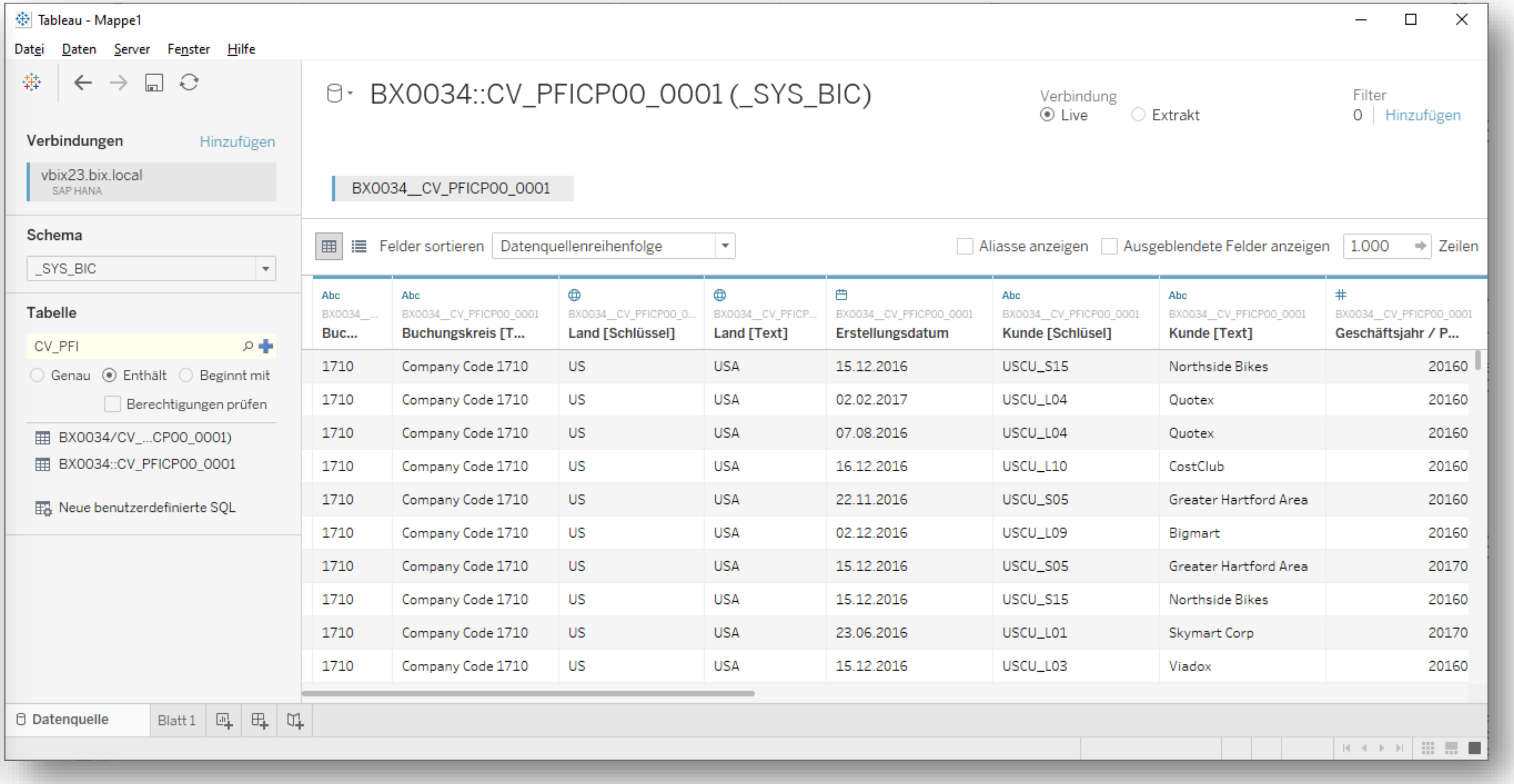


Tableau - Mappe1

Datei Daten Server Fenster Hilfe

BX0034::CV_PFICP00_0001 (_SYS_BIC)

Verbindung ● Live ○ Extrakt

Filter 0 | Hinzufügen

Verbindungen Hinzufügen

vbix23.bix.local
SAP HANA

Schema
_SYS_BIC

Tabelle
CV_PFI

○ Genau ● Enthält ○ Beginnt mit

□ Berechtigungen prüfen

BX0034/CV_...CP00_0001)
BX0034::CV_PFICP00_0001

Neue benutzerdefinierte SQL

BX0034__CV_PFICP00_0001

Felder sortieren Datenquellenreihenfolge □ Aliasse anzeigen □ Ausgeblendete Felder anzeigen 1.000 Zeilen

Buc...	Buchungskreis [T...	Land [Schlüssel]	Land [Text]	Erstellungsdatum	Kunde [Schlüssel]	Kunde [Text]	Geschäftsjahr / P...
1710	Company Code 1710	US	USA	15.12.2016	USCU_S15	Northside Bikes	20160
1710	Company Code 1710	US	USA	02.02.2017	USCU_L04	Quotex	20160
1710	Company Code 1710	US	USA	07.08.2016	USCU_L04	Quotex	20160
1710	Company Code 1710	US	USA	16.12.2016	USCU_L10	CostClub	20160
1710	Company Code 1710	US	USA	22.11.2016	USCU_S05	Greater Hartford Area	20160
1710	Company Code 1710	US	USA	02.12.2016	USCU_L09	Bigmart	20160
1710	Company Code 1710	US	USA	15.12.2016	USCU_S05	Greater Hartford Area	20170
1710	Company Code 1710	US	USA	15.12.2016	USCU_S15	Northside Bikes	20160
1710	Company Code 1710	US	USA	23.06.2016	USCU_L01	Skymart Corp	20170
1710	Company Code 1710	US	USA	15.12.2016	USCU_L03	Viadox	20160

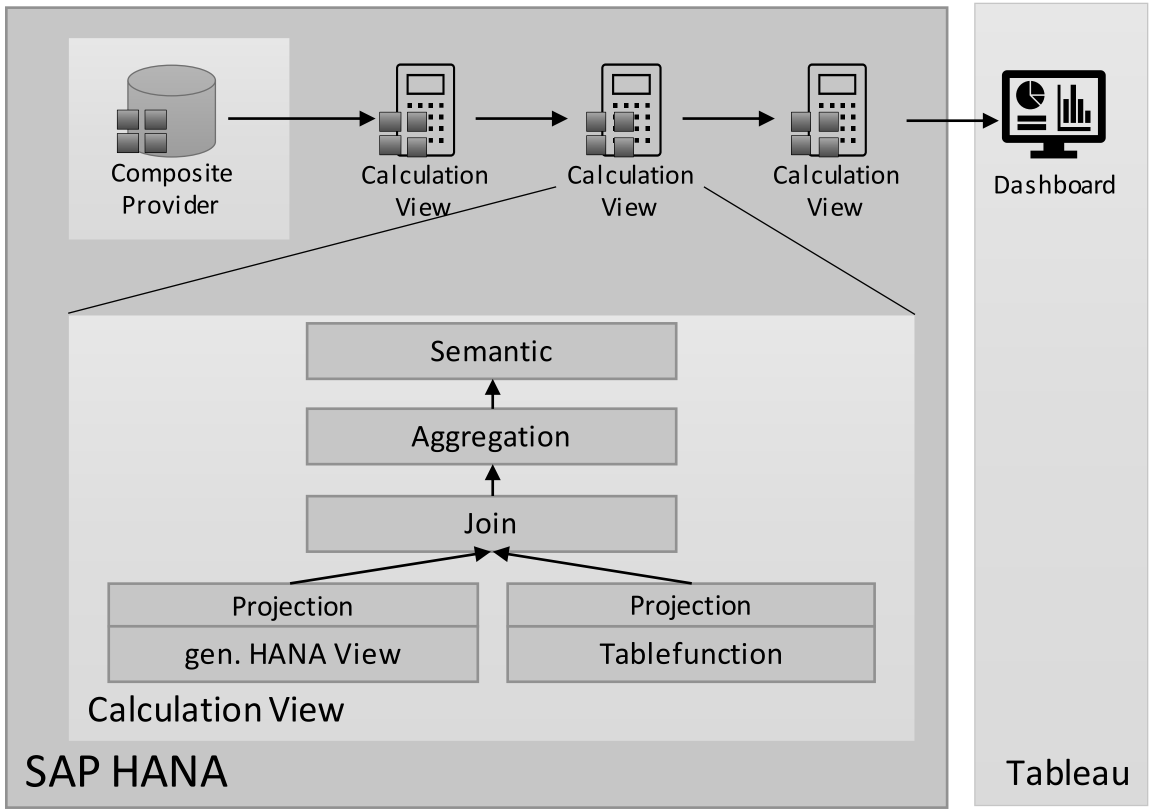
Datenquelle | Blatt 1

Extensibility

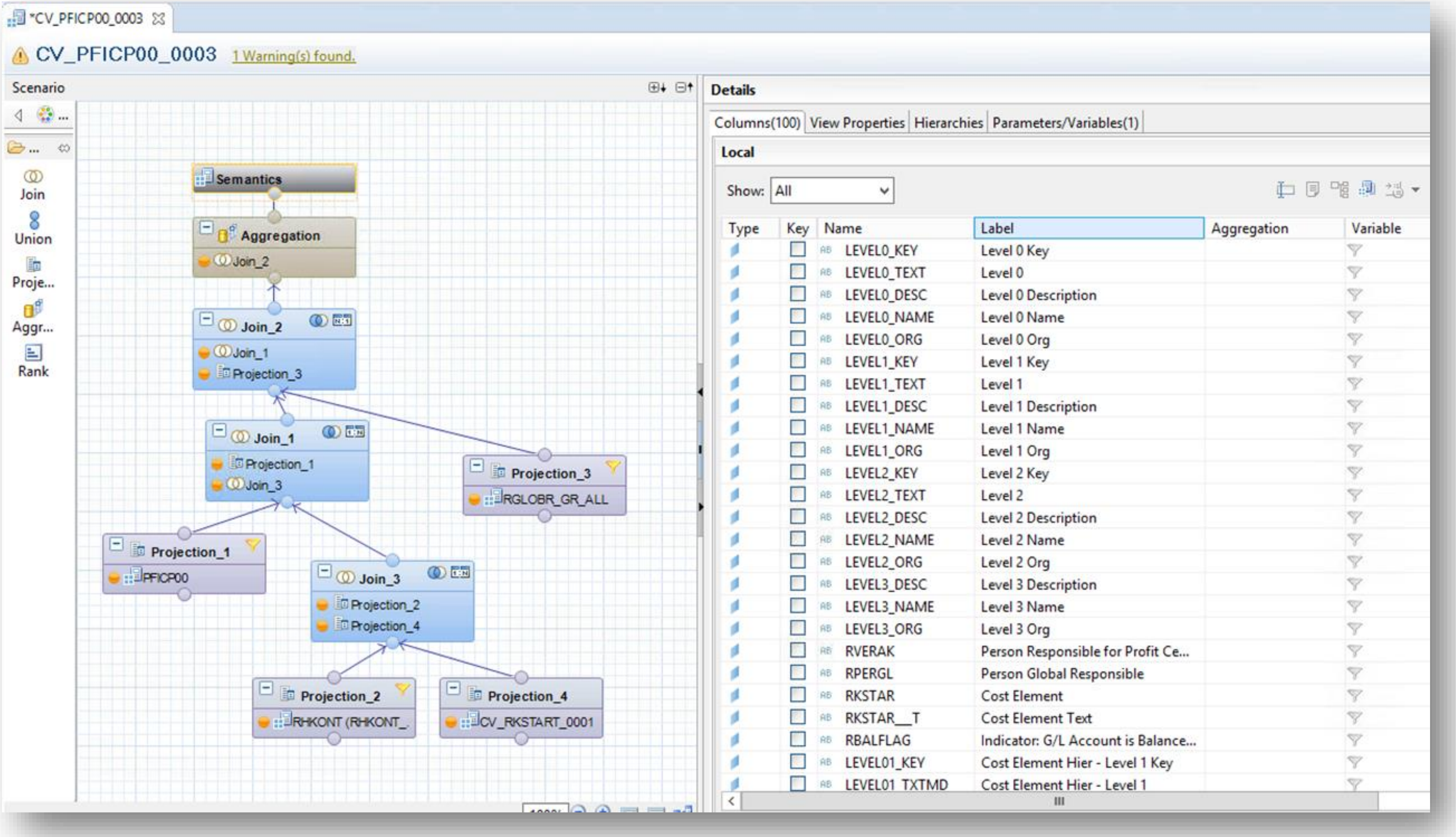
- Enable Calculation Views
  - Usage of SQL script in the form of table functions
  - Transformation of BW hierarchies in flat structures
    - Usage of unbalanced hierarchies
    - Usage of version-dependent hierarchies
    - Individual transfer towards Tableau

- Addition of data (virtual)
  - Prototypic extensions instead of remodelling
  - Subsequent transfer to ultimate model

- Robust to changes and extensions



Extensibility



*CV_PFICP00_0003

CV_PFICP00_0003 1 Warning(s) found.

Scenario

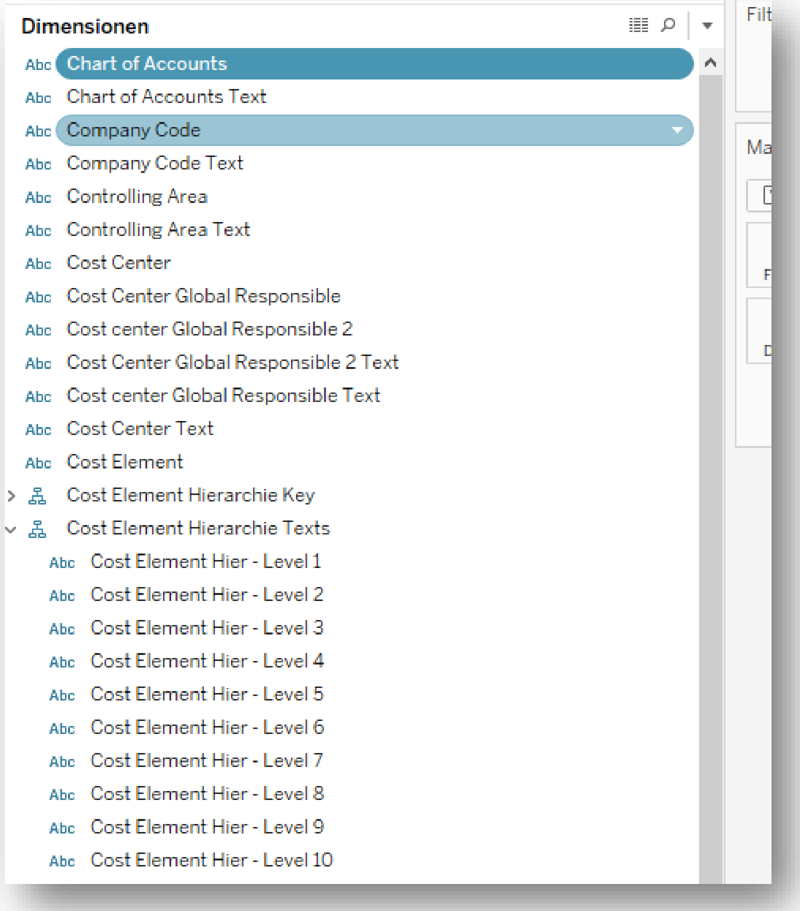
Details

Columns(100) | View Properties | Hierarchies | Parameters/Variables(1)

Local

Show: All

Type	Key	Name	Label	Aggregation	Variable
	☐	LEVEL0_KEY	Level 0 Key		
	☐	LEVEL0_TEXT	Level 0		
	☐	LEVEL0_DESC	Level 0 Description		
	☐	LEVEL0_NAME	Level 0 Name		
	☐	LEVEL0_ORG	Level 0 Org		
	☐	LEVEL1_KEY	Level 1 Key		
	☐	LEVEL1_TEXT	Level 1		
	☐	LEVEL1_DESC	Level 1 Description		
	☐	LEVEL1_NAME	Level 1 Name		
	☐	LEVEL1_ORG	Level 1 Org		
	☐	LEVEL2_KEY	Level 2 Key		
	☐	LEVEL2_TEXT	Level 2		
	☐	LEVEL2_DESC	Level 2 Description		
	☐	LEVEL2_NAME	Level 2 Name		
	☐	LEVEL2_ORG	Level 2 Org		
	☐	LEVEL3_DESC	Level 3 Description		
	☐	LEVEL3_NAME	Level 3 Name		
	☐	LEVEL3_ORG	Level 3 Org		
	☐	RVERAK	Person Responsible for Profit Ce...		
	☐	RPERGL	Person Global Responsible		
	☐	RKSTAR	Cost Element		
	☐	RKSTAR__T	Cost Element Text		
	☐	RBALFLAG	Indicator: G/L Account is Balance...		
	☐	LEVEL01_KEY	Cost Element Hier - Level 1 Key		
	☐	LEVEL01_TXTMD	Cost Element Hier - Level 1		



Dimensionen

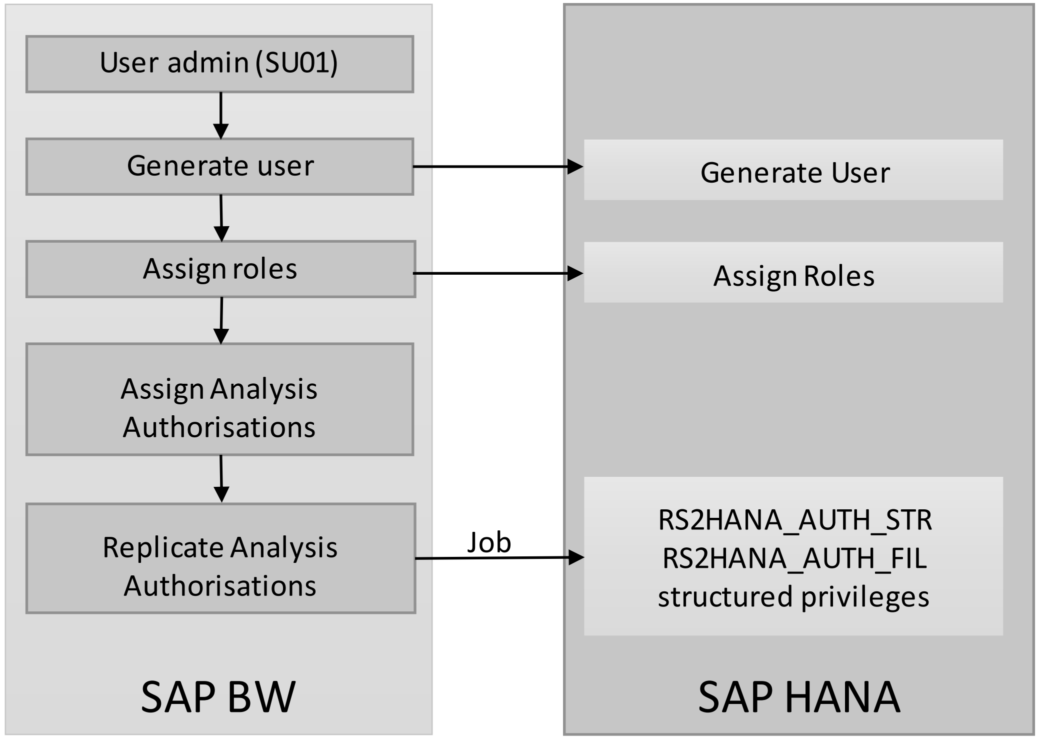
- Abc Chart of Accounts
- Abc Chart of Accounts Text
- Abc Company Code
- Abc Company Code Text
- Abc Controlling Area
- Abc Controlling Area Text
- Abc Cost Center
- Abc Cost Center Global Responsible
- Abc Cost center Global Responsible 2
- Abc Cost Center Global Responsible 2 Text
- Abc Cost center Global Responsible Text
- Abc Cost Center Text
- Abc Cost Element
- Cost Element Hierarchie Key
- Cost Element Hierarchie Texts
  - Abc Cost Element Hier - Level 1
  - Abc Cost Element Hier - Level 2
  - Abc Cost Element Hier - Level 3
  - Abc Cost Element Hier - Level 4
  - Abc Cost Element Hier - Level 5
  - Abc Cost Element Hier - Level 6
  - Abc Cost Element Hier - Level 7
  - Abc Cost Element Hier - Level 8
  - Abc Cost Element Hier - Level 9
  - Abc Cost Element Hier - Level 10

BW Analysis Authorisations

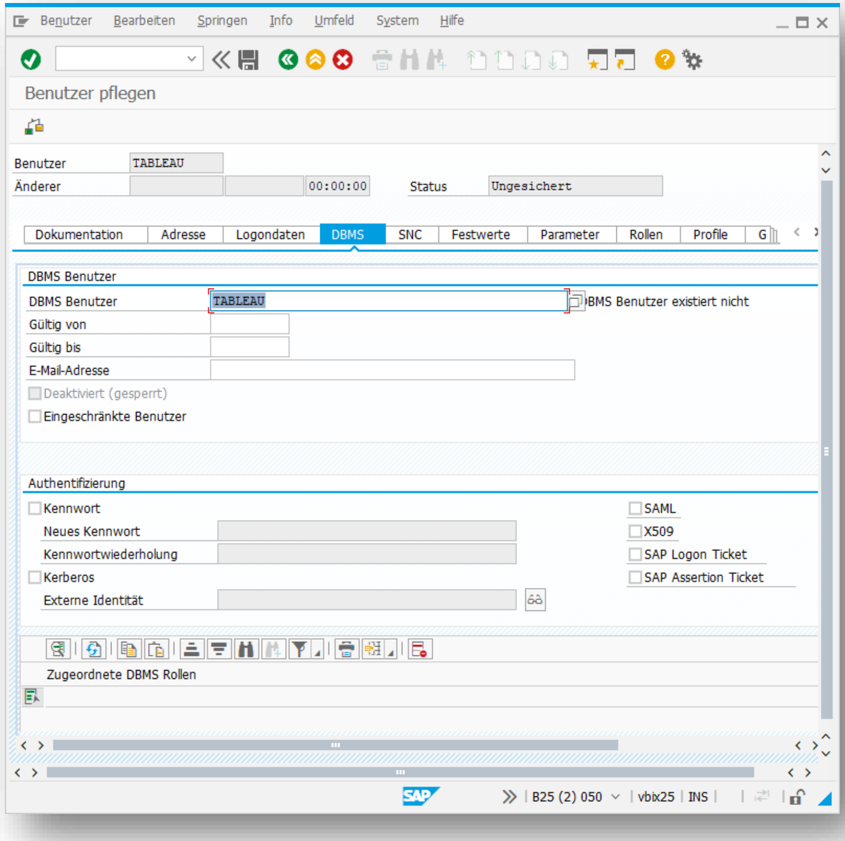
- Direct ministration of users in BW
  - DBMS connection in SU01
  - Direct generation of DB-users
  - Role-assignment via BW

- Push-Down
  - BW Analysis-authorizations accessible in HANA
  - Periodical job scheduled
  - HANA users automatically restricted

- Single-Sign-On



BW Analysis Authorisations



Benutzer pflegen

Benutzer	TABLEAU			
Änderer		00:00:00	Status	Ungesichert

DBMS Benutzer

DBMS Benutzer	TABLEAU
Gültig von	
Gültig bis	
E-Mail-Adresse	

DBMS Benutzer existiert nicht

□ Deaktiviert (gesperrt)

□ Eingeschränkte Benutzer

Authentifizierung

□ Kennwort

Neues Kennwort

Kennwortwiederholung

□ Kerberos

Externe Identität

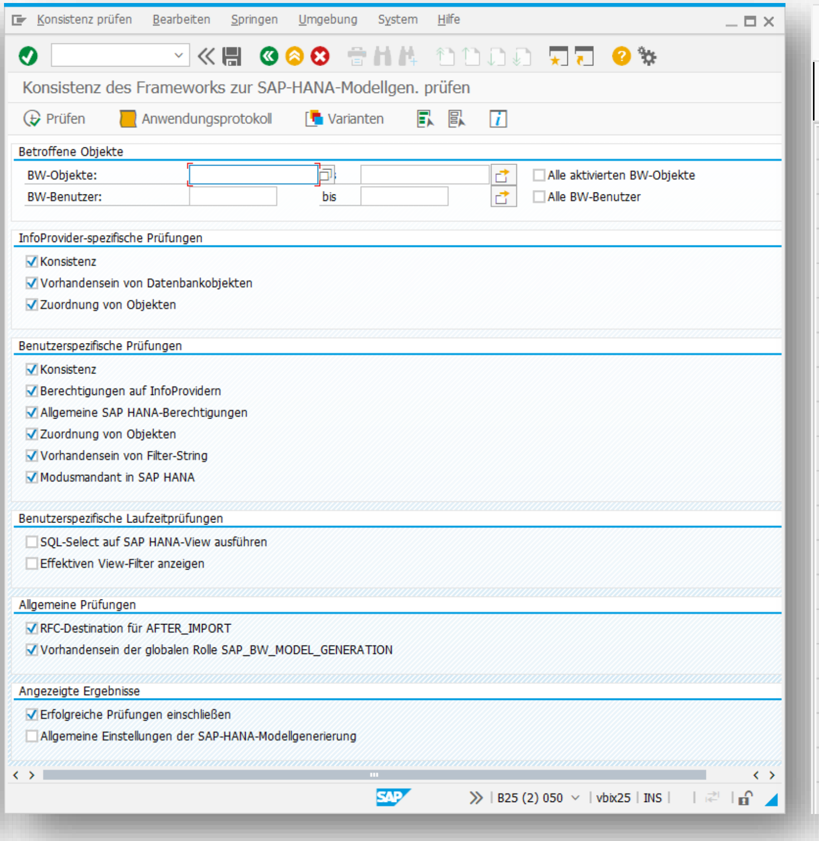
□ SAML

□ X509

□ SAP Logon Ticket

□ SAP Assertion Ticket

Zugeordnete DBMS Rollen



Konsistenz des Frameworks zur SAP-HANA-Modellgen. prüfen

Betroffene Objekte

BW-Objekte:

BW-Benutzer: bis

□ Alle aktivierten BW-Objekte

□ Alle BW-Benutzer

InfoProvider-spezifische Prüfungen

☑ Konsistenz

☑ Vorhandensein von Datenbankobjekten

☑ Zuordnung von Objekten

Benutzerspezifische Prüfungen

☑ Konsistenz

☑ Berechtigungen auf InfoProvidern

☑ Allgemeine SAP HANA-Berechtigungen

☑ Zuordnung von Objekten

☑ Vorhandensein von Filter-String

☑ Modusmandant in SAP HANA

Benutzerspezifische Laufzeitprüfungen

□ SQL-Select auf SAP HANA-View ausführen

□ Effektiven View-Filter anzeigen

Allgemeine Prüfungen

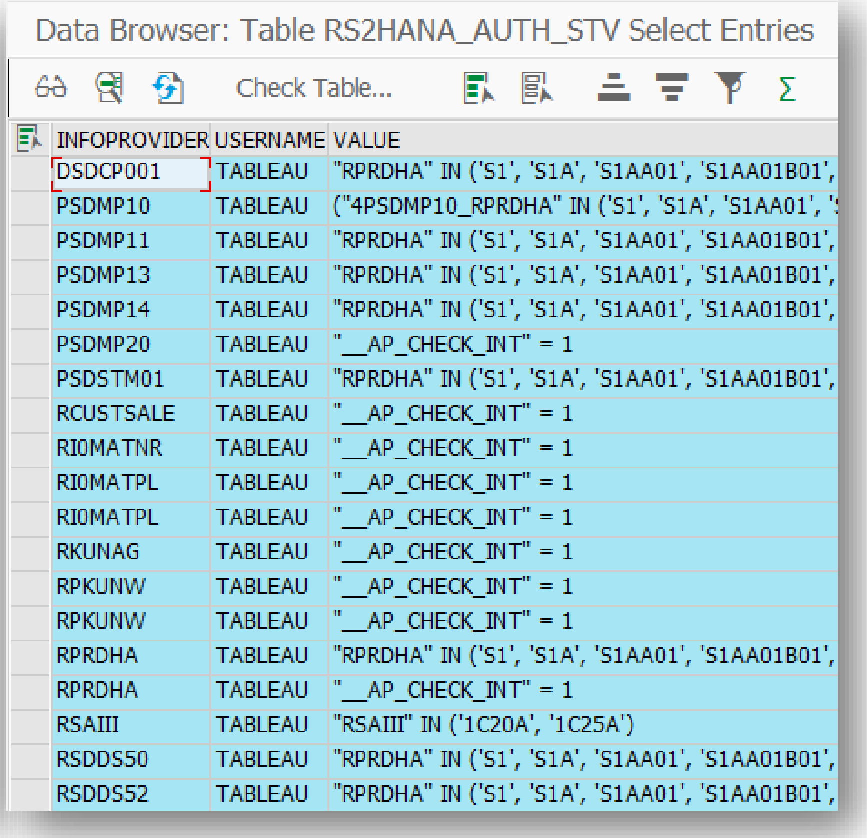
☑ RFC-Destination für AFTER_IMPORT

☑ Vorhandensein der globalen Rolle SAP_BW_MODEL_GENERATION

Angezeigte Ergebnisse

☑ Erfolgreiche Prüfungen einschließen

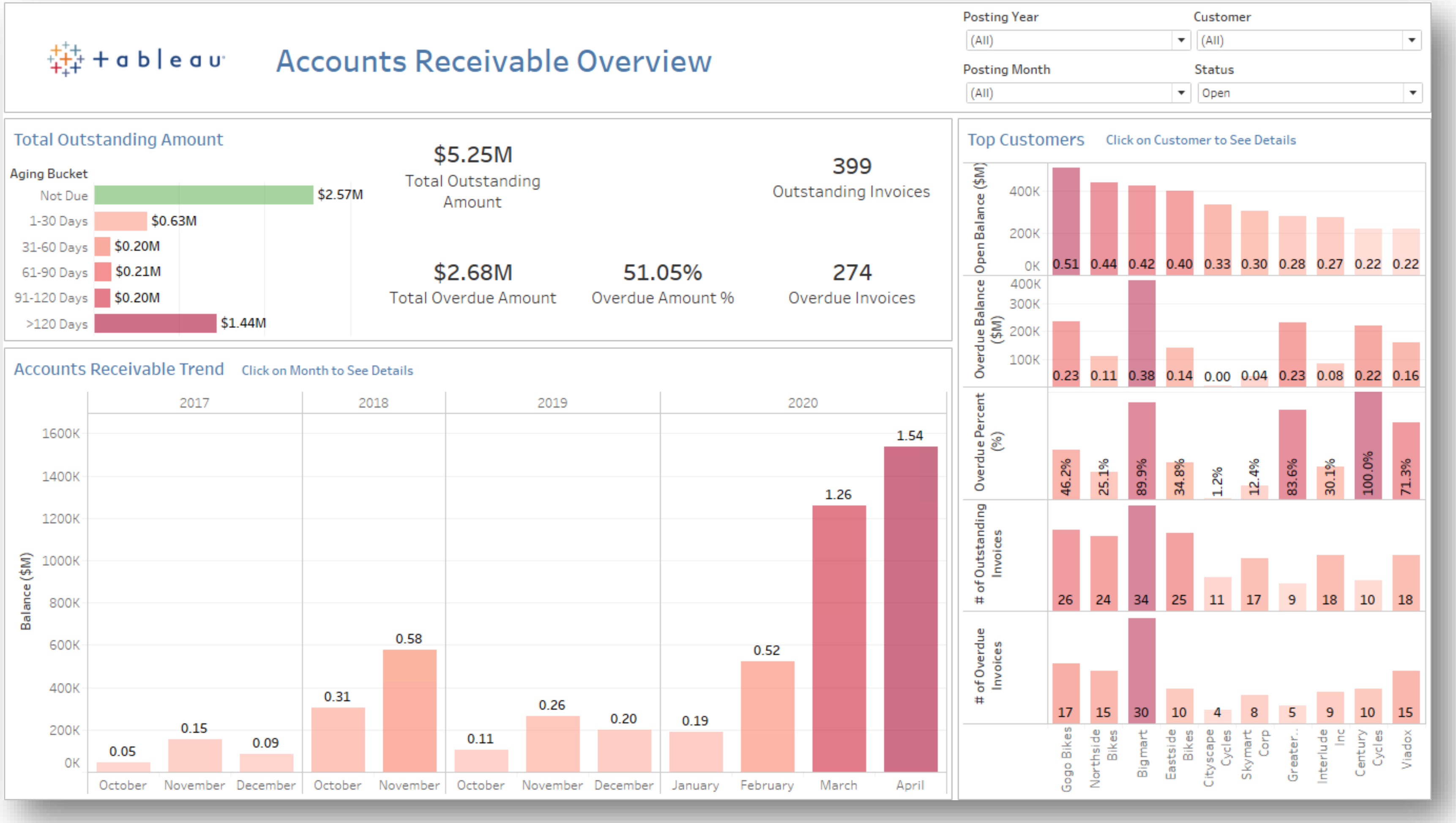
□ Allgemeine Einstellungen der SAP-HANA-Modellgenerierung


Data Browser: Table RS2HANA_AUTH_STV Select Entries

INFOPROVIDER	USERNAME	VALUE
DSDCP001	TABLEAU	"RPRDHA" IN ('S1', 'S1A', 'S1AA01', 'S1AA01B01',
PSDMP10	TABLEAU	("4PSDMP10_RPRDHA" IN ('S1', 'S1A', 'S1AA01', '
PSDMP11	TABLEAU	"RPRDHA" IN ('S1', 'S1A', 'S1AA01', 'S1AA01B01',
PSDMP13	TABLEAU	"RPRDHA" IN ('S1', 'S1A', 'S1AA01', 'S1AA01B01',
PSDMP14	TABLEAU	"RPRDHA" IN ('S1', 'S1A', 'S1AA01', 'S1AA01B01',
PSDMP20	TABLEAU	"__AP_CHECK_INT" = 1
PSDSTM01	TABLEAU	"RPRDHA" IN ('S1', 'S1A', 'S1AA01', 'S1AA01B01',
RCUSTSALE	TABLEAU	"__AP_CHECK_INT" = 1
RIOMATNR	TABLEAU	"__AP_CHECK_INT" = 1
RIOMATPL	TABLEAU	"__AP_CHECK_INT" = 1
RIOMATPL	TABLEAU	"__AP_CHECK_INT" = 1
RKUNAG	TABLEAU	"__AP_CHECK_INT" = 1
RPKUNW	TABLEAU	"__AP_CHECK_INT" = 1
RPKUNW	TABLEAU	"__AP_CHECK_INT" = 1
RPRDHA	TABLEAU	"RPRDHA" IN ('S1', 'S1A', 'S1AA01', 'S1AA01B01',
RPRDHA	TABLEAU	"__AP_CHECK_INT" = 1
RSAIII	TABLEAU	"RSAIII" IN ('1C20A', '1C25A')
RSDDS50	TABLEAU	"RPRDHA" IN ('S1', 'S1A', 'S1AA01', 'S1AA01B01',
RSDDS52	TABLEAU	"RPRDHA" IN ('S1', 'S1A', 'S1AA01', 'S1AA01B01',

Key Takeaways

- Performance
- Usability
- Extensibility
- Integration



Accounts Receivable Overview

Posting Year: (All)
Customer: (All)
Posting Month: (All)
Status: Open

Total Outstanding Amount

$5.25M Total Outstanding Amount

399 Outstanding Invoices

$2.68M Total Overdue Amount

51.05% Overdue Amount %

274 Overdue Invoices

Accounts Receivable Trend — Click on Month to See Details

Top Customers — Click on Customer to See Details

BW Best Practices

- Use the SAP HANA connector whenever it is possible
- Do not expose BW Content Data Models directly to Tableau
- Do not reuse already existing BEx Queries
- Use Calculation Views directly designed on InfoProvider and not on BEx Queries
- Limit the amount of columns as early as possible (BW or HANA layer)

Best Practices: HANA and Tableau

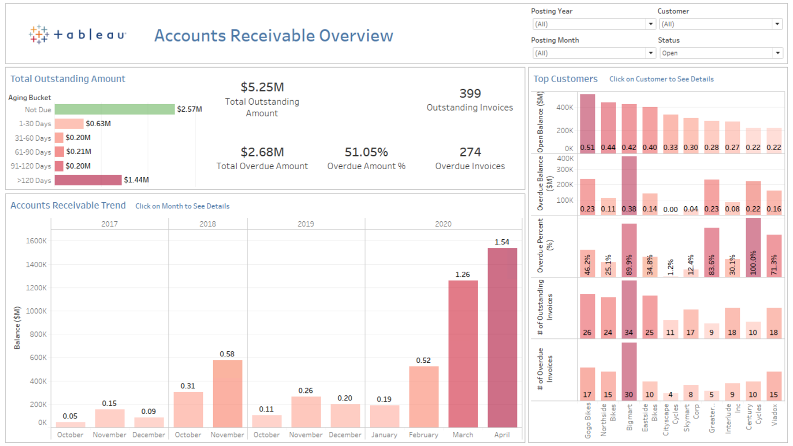
Tableau

- Limit the use of quick filters in Tableau - If quick filters: use wildcard filters
- Limit the amount of columns in the dashboards
- Guided analytics (filter with actions | no data without selection)
- Avoid Custom SQL in Tableau - if not required

HANA

- Persist data in SAP HANA (if performance is critical)
- Avoid switching calculation engines (Row, Column)
- Use HANA Star Joins when joining facts and dimension tables
- Avoid tables that are spread across multiple nodes
- **Do performance analysis while developing new dashboards**

S4/HANA Analytics via BW/4HANA & SAP HANA Views



Accounts Receivable Overview

Posting Year: (All)

Customer: (All)

Posting Month: (All)

Status: Open

Total Outstanding Amount

$5.25M Total Outstanding Amount

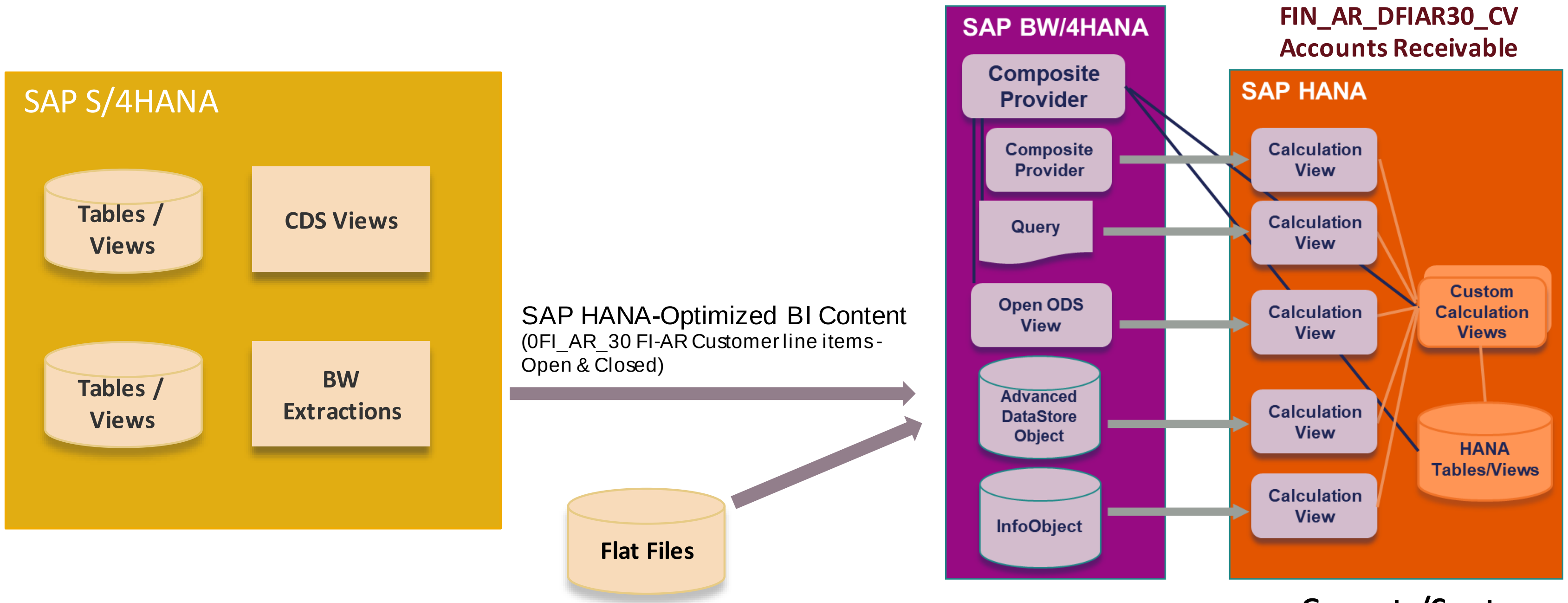
399 Outstanding Invoices

$2.68M Total Overdue Amount

51.05% Overdue Amount %

274 Overdue Invoices

SQL



SAP S/4HANA

Tables / Views

CDS Views

Tables / Views

BW Extractions

SAP HANA-Optimized BI Content
(0FI_AR_30 FI-AR Customer line items - Open & Closed)

Flat Files

Augment/Change Dataset

SAP BW/4HANA

Composite Provider

Composite Provider

Query

Open ODS View

Advanced DataStore Object

InfoObject

FIN_AR_DFIAR30_CV
Accounts Receivable

SAP HANA

Calculation View

Calculation View

Calculation View

Calculation View

Calculation View

Custom Calculation Views

HANA Tables/Views

Generate/Create Calculation Views

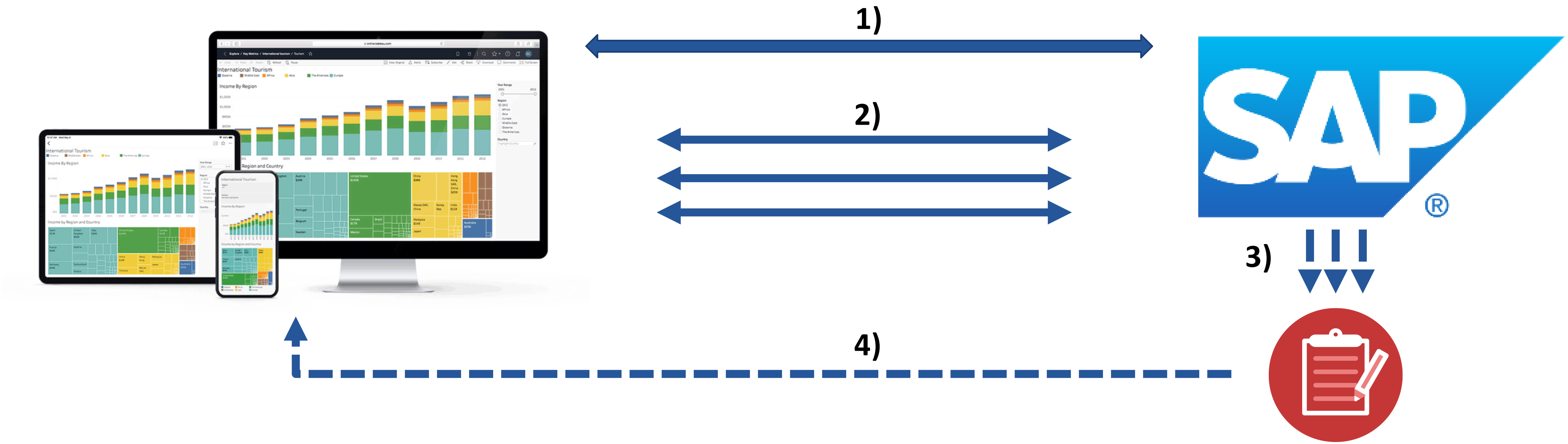
Performance and Workload Management for Tableau on SAP Hana

Simon Rech | Solution Engineer - Tableau

Frequent Performance / Workload Topics

- Identify workload that comes from Tableau vs. other solutions
- Identify queries with high memory consumptions / long runtimes
    - Which Tableau workbooks / users are causing these?
- Where is my runtime spent? Could it be HANA or a network issue?
- Investigate potential performance optimisations inside SAP HANA

SAP HANA Session Variables & Initial SQL



1) Authentication and assignment of session Variables
2) Workbook interactions cause queries to SAP HANA
3) SAP HANA uses the Session Variables information for statistics and traces
4) Tableau can visualise SAP HANA statistics and traces

SAP HANA Session Variables

- **SAP has pre-defined Session Variables that are used in Traces / Statistics**
  - Examples are APPLICATION, APPLICATIONVERSION, APPLICATIONUSER, APPLICATIONSOURCE, DEBUG_TOKEN, and TRACEPROFILE.
  - A complete list of the pre-defined session variables and their uses can be found in the “SAP HANA SQL and System Views Reference”.

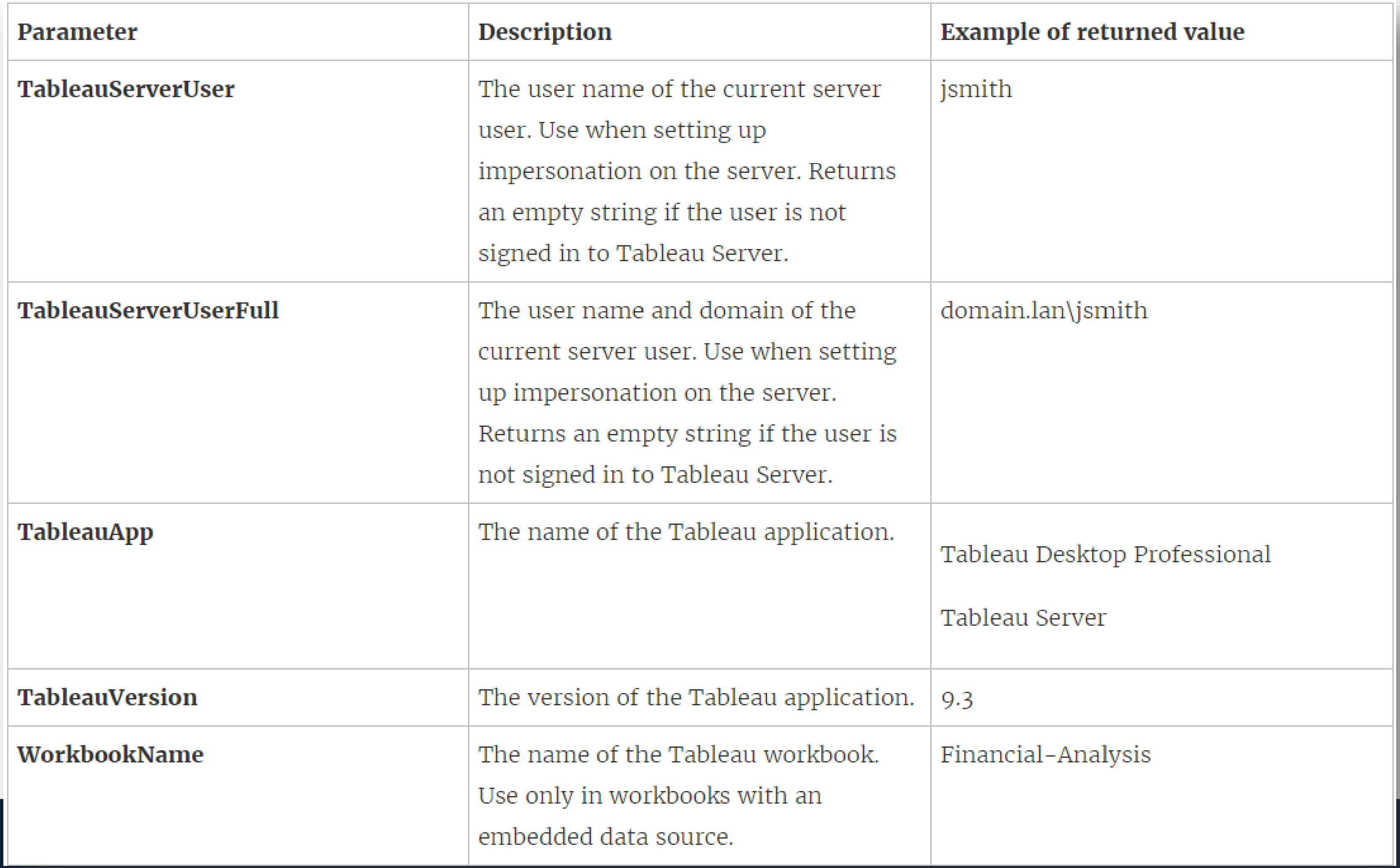
- **Some Session Variables are automatically filled by Tableau (e.g. APPLICATION) depending on the Tableau version**

- **Example for Initial SQL setting Session Variables**

```
SET SESSION 'APPLICATION' = [TableauApp];
SET SESSION 'APPLICATIONVERSION' = [TableauVersion];
SET SESSION 'APPLICATIONSOURCE' = [WorkbookName];
SET SESSION 'APPLICATIONUSER' = [TableauServerUser];
```

Initial SQL Parameters

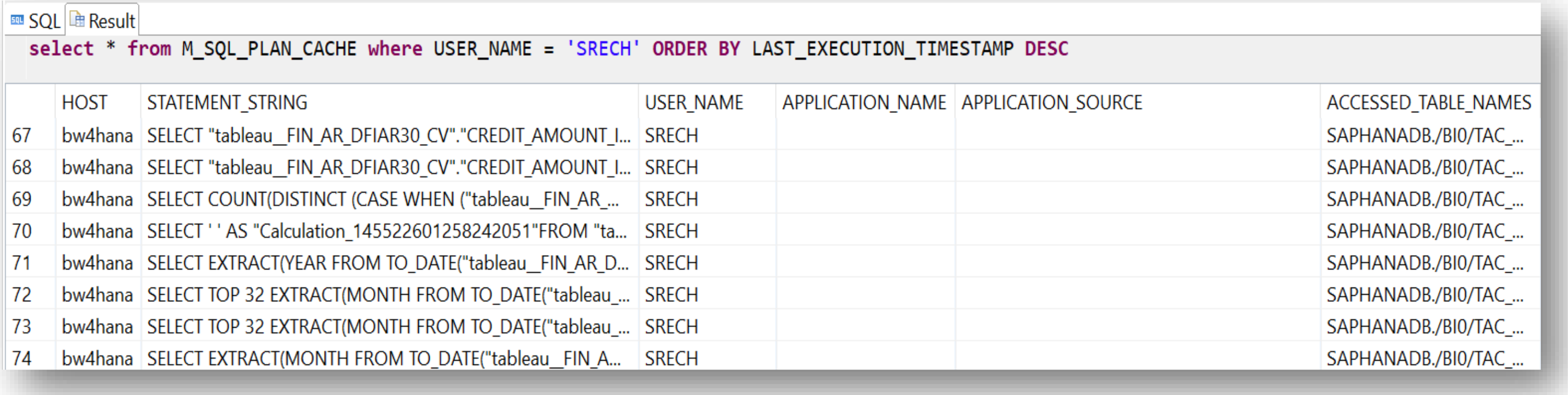
The following parameters can be used in Initial SQL to set Session Variables



Parameter	Description	Example of returned value
TableauServerUser	The user name of the current server user. Use when setting up impersonation on the server. Returns an empty string if the user is not signed in to Tableau Server.	jsmith
TableauServerUserFull	The user name and domain of the current server user. Use when setting up impersonation on the server. Returns an empty string if the user is not signed in to Tableau Server.	domain.lan\jsmith
TableauApp	The name of the Tableau application.	Tableau Desktop Professional Tableau Server
TableauVersion	The version of the Tableau application.	9.3
WorkbookName	The name of the Tableau workbook. Use only in workbooks with an embedded data source.	Financial-Analysis

Initial SQL Example

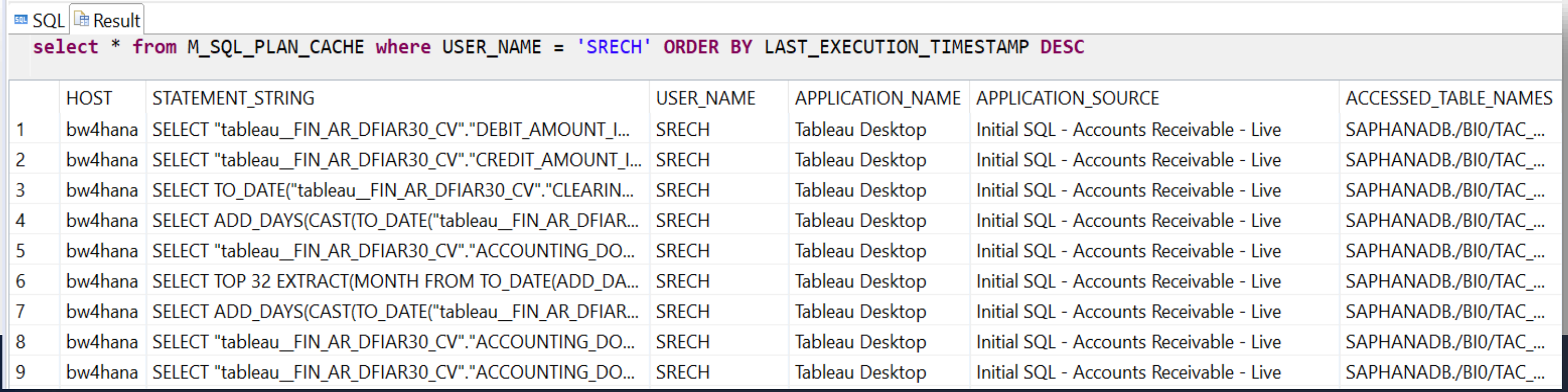
Without initial SQL



```
select * from M_SQL_PLAN_CACHE where USER_NAME = 'SRECH' ORDER BY LAST_EXECUTION_TIMESTAMP DESC
```

	HOST	STATEMENT_STRING	USER_NAME	APPLICATION_NAME	APPLICATION_SOURCE	ACCESSED_TABLE_NAMES
67	bw4hana	SELECT "tableau__FIN_AR_DFIAR30_CV"."CREDIT_AMOUNT_I...	SRECH			SAPHANADB./BI0/TAC_...
68	bw4hana	SELECT "tableau__FIN_AR_DFIAR30_CV"."CREDIT_AMOUNT_I...	SRECH			SAPHANADB./BI0/TAC_...
69	bw4hana	SELECT COUNT(DISTINCT (CASE WHEN ("tableau__FIN_AR_...	SRECH			SAPHANADB./BI0/TAC_...
70	bw4hana	SELECT ' ' AS "Calculation_145522601258242051"FROM "ta...	SRECH			SAPHANADB./BI0/TAC_...
71	bw4hana	SELECT EXTRACT(YEAR FROM TO_DATE("tableau__FIN_AR_D...	SRECH			SAPHANADB./BI0/TAC_...
72	bw4hana	SELECT TOP 32 EXTRACT(MONTH FROM TO_DATE("tableau_...	SRECH			SAPHANADB./BI0/TAC_...
73	bw4hana	SELECT TOP 32 EXTRACT(MONTH FROM TO_DATE("tableau_...	SRECH			SAPHANADB./BI0/TAC_...
74	bw4hana	SELECT EXTRACT(MONTH FROM TO_DATE("tableau__FIN_A...	SRECH			SAPHANADB./BI0/TAC_...

With initial SQL



```
select * from M_SQL_PLAN_CACHE where USER_NAME = 'SRECH' ORDER BY LAST_EXECUTION_TIMESTAMP DESC
```

	HOST	STATEMENT_STRING	USER_NAME	APPLICATION_NAME	APPLICATION_SOURCE	ACCESSED_TABLE_NAMES
1	bw4hana	SELECT "tableau__FIN_AR_DFIAR30_CV"."DEBIT_AMOUNT_I...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB./BI0/TAC_...
2	bw4hana	SELECT "tableau__FIN_AR_DFIAR30_CV"."CREDIT_AMOUNT_I...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB./BI0/TAC_...
3	bw4hana	SELECT TO_DATE("tableau__FIN_AR_DFIAR30_CV"."CLEARIN...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB./BI0/TAC_...
4	bw4hana	SELECT ADD_DAYS(CAST(TO_DATE("tableau__FIN_AR_DFIAR...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB./BI0/TAC_...
5	bw4hana	SELECT "tableau__FIN_AR_DFIAR30_CV"."ACCOUNTING_DO...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB./BI0/TAC_...
6	bw4hana	SELECT TOP 32 EXTRACT(MONTH FROM TO_DATE(ADD_DA...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB./BI0/TAC_...
7	bw4hana	SELECT ADD_DAYS(CAST(TO_DATE("tableau__FIN_AR_DFIAR...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB./BI0/TAC_...
8	bw4hana	SELECT "tableau__FIN_AR_DFIAR30_CV"."ACCOUNTING_DO...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB./BI0/TAC_...
9	bw4hana	SELECT "tableau__FIN_AR_DFIAR30_CV"."ACCOUNTING_DO...	SRECH	Tableau Desktop	Initial SQL - Accounts Receivable - Live	SAPHANADB./BI0/TAC_...

Relevant Traces / Caches in SAP HANA

- **Traces**

CONFIGURE OR RUN...	TO...
SQL TRACE	Collect information about all SQL statements executed on the index server
PLAN TRACE	Visualize the execution plans of SQL SELECT statements for in-depth query performance analysis
EXPENSIVE STATEMENTS TRACE	Record information about individual SQL statements whose execution time exceeded a configured threshold

(Source: https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/2.0.04/en-US/7e31247372fb4dd7b8c6bbac758b8c91.html)

- **SQL Plan Cache**
  - Overview of the SQL execution plans and their runtime statistics in the system

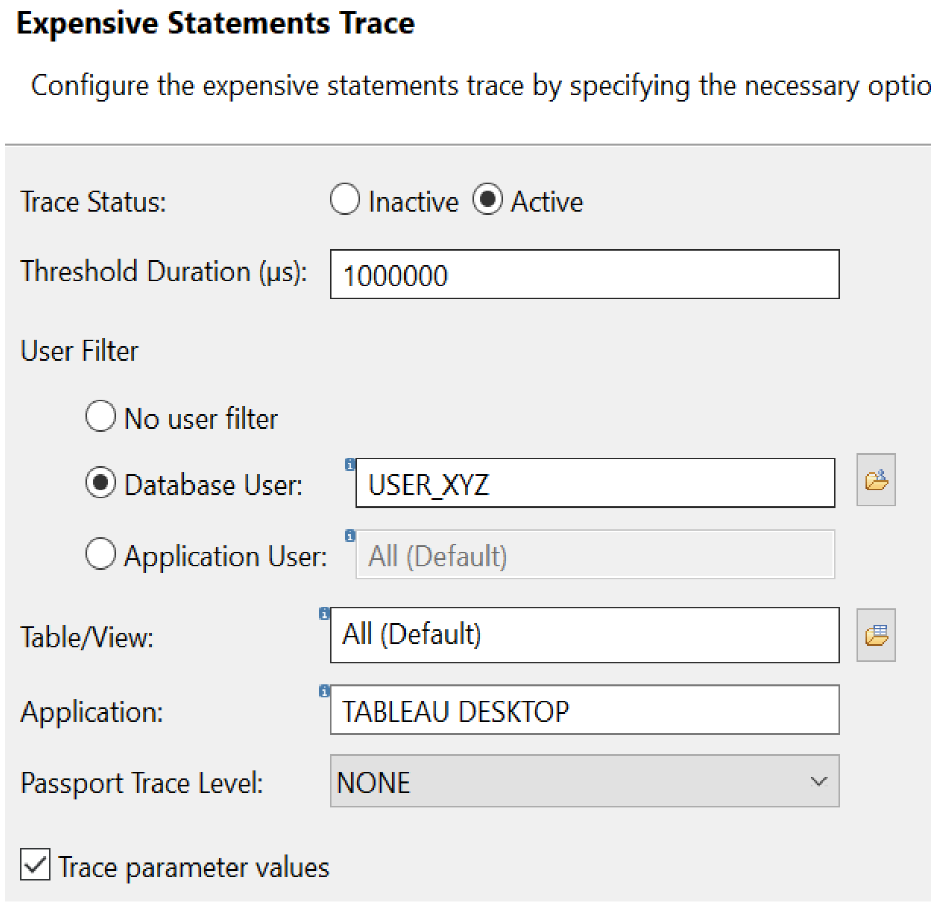
Expensive Statements Trace

- **Recommend to remain activated**
  - “Due to the significant added value and the small overhead (in case of reasonable thresholds) it is recommended to activate this trace on a permanent basis.” - SAP Note 2180165

- **Only tracking relevant queries (runtime threshold)**

- **Can be used for both specific performance analysis and long-term monitoring**

- **Stores results in a DB table –> easy to analyze**



Expensive Statements Trace

Configure the expensive statements trace by specifying the necessary optio

Trace Status: ○ Inactive ● Active

Threshold Duration (µs): 1000000

User Filter

○ No user filter

● Database User: USER_XYZ

○ Application User: All (Default)

Table/View: All (Default)

Application: TABLEAU DESKTOP

Passport Trace Level: NONE

☑ Trace parameter values

Expensive Statements Trace

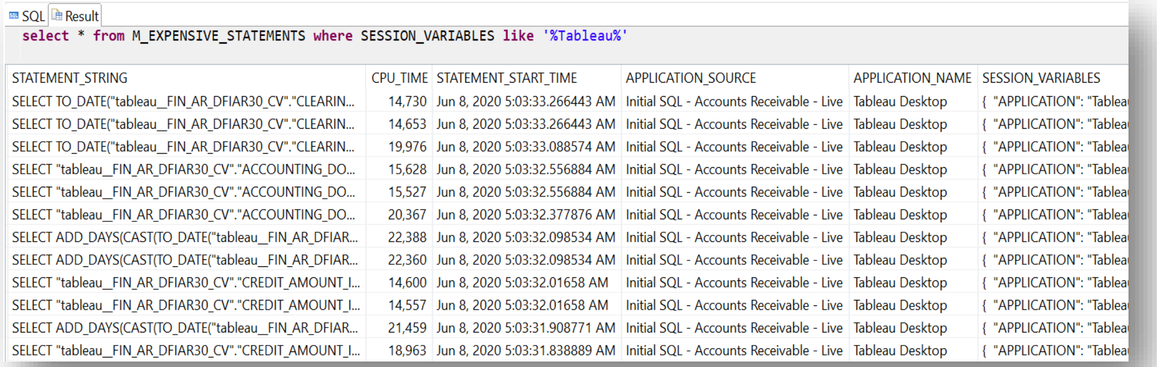
Tracked information a.o.:

- The type of operation during the statement execution (OPERATION)
- When the query started (START_TIME)
- How long the query took (DURATION_MICROSEC)
- The CPU time (in microseconds) to compute the statement (CPU_TIME)
- Name(s) of the objects accessed (OBJECT_NAME)
- The SQL statement (STATEMENT_STRING)
- Peak memory usage (in bytes) during the execution of the statement (MEMORY_SIZE

Expensive Statements Trace

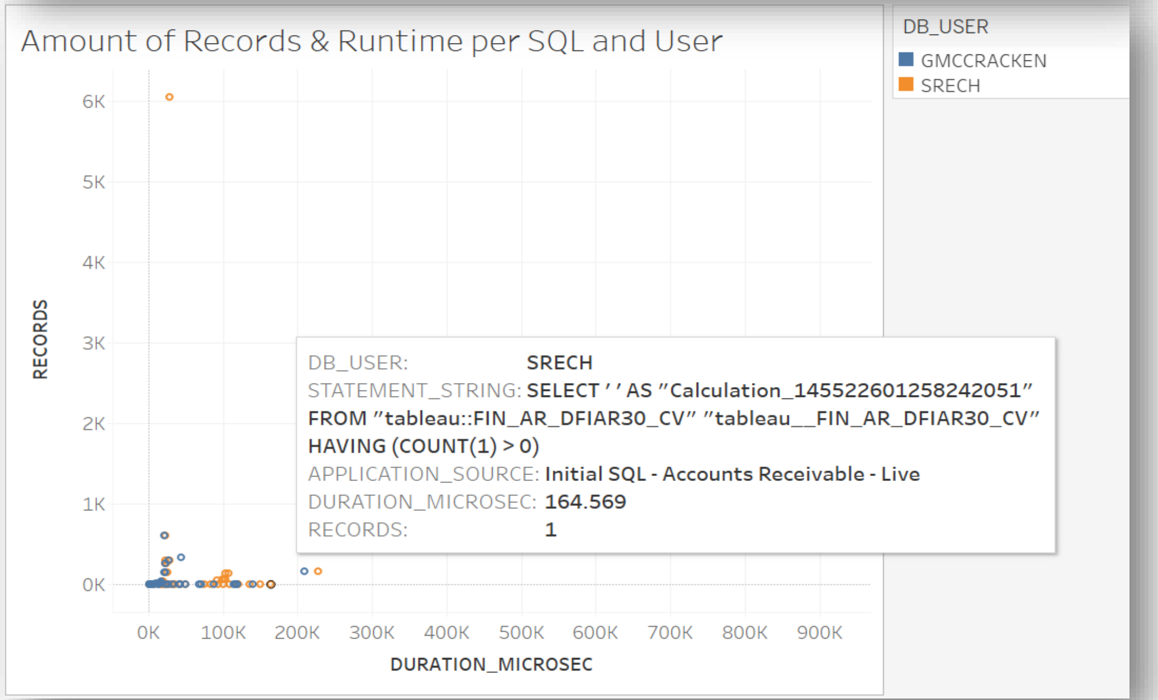
- **Results can be queried using SQL**
  - Select * from M_EXPENSIVE_STATEMENTS where APPLICATION_NAME like '%Tableau%'

- **Tableau can be used for analysing the results**
  - View 'M_EXPENSIVE_STATEMENTS' Schema 'SYS'



```
select * from M_EXPENSIVE_STATEMENTS where SESSION_VARIABLES like '%Tableau%'
```

STATEMENT_STRING	CPU_TIME	STATEMENT_START_TIME	APPLICATION_SOURCE	APPLICATION_NAME	SESSION_VARIABLES
SELECT TO_DATE("tableau__FIN_AR_DFIAR30_CV"."CLEARIN...	14,730	Jun 8, 2020 5:03:33.266443 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	{ "APPLICATION": "Tablea
SELECT TO_DATE("tableau__FIN_AR_DFIAR30_CV"."CLEARIN...	14,653	Jun 8, 2020 5:03:33.266443 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	{ "APPLICATION": "Tablea
SELECT TO_DATE("tableau__FIN_AR_DFIAR30_CV"."CLEARIN...	19,976	Jun 8, 2020 5:03:33.088574 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	{ "APPLICATION": "Tablea
SELECT "tableau__FIN_AR_DFIAR30_CV"."ACCOUNTING_DO...	15,628	Jun 8, 2020 5:03:32.556884 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	{ "APPLICATION": "Tablea
SELECT "tableau__FIN_AR_DFIAR30_CV"."ACCOUNTING_DO...	15,527	Jun 8, 2020 5:03:32.556884 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	{ "APPLICATION": "Tablea
SELECT "tableau__FIN_AR_DFIAR30_CV"."ACCOUNTING_DO...	20,367	Jun 8, 2020 5:03:32.377876 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	{ "APPLICATION": "Tablea
SELECT ADD_DAYS(CAST(TO_DATE("tableau__FIN_AR_DFIAR...	22,388	Jun 8, 2020 5:03:32.098534 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	{ "APPLICATION": "Tablea
SELECT ADD_DAYS(CAST(TO_DATE("tableau__FIN_AR_DFIAR...	22,360	Jun 8, 2020 5:03:32.098534 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	{ "APPLICATION": "Tablea
SELECT "tableau__FIN_AR_DFIAR30_CV"."CREDIT_AMOUNT_I...	14,600	Jun 8, 2020 5:03:32.01658 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	{ "APPLICATION": "Tablea
SELECT "tableau__FIN_AR_DFIAR30_CV"."CREDIT_AMOUNT_I...	14,557	Jun 8, 2020 5:03:32.01658 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	{ "APPLICATION": "Tablea
SELECT ADD_DAYS(CAST(TO_DATE("tableau__FIN_AR_DFIAR...	21,459	Jun 8, 2020 5:03:31.908771 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	{ "APPLICATION": "Tablea
SELECT "tableau__FIN_AR_DFIAR30_CV"."CREDIT_AMOUNT_I...	18,963	Jun 8, 2020 5:03:31.838889 AM	Initial SQL - Accounts Receivable - Live	Tableau Desktop	{ "APPLICATION": "Tablea



DB_USER: SRECH
STATEMENT_STRING: SELECT ' ' AS "Calculation_1455226012582420 51" FROM "tableau::FIN_AR_DFIAR30_CV" "tableau__FIN_AR_DFIAR30_CV" HAVING (COUNT(1) > 0)
APPLICATION_SOURCE: Initial SQL - Accounts Receivable - Live
DURATION_MICROSEC: 164.569
RECORDS: 1

SQL Plan Cache Analysis

- Can be queried like a SAP HANA view
- Overview of the SQL execution plans and their run time statistics
- No activation needed > always available
- Insights into frequently executed queries and slow queries
- But it was not designed for performance analysis:
  - Its entries change overtime and get overwritten
  - It might not be possible to find entries related to a dedicated query
  - Shows aggregate values (AVG, MIN, MAX) instead of values specific to one execution

➡ Suitable for understanding system workload and identifying problematic queries rather than performing specific performance analysis.

SQL Plan Cache Analysis

Tracked information a.o.:

- Dominant statements (TOTAL_EXECUTION_TIME)
- Long-running statements (AVG_EXECUTION_TIME / MAX _EXECUTION_TIME)
- Memory intensive statements (AVG_EXECUTION_MEMORY_SIZE / MAX _EXECUTION_MEMORY_SIZE)
- Frequently executed plans (EXECUTION_COUNT)
- Number of records returned (TOTAL_RESULT_RECORD_COUNT)

How to read the SQL Plan Cache

https://help.sap.com/viewer/bed8c14f9f024763b0777aa72b5436f6/2.0.00/en-US/c44c125ed4ae467a903cf4bb8527facb.html

Recommendations concerning the analysis of the SQL Plan

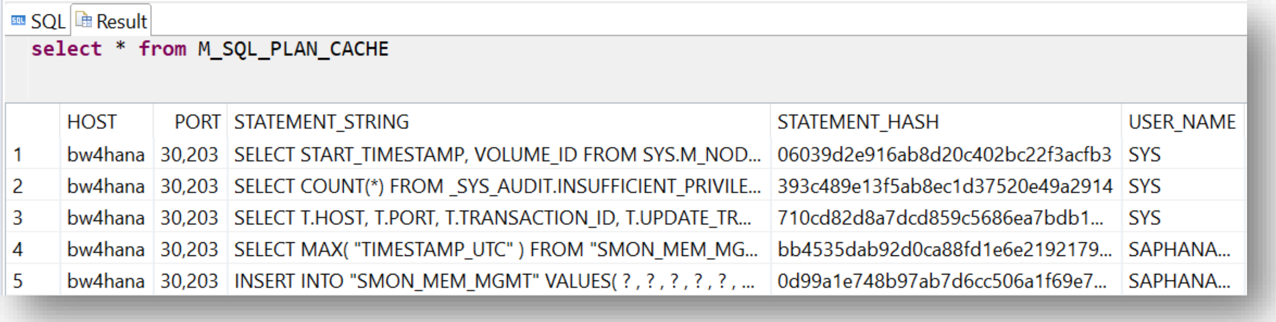
https://help.sap.com/viewer/bed8c14f9f024763b0777aa72b5436f6/2.0.00/en-US/a6c880a896cc41d7b02aff472d11b242.html

(SAP HANA Troubleshooting and Performance Analysis Guide)

SQL Plan Cache Analysis

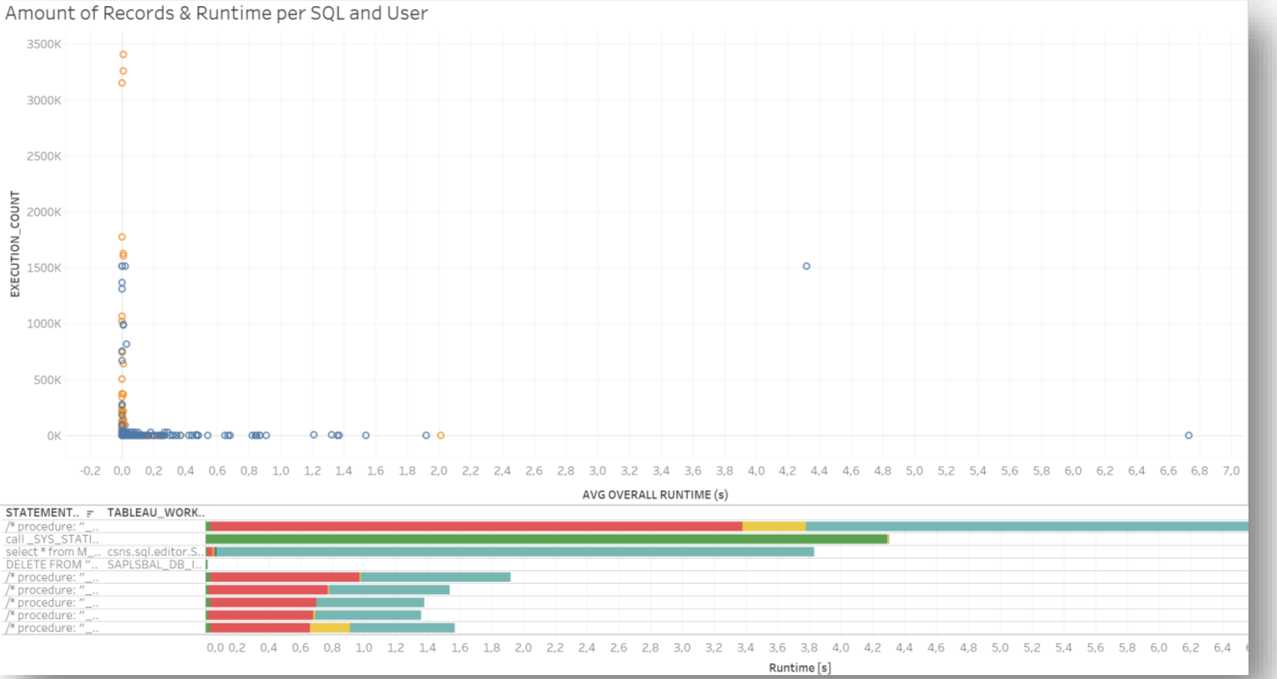
- **Results can be queried using SQL**
  - Select * from M_SQL_PLAN_CACHE

- **Tableau can be used for analysing the results**
  - View ‘M_SQL_PLAN_CACHE’ Schema ‘SYS;



```
select * from M_SQL_PLAN_CACHE
```

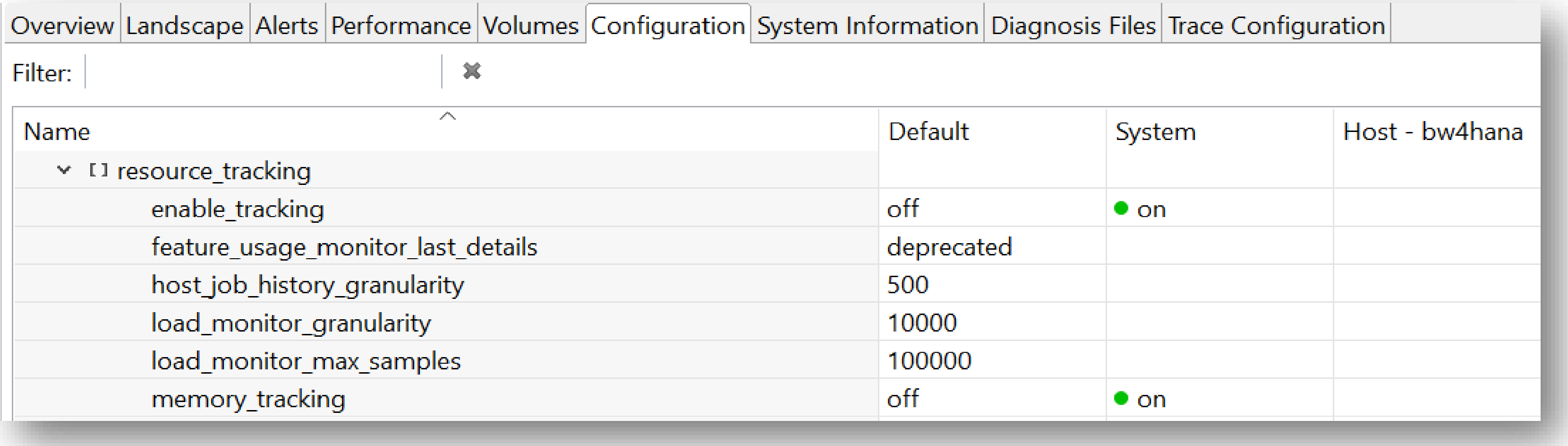
	HOST	PORT	STATEMENT_STRING	STATEMENT_HASH	USER_NAME
1	bw4hana	30,203	SELECT START_TIMESTAMP, VOLUME_ID FROM SYS.M_NOD...	06039d2e916ab8d20c402bc22f3acfb3	SYS
2	bw4hana	30,203	SELECT COUNT(*) FROM _SYS_AUDIT.INSUFFICIENT_PRIVILE...	393c489e13f5ab8ec1d37520e49a2914	SYS
3	bw4hana	30,203	SELECT T.HOST, T.PORT, T.TRANSACTION_ID, T.UPDATE_TR...	710cd82d8a7dcd859c5686ea7bdb1...	SYS
4	bw4hana	30,203	SELECT MAX("TIMESTAMP_UTC") FROM "SMON_MEM_MG...	bb4535dab92d0ca88fd1e6e2192179...	SAPHANA...
5	bw4hana	30,203	INSERT INTO "SMON_MEM_MGMT" VALUES(? , ? , ? , ? , ? , ...	0d99a1e748b97ab7d6cc506a1f69e7...	SAPHANA...


Amount of Records & Runtime per SQL and User

Enabling the tracking of memory usage

If the SAP HANA Memory consumption should be tracked e.g. for an Expensive Statements trace, the following parameters need to be set to 'on' in the global.ini file [resource_tracking] section:

- enable_tracking
- memory_tracking



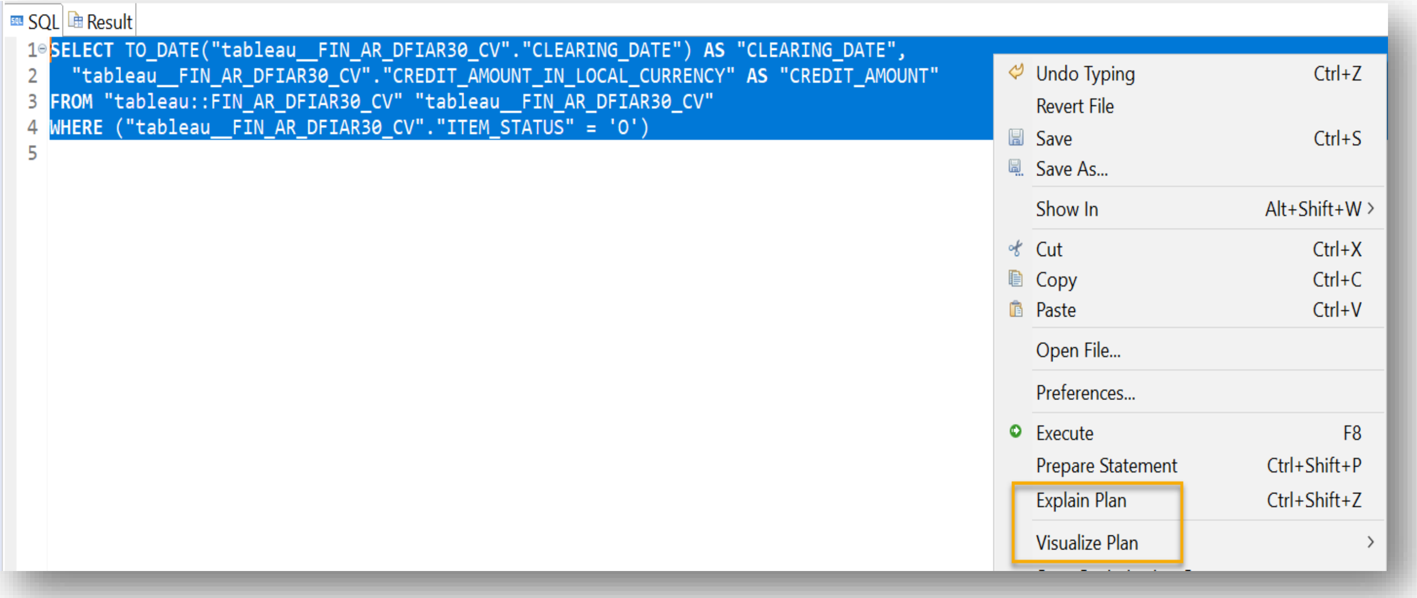
Name	Default	System	Host - bw4hana
resource_tracking			
enable_tracking	off	● on	
feature_usage_monitor_last_details	deprecated		
host_job_history_granularity	500		
load_monitor_granularity	10000		
load_monitor_max_samples	100000		
memory_tracking	off	● on	

Note

- In general, it is advisable to limit the maximum memory allocation per statement
- This can be achieved by setting a value for the [memorymanager] statement_memory_limit parameter in the global.ini file. A value of 10 would correspond to a limit of 10 GB.

SAP HANA Plan Explanation & Visualisation

- After analysing problematic queries they can be further analyzed using SAP HANA’s Plan Explanation and Visualization.
- Paste the query in the SQL Editor, select the whole statement and the right-click menu allows to ‘Explain Plan’ or ‘Visualize Plan’



```sql
SELECT TO_DATE("tableau__FIN_AR_DFIAR30_CV"."CLEARING_DATE") AS "CLEARING_DATE",
  "tableau__FIN_AR_DFIAR30_CV"."CREDIT_AMOUNT_IN_LOCAL_CURRENCY" AS "CREDIT_AMOUNT"
FROM "tableau::FIN_AR_DFIAR30_CV" "tableau__FIN_AR_DFIAR30_CV"
WHERE ("tableau__FIN_AR_DFIAR30_CV"."ITEM_STATUS" = 'O')
```

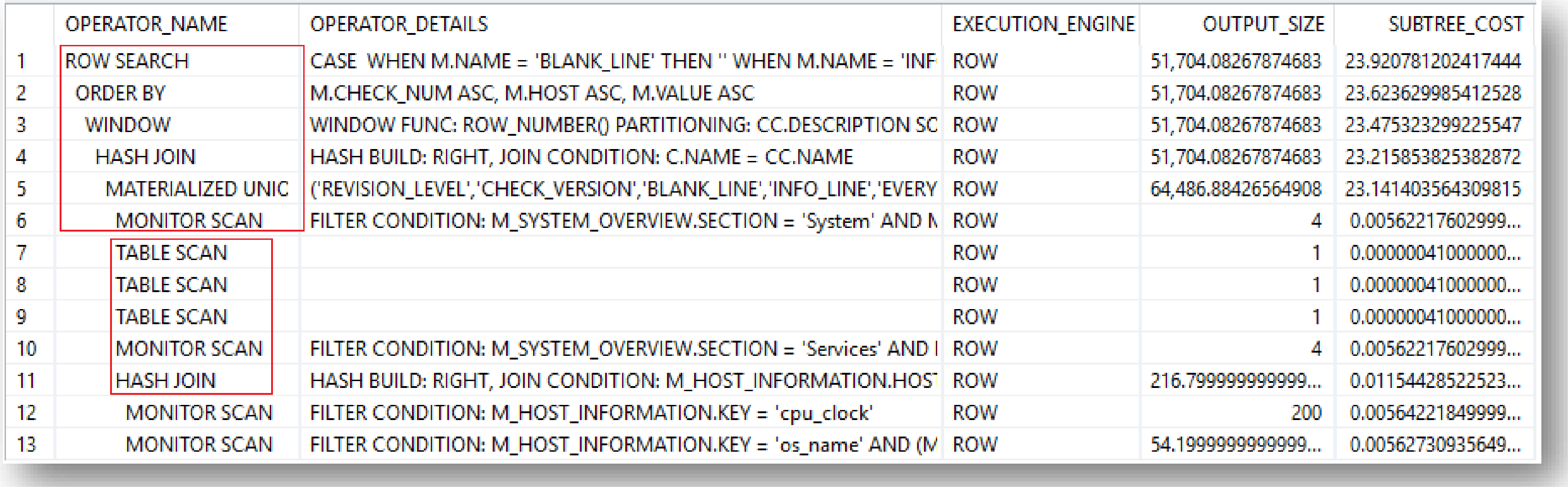
- The runtime information is based on estimates

SAP HANA Explanation

- Important information that can be contained

AREA	DETAIL
OPERATION DETAILS	The OPERATOR_NAME value shows the type of operation which was executed, such as joins, unions, aggregations and so on. Operations depend on the engine used - essentially row engine or column engine. Dependencies are shown by indentation - see examples below.
ENGINE	The type of engine where an operator is executed is shown in the EXECUTION_ENGINE column: ROW, COLUMN, OLAP, HEX, ESX.
TABLE DETAILS	Table details include table name, type, size, tables or objects which were accessed.
ESTIMATED COST	Cost values include the estimated output row count (OUTPUT_SIZE) and the estimated time in seconds (SUBTREE_COST).

- Example output

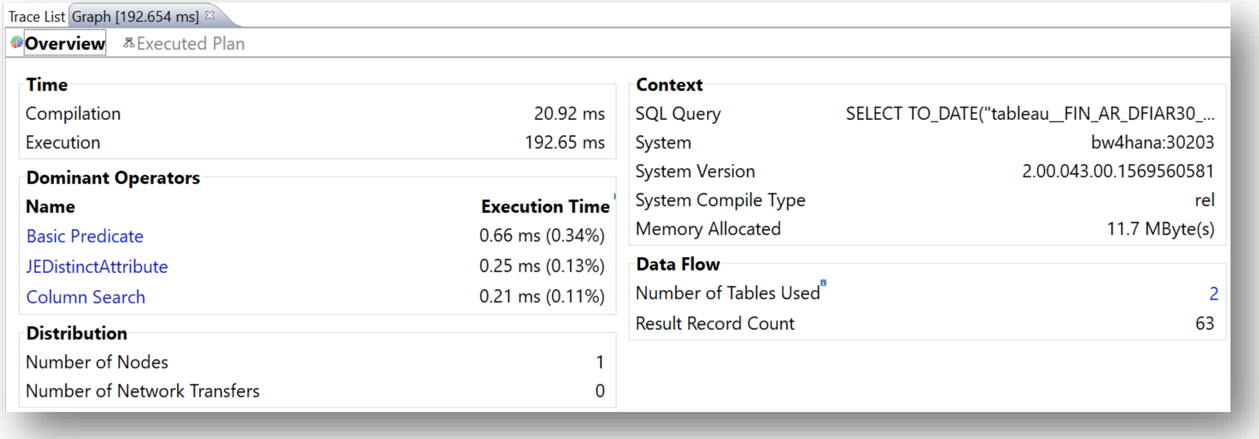


	OPERATOR_NAME	OPERATOR_DETAILS	EXECUTION_ENGINE	OUTPUT_SIZE	SUBTREE_COST
1	ROW SEARCH	CASE WHEN M.NAME = 'BLANK_LINE' THEN '' WHEN M.NAME = 'INF	ROW	51,704.08267874683	23.920781202417444
2	ORDER BY	M.CHECK_NUM ASC, M.HOST ASC, M.VALUE ASC	ROW	51,704.08267874683	23.623629985412528
3	WINDOW	WINDOW FUNC: ROW_NUMBER() PARTITIONING: CC.DESCRIPTION SC	ROW	51,704.08267874683	23.475323299225547
4	HASH JOIN	HASH BUILD: RIGHT, JOIN CONDITION: C.NAME = CC.NAME	ROW	51,704.08267874683	23.215853825382872
5	MATERIALIZED UNIC	('REVISION_LEVEL','CHECK_VERSION','BLANK_LINE','INFO_LINE','EVERY	ROW	64,486.88426564908	23.141403564309815
6	MONITOR SCAN	FILTER CONDITION: M_SYSTEM_OVERVIEW.SECTION = 'System' AND N	ROW	4	0.00562217602999...
7	TABLE SCAN		ROW	1	0.00000041000000...
8	TABLE SCAN		ROW	1	0.00000041000000...
9	TABLE SCAN		ROW	1	0.00000041000000...
10	MONITOR SCAN	FILTER CONDITION: M_SYSTEM_OVERVIEW.SECTION = 'Services' AND I	ROW	4	0.00562217602999...
11	HASH JOIN	HASH BUILD: RIGHT, JOIN CONDITION: M_HOST_INFORMATION.HOST	ROW	216.799999999999...	0.01154428522523...
12	MONITOR SCAN	FILTER CONDITION: M_HOST_INFORMATION.KEY = 'cpu_clock'	ROW	200	0.00564221849999...
13	MONITOR SCAN	FILTER CONDITION: M_HOST_INFORMATION.KEY = 'os_name' AND (M	ROW	54.1999999999999...	0.00562730935649...

Source: SAP HANA Troubleshooting and Performance Analysis Guide
https://help.sap.com/viewer/bed8c14f9f024763b0777aa72b5436f6/2.0.04/en-US/c0d42fd3bb571014a0688254f3de593f.html

SAP HANA PlanViz

Example Results



Trace List | Graph [192.654 ms] ☒

Overview | Executed Plan

Time

Compilation	20.92 ms
Execution	192.65 ms

Dominant Operators

Name	Execution Time
Basic Predicate	0.66 ms (0.34%)
JEDistinctAttribute	0.25 ms (0.13%)
Column Search	0.21 ms (0.11%)

Distribution

Number of Nodes	1
Number of Network Transfers	0

Context

SQL Query	SELECT TO_DATE("tableau__FIN_AR_DFIAR30_...
System	bw4hana:30203
System Version	2.00.043.00.1569560581
System Compile Type	rel
Memory Allocated	11.7 MByte(s)

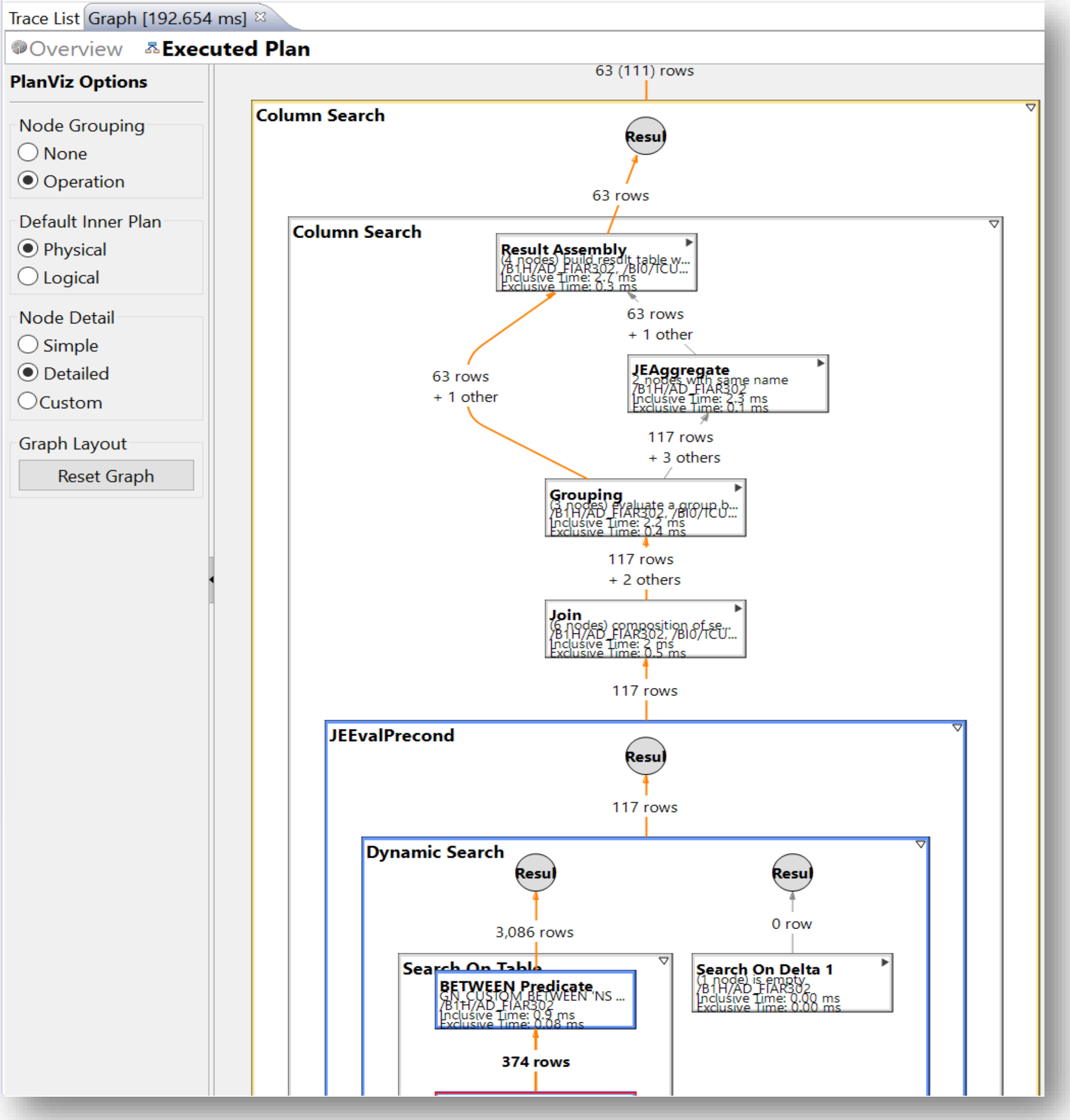
Data Flow

Number of Tables Used	2
Result Record Count	63

Runtime Differentiation

“Exclusive Time" = execution time of the node itself

"Inclusive Time" = execution time incl. descendants



Trace List | Graph [192.654 ms] ☒

Overview | **Executed Plan**

PlanViz Options

Node Grouping
○ None
● Operation

Default Inner Plan
● Physical
○ Logical

Node Detail
○ Simple
● Detailed
○ Custom

Graph Layout
Reset Graph

How to Create a Tableau Log Table in HANA

- Initial SQL can be used for maintaining a Tableau log that tracks Tableau usage
- Example

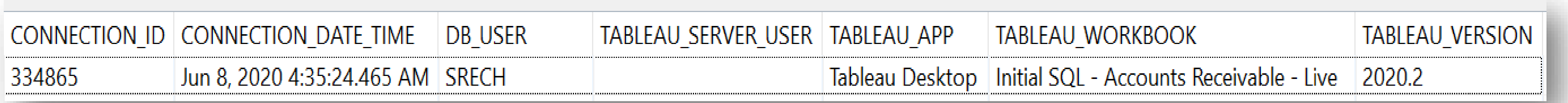
Log Table Creation

```
Create Column Table "SCHEMA_NAME"."TABLE_NAME"
        ("CONNECTION_ID" VARCHAR (100) null,
                "CONNECTION_DATE_TIME" TIMESTAMP null,
                "DB_USER" VARCHAR (30) null,
                "TABLEAU_SERVER_USER" VARCHAR (30) null,
                "TABLEAU_APP" VARCHAR (30) null,
                "TABLEAU_WORKBOOK" VARCHAR (100) null,
         "TABLEAU_VERSION" VARCHAR (20) null);
```

Initial SQL

```
INSERT INTO
"SCHEMA_NAME"."TABLE_NAME"
SELECT
CURRENT_CONNECTION,
CURRENT_TIMESTAMP,
SESSION_USER,
[TableauServerUser],
[TableauApp],
[WorkbookName],
[TableauVersion] FROM DUMMY;
```

- Log Table Entry



CONNECTION_ID	CONNECTION_DATE_TIME	DB_USER	TABLEAU_SERVER_USER	TABLEAU_APP	TABLEAU_WORKBOOK	TABLEAU_VERSION
334865	Jun 8, 2020 4:35:24.465 AM	SRECH		Tableau Desktop	Initial SQL - Accounts Receivable - Live	2020.2

Further Information

For more information about optimisation possibilities please refer to:

- SAP HANA Troubleshooting and Performance Analysis Guide
- The examples given in the SAP HANA SQL and System Views Reference
- SAP Note 2000002 - FAQ: SAP HANA SQL Optimization
- SAP Note 2142945 - FAQ: SAP HANA Hints
- SAP Note 2180165 - FAQ: SAP HANA Expensive Statements Trace
- SAP Note 2410208 - Collect Explain Plan of a Prepared Statement
- SAP released several blogs with instructions on how to use the PlanViz:
  - https://blogs.sap.com/2019/03/15/the-hana-planvisualizer-planviz-quick-and-easy/
  - https://blogs.sap.com/2018/04/29/analyzing-sql-execution-with-the-plan-visualizer-planviz/

Tableau and SAP Road Map

Recent Changes

- Multi-Node Support 10.4
- Parameter Binding & Query Cache (2018.3)
- Support for HANA 2.0 and HDI Containers (2019.3)
- HANA Leveled Hierarchies (2019.4)
- HANA 2.0 Certification (2020.2

Future

- Multi-Node Support 10.4
- 2020.3: HANA table functions
- 2020.4: Parameter Binding on by default, metadata performance, performance traceability
- TC 2020: Parent/Child Hierarchy Prototype then Private Alpha

Subject to Change.
HIGHLY CONFIDENTIAL.

Q&A

tableau®