

# SAP HANA 上的 Tableau:

性能跟踪和工作负载分析

# 目录

|   |    |
|---|----|
| <b>简介：目的和先决条件</b> .....                           | 3  |
| SAP HANA 运行时间跟踪和统计数据 <b>概览</b> .....              | 4  |
| <b>端到端单次执行跟踪：</b>                                 |    |
| 对 Tableau 和 SAP HANA 的性能统计数据合并 .....              | 6  |
| 激活 SAP HANA 的 SQL 跟踪 .....                        | 6  |
| 激活 Tableau 的性能记录器 .....                           | 7  |
| 分析来自 Tableau 的信息<br>性能工作簿和 HANA SQL 跟踪 .....      | 9  |
| <b>监测：</b>  |    |
| 找出长时间运行的和内存密集型 SQL 查询 .....                       | 12 |
| 通过 Tableau 的初始 SQL<br>增强 SAP HANA 的会话信息 .....     | 12 |
| 高成本语句跟踪：<br>监测长时间运行的查询和内存密集型查询 .....              | 15 |
| 分析 SAP HANA 高成本语句<br>在 Tableau 中跟踪结果 .....        | 18 |
| SQL Plan Cache 统计数据：确定优化候选对象<br>（长期运行和频繁执行） ..... | 18 |
| 在 Tableau 中分析 SAP HANA SQL Plan Cache .....       | 21 |
| <b>详细分析单个 SQL 查询：</b>                             |    |
| SAP HANA Explain Plan 和 Visualize Plan .....      | 22 |
| SAP HANA Plan Explanation .....                   | 22 |
| SAP HANA Plan Visualizer .....                    | 23 |
| <b>结论</b> .....                                   | 25 |
| <b>关于 Tableau</b> .....                           | 25 |



## 简介：目的和先决条件

性能和稳定性对于 Tableau 仪表板在 SAP HANA 实时数据源上的成功实施至关重要。为了实现最佳性能，在设计过程中分析仪表板运行时间并在此后持续监测系统中是否存在高成本查询非常重要。本文档旨在概述 SAP HANA 和 Tableau 中用于识别性能瓶颈的性能跟踪和监测功能。要确定合适的性能优化措施，就必须先了解这些功能。本文档不涉及如何设计高效 Tableau 仪表板的相关建议 — 此类建议可以在其他资源中找到，例如[设计高效工作簿](#)（白皮书）和[仪表板性能最佳做法](#)（演示录像）。

要激活本文档中提到的 SAP HANA 跟踪功能，必须获得足够的授权或 **SAP HANA 管理员** 的帮助。此外，必须有 **Tableau Creator**（例如分析师或仪表板设计师）角色才能收集 Tableau 测量的前端运行时间。结合使用 Tableau 和 SAP HANA 跟踪可提供完整的端到端运行时间分布。这些跟踪提供了找出性能瓶颈所需的信息，而性能瓶颈是性能优化的最佳候选对象。

虽然我们建议仅对单次执行进行详细的端到端跟踪（鉴于其资源开销），但您可以使用其他类型的跟踪来连续监测性能和工作负载，找出高成本查询。为了避免在高负载情况下出现瓶颈，您必须能够分析和找出高成本查询（CPU 运行时间和内存消耗）和主要查询（运行时间乘以执行计数）的来源。SAP HANA 可以根据配置收集这些统计数据，而 Tableau 可以连接到这些数据进行分析。

找出长时间运行的或内存密集型 SQL 语句后，可以使用 SAP HANA Plan Explanation 或 SAP HANA Plan Visualizer 等工具进一步分析其在 HANA 内部的处理情况。这些工具可提供详细的分步信息来说明查询结果是如何计算的，以及哪些处理步骤成本较高。了解这些情况有助于确定优化策略，例如更早地应用筛选器、在不同级别聚合等。

## SAP HANA 运行时间跟踪和统计数据概览

### 高成本语句跟踪

高成本语句跟踪可以捕获执行时间超过配置阈值的 SQL 语句的有关信息。阈值的输入单位是微秒 ( $\mu\text{s}$ )，即百万分之一秒。

跟踪功能有多种不同的使用方法。当阈值设置为 1 时，它几乎可以记录所有查询，但这样的设置应该仅在需要进行专门的性能分析时才使用，并且持续时间不能太长。但如果将阈值设置为更高的值，例如 5 秒 (5,000,000  $\mu\text{s}$ )，则这项功能可以永久保持激活状态。SAP Note 2180165 ([常见问题：SAP HANA 高成本语句跟踪](#)) 指出，“由于显著的附加价值和较小的开销（在设置合理阈值的情况下），建议永久激活这项跟踪”。

这项跟踪的另一个优点是它允许对其记录进行灵活分析，因为可以通过从 SAP HANA 视图（Schema“SYS”中的“M\_EXPENSIVE\_STATEMENTS”）中进行选择来检索其结果。这意味着可以使用 Tableau 的强大分析功能来分析 SAP HANA 中记录的高成本语句。

鉴于上述优点（可以永久保持激活状态并且可以使用 SQL 进行查询），我们决定在下一章更详细地介绍高成本语句跟踪。

### SQL 跟踪

除了通过永久保持激活状态的高成本语句跟踪功能来进行性能监测，SAP HANA 还提供了用于专门的性能分析的跟踪功能 — 在激活时捕获所有 SQL 查询及其运行时间统计数据（通常适用于专门的用户并持续较短的时间）。由于要提供此类极为详细的信息，SQL 跟踪会消耗更多资源（存储和 CPU），仅应在执行明确的短期性能分析（而不是长期监测）时激活。

SQL 跟踪会将它收集的信息写入 .py 文本文件而不是数据库表中。这种文件类型的优点是能够重放跟踪的数据库操作，缺点是很难读取和分析跟踪的性能信息。为了解决这个问题，SAP 提供了一个 python 工具 — SAP HANA SQL 跟踪分析器，用于聚合文件中的信息并简化其分析。

如需通过示例了解跟踪结果的外观以及 SQL 跟踪分析器工具的工作原理，请参阅来自 SAP HANA 学院的这个视频，该视频展示了如何使用 SAP HANA SQL 跟踪分析器。

## SQL Plan Cache

SQL Plan Cache 是一款很有用的工具，可以帮助您了解 SAP HANA 数据库的 SQL 处理。由于它不是跟踪，而是默认激活的缓存，因此 SQL Plan Cache 可以收集有价值的统计信息，而无需显式打开。

SQL Plan Cache 可提供系统中频繁执行的语句的概览，并跟踪运行时间统计数据。这些统计数据是汇总的，因此不会显示具体仪表板执行的运行时间信息，但它们可用于确定优化的候选对象 — 例如执行最频繁的查询、运行时间最长的查询等。SQL Plan Cache 可以作为 SAP HANA 视图来进行查询，因此可以使用 Tableau 来进行分析。

## 启用 SAP HANA 内存跟踪

除了性能 KPI，一些 SAP HANA 跟踪还允许捕获 SQL 语句的内存消耗信息。如果要分析内存不足的情况，或尝试找出大量消耗数据库内存的用户，此信息可能会非常有帮助。要启用内存使用跟踪，必须将 global.ini 文件 [resource\_tracking] 部分中的以下两个参数设置为“on”（打开）：

- enable\_tracking
- memory\_tracking

| Overview                           | Landscape  | Alerts | Performance    | Volumes | Configuration | System Information | Diagnosis Files | Trace Configuration |  |
|------------------------------------|------------|--------|----------------|---------|---------------|--------------------|-----------------|---------------------|--|
| Filter:   ✕                        |            |        |                |         |               |                    |                 |                     |  |
| Name                               | Default    | System | Host - bw4hana |         |               |                    |                 |                     |  |
| ▼ [1] 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   |                |         |               |                    |                 |                     |  |

## 端到端单次执行跟踪： 对 Tableau 和 SAP HANA 的性能统计数据合并

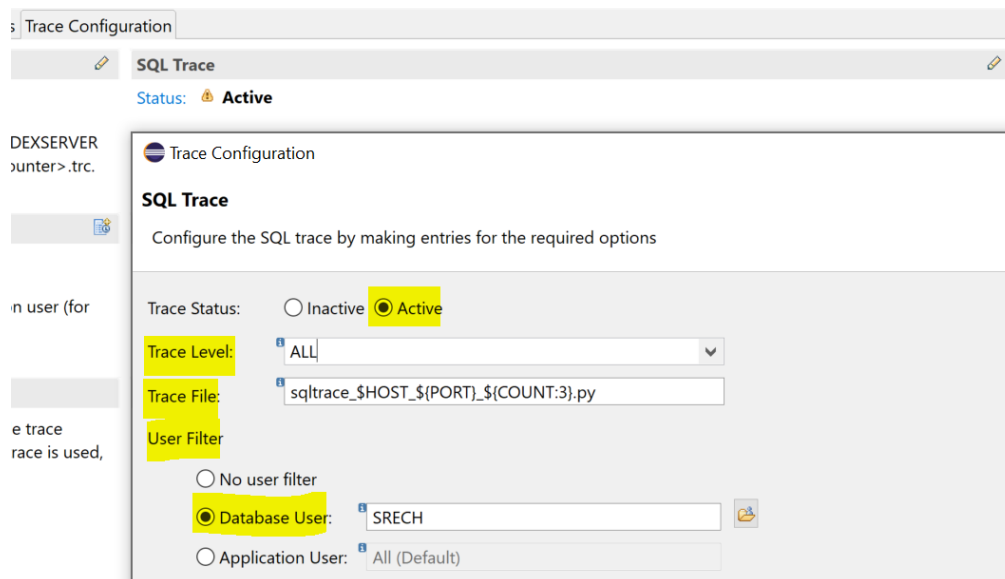
端到端分析需要同时在 Tableau 端和 SAP HANA 端临时激活跟踪，只适用于单次执行，而不应用于持续监测。

要了解 Tableau 工作簿的运行时间分布，建议同时在 Tableau 端和数据库端（端到端）跟踪性能。这样便能够得出相关结论，了解哪一个层（Tableau、网络、HANA）消耗了大多数运行时间，以及哪些步骤会减慢整体执行速度。这将有助于找出那些最适合进行性能优化的步骤。

### 激活 SAP HANA 的 SQL 跟踪

第一步，需要为执行跟踪的数据库用户激活 SAP HANA 的 SQL 跟踪。这样可以确保捕获来自该用户的所有传入 SQL 查询，包括其运行时间信息。激活此跟踪的一种方法是在 SAP HANA Studio 的“Administration”（管理）>“Trace Configuration”（跟踪配置）部分中进行设置。

要启用跟踪，请将“Trace Status”（跟踪状态）设置为“Active”（活动）并使用跟踪级别“ALL”（全部）或“ALL\_WITH\_RESULTS”（全部并附带结果）。您可以在“Trace File”（跟踪文件）字段指定生成的跟踪文件的名称。此外，“Database User”（数据库用户）应筛选为执行 Tableau 工作簿相关操作的用户。



有关 SQL 跟踪配置的详细说明，请参阅 SAP Note 2031647（如何在 SAP HANA Studio 中启用 SQL 跟踪）。

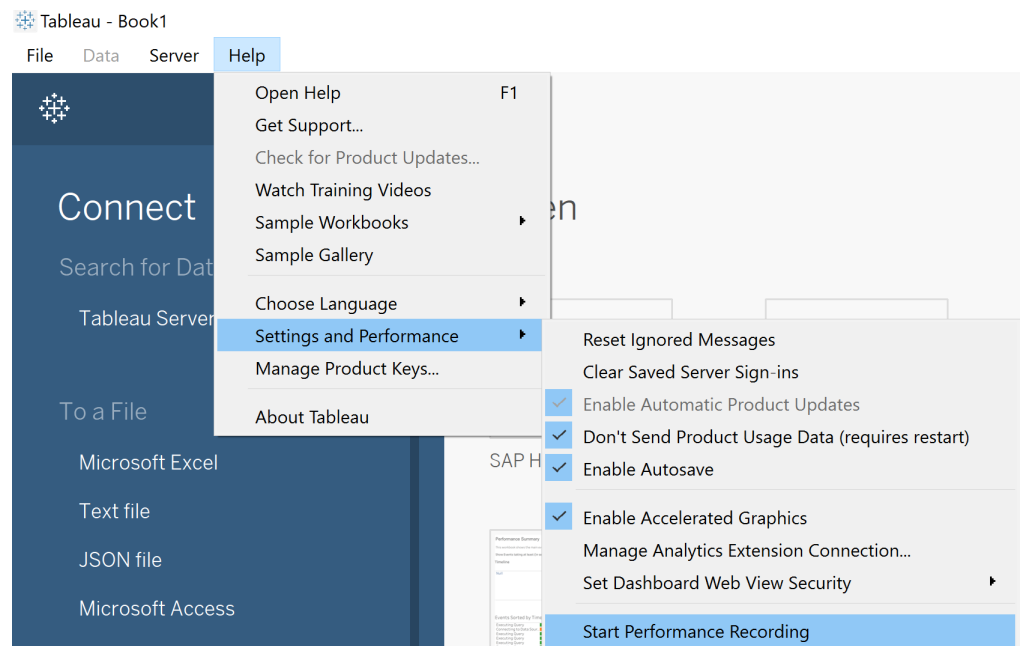
## 激活 Tableau 的性能记录器

要在 Tableau 端记录运行时间分布，可以使用 Tableau 性能跟踪。

### Tableau Desktop

使用 Tableau Desktop 时，通过单击以下选项启动性能跟踪：

“Help”（帮助）>“Settings and Performance”（设置和性能）>  
“Start Performance Recording”（启动性能记录）



现在，在 Tableau 中执行的每个步骤都将记录在性能跟踪之中。

要停止记录并查看包含记录会话结果的临时工作簿，请再次单击同一个按钮（该按钮的名称现在是“停止性能记录”）：

“Help”（帮助）>“Settings and Performance”（设置和性能）>  
“Stop Performance Recording”（停止性能记录）

上述操作将生成包含跟踪信息的 Tableau 工作簿。可以通过选择“文件”>“另存为”来保存此工作簿，以供将来参考。

有关激活 Tableau Desktop 跟踪的详细说明，请参阅 Tableau 帮助中的[记录和分析工作簿性能](#)。

## Tableau Server

使用 Tableau Server 时，性能跟踪可通过在视图 URL 末尾添加“:record\_performance=yes&”来启动。

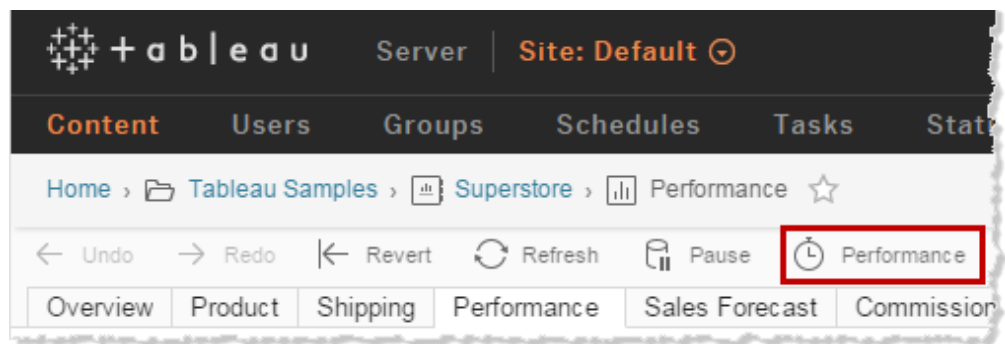
### 示例：查看 URL

```
http://10.32.139.22/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?:iid=1
```

### 示例：启用了性能跟踪的视图 URL：

```
http://10.32.139.22/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?:record_performance=yes&:iid=1
```

要从 Tableau Server 查看性能记录，请单击“Performance”（性能）按钮。



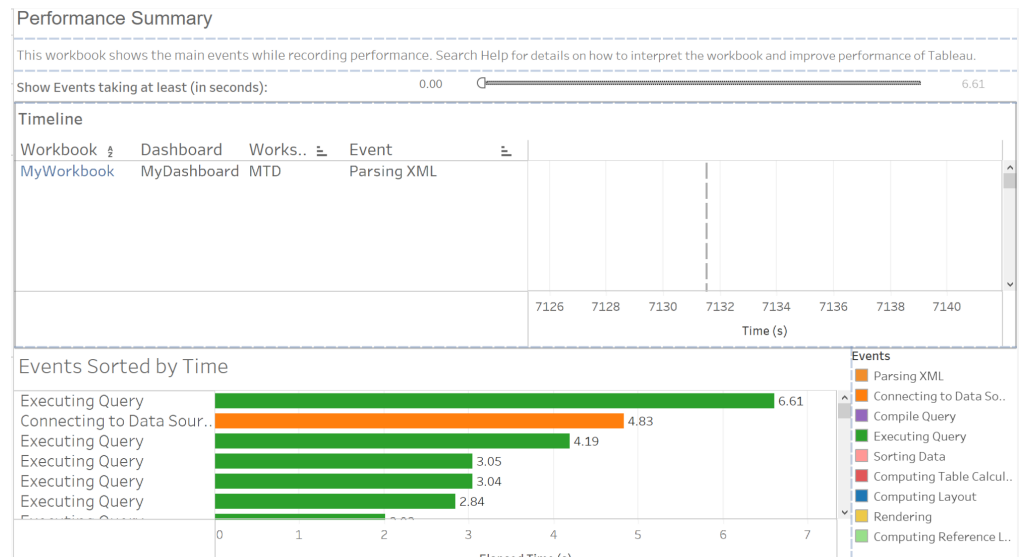
要停止性能跟踪，您可以单击到其他页面或从 URL 中删除“:record\_performance=yes”。有关激活 Tableau Server 跟踪的详细说明，请参阅 Tableau 帮助中的[创建性能记录](#)。



## 分析来自 Tableau 性能工作簿的信息和 HANA SQL 跟踪

Tableau 性能跟踪的结果以 Tableau 工作簿的形式呈现。

“Performance Summary”（性能摘要）仪表板概述了执行的各个步骤的时间线，底部的图表按运行时间对事件进行排序。

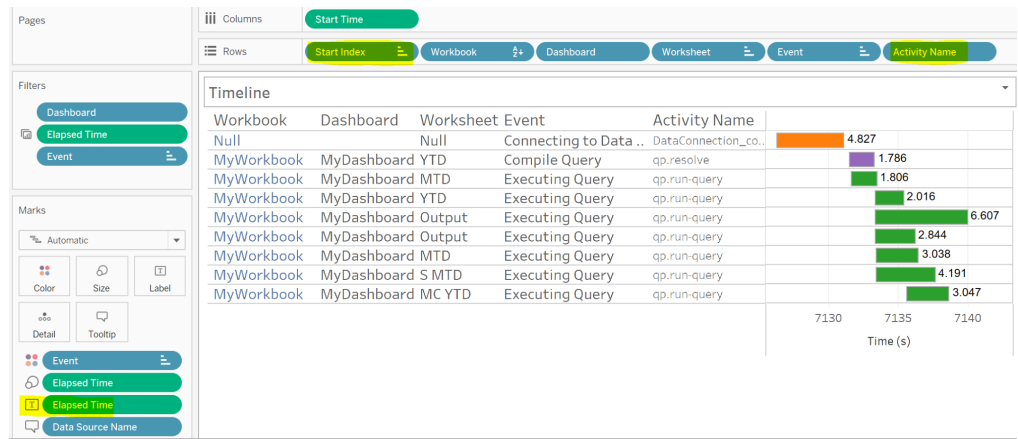


为了专注于最重要的事件，建议通过筛选出至少需要 0.5 秒的事件来消除噪音。

Show Events taking at least (in seconds): 0.50

可以调整时间线工作表来显示更多信息。例如，“Elapsed Time”（经历的时间）可以添加为标签，“Start Index”（开始索引）可以拖放为行中的第一个字段（按开始时间对操作进行排序），并且可以将“Activity Name”（活动名称）添加到行中。

结果将类似于：



在这个例子中，大部分运行时间都用于执行查询。“Executing Query”（执行查询）是 Tableau 等待来自 SAP HANA 的查询响应的的时间，包括通过网络传输响应的的时间。

要找出与“Executing Query”（执行查询）事件对应的完整 SQL 语句，可以在性能摘要中选择该事件。此操作会筛选查询工作表以显示来自该事件的 SQL 命令。

Tableau 工作簿工作表的空间太小，无法显示语句，但可以选择语句并将其复制到文本编辑器中以进行完整显示。

```
SELE..

*Untitled - Notepad
File Edit Format View Help
Command
"SELECT ""BI.PLANT/PLANT_CV_PLNTMNG"".""BI_PLANT__T"" AS ""BI_PLANT__T""
FROM ""_SYS_BIC"".""BI.PLANT/PLANT_CV_PLNTMNG"" ""BI.PLANT/PLANT_CV_PLNTMNG""
GROUP BY ""BI.PLANT/PLANT_CV_PLNTMNG"".""BI_PLANT__T""
ORDER BY 1 ASC"
Ln 6, Col 1 100% Windows (CRLF) UTF-8
```

确定哪些工作表和查询消耗了最多的运行时间后，可以将这些信息与 HANA SQL 跟踪相结合。

对跟踪文件执行 SAP 的 SQL 跟踪分析器（参见 SAP Note 2412519 [常见问题：SAP HANA SQL 跟踪分析器](#)）后，输出将显示 HANA 中每个 SQL 查询的运行时间信息，包括执行、编译、光标和获取运行时间。这些信息以微秒为单位显示，因此除以 1,000,000 即可将其转换为秒。

|   | A                            | B                  | C      | D     | H                      | L                      |
|---|------------------------------|--------------------|--------|-------|------------------------|------------------------|
| 1 | STATEMENT_STRING             | STATEMENT_HASH     | SCHEMA | COUNT | TOTAL_EXECUTE_DURATION | TOTAL_COMPILE_DURATION |
| 2 | SELECT "t2"."X_measure_4" AS | 270566e01b431ee3c6 | SRECH  | 1     | 18900599               | 241                    |
| 3 | SELECT "t2"."X_measure_6" AS | 2509cbf5a7db02d7a6 | SRECH  | 1     | 18740081               | 432                    |
| 4 | SELECT "t2"."X_measure_4" AS | 2925ec907324f130a9 | SRECH  | 1     | 15357529               | 171                    |

将 HANA 中的 SQL 语句运行时间与 Tableau 中相应的“Executing Query”（执行查询）运行时间进行比较可以提供有价值的见解。如果二者差异很大，原因可能与网速或 HANA 服务器和 Tableau 前端之间的防火墙有关。

如果某个 SQL 查询在 HANA 中的运行时间很长，有多种方法可以解决这个问题。例如：

- 检查是否可以通过重新设计 Tableau 工作表来简化查询。
- 检查 HANA 数据模型是否可以针对查询进行优化（请参阅本文档的“HANA Plan Explanation 和 HANA Plan Visualization”部分）。

如果 SAP HANA 中的 SQL 执行计划编译时间很长，甚至比查询执行时间更长，则表明应该激活 Tableau 的[绑定变量功能](#)。此功能可以提高 SQL Plan Cache 的命中率，从而减少编译 SQL 执行计划的需要。

如果有一个长时间运行的 SQL 查询以 'HAVING (COUNT(1) > 0)' 结尾，则可以通过一个 TDC 设置 ('CAP\_QUERY\_HAVING\_REQUIRES\_GROUP\_BY') 在启用时将此条件替换为 'GROUP BY' 语句。在某些用例中，此设置显著减少了 SQL 查询运行时间。

这些示例说明了我们通过对 Tableau 跟踪和 SAP HANA 跟踪进行结合分析能够发现哪些见解。

## 监测：

### 找出长时间运行的和内存密集型 SQL 查询

SAP HANA 提供多种跟踪、日志和统计数据，其中包含非常有助于进行性能分析和监测的信息。例如，在出现高负载资源瓶颈时，它们可用于找出性能优化的候选对象和/或执行根本原因分析。

遗憾的是，我们可能很难确定这些日志中的哪些信息与 Tableau 查询相关，甚至无法确定它们是否与特定的 Tableau 工作簿或工作表相关。

如果您想跟踪 Tableau 查询的这些信息，Tableau 的初始 SQL 功能就能派上用场。通过使用 SAP HANA 会话变量，此功能让我们有更多方法来分析 SAP HANA 中的 Tableau 工作负载和性能。结合来自 SAP HANA 日志的信息（例如高成本语句跟踪或 SQL Plan Cache），这项功能为 SAP HANA 管理员和 Tableau 工作簿设计者提供了更多选择。

### 通过 Tableau 的初始 SQL 功能来增强 SAP HANA 的会话信息

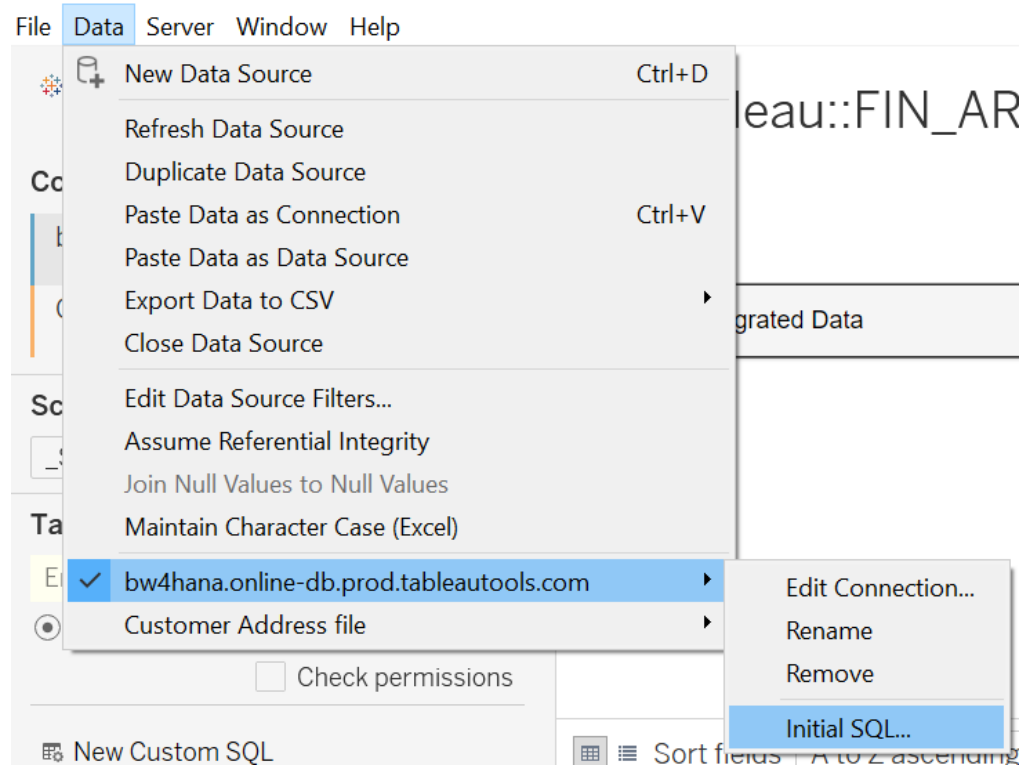
当与 SAP HANA 建立新连接时，SAP HANA Session Management（SAP HANA 会话管理）中将建立一个新会话。它会保存会话来源信息和技术信息，SAP HANA 使用这些信息进行跟踪并收集统计数据。

预定义会话变量的示例包括 APPLICATION、APPLICATIONVERSION、APPLICATIONUSER 和 APPLICATIONSOURCE。要查看预定义会话变量的完整列表及其用途，请参阅 [SAP HANA SQL 和系统视图参考](#)。

当 Tableau 连接到 SAP HANA 时，Tableau 会自动分配一些会话变量（例如 APPLICATION 和 APPLICATIONVERSION）。此功能有助于将 Tableau 查询和/或工作负载与其他应用程序/工具分开。Tableau 的初始 SQL 功能可用于分配额外的会话变量来存储信息；例如 Tableau Workbook Name（Tableau 工作簿名称）或 Tableau Server User（Tableau Server 用户）。初始 SQL 是在数据源级别定义的一组 SQL 命令，将在建立数据库连接时执行，例如打开工作簿或刷新数据时。

在此示例中，我们将使用 APPLICATIONSOURCE 提交建立连接的 Tableau 工作簿的名称，并使用 APPLICATIONUSER 来标识运行查询的 Tableau 用户。

设置初始 SQL 时，需要打开“数据源”选项卡，还需要在“数据”菜单中选择 SAP HANA 数据源：



设置或覆盖会话变量的语法如下：

```
SET [SESSION] <variable_string_literal> = <value_string_literal>
```

例如，可以在初始 SQL 中使用以下命令将 Tableau 应用程序、版本和工作簿名称存储在 SAP HANA 会话变量中：

```
SET SESSION 'APPLICATIONSOURCE' = [WorkbookName];
```

```
SET SESSION 'APPLICATIONUSER' = [TableauServerUser];
```

要检查初始 SQL 是否提供了您期望的信息，可以通过查询 M\_SESSION\_CONTEXT 系统视图（例如，'select \* from M\_SESSION\_CONTEXT'）来检索会话变量。

```
select * from M_SESSION_CONTEXT
```

| HOST    | PORT   | CONNECTION_ID | KEY                | VALUE                                    |
|---------|--------|---------------|--------------------|--|
| bw4hana | 30,203 | 326,188       | APPLICATION        | Tableau Desktop                          |
| bw4hana | 30,203 | 326,188       | XS_APPLICATIONUSER | SRECH                                    |
| bw4hana | 30,203 | 326,188       | PROTOCOL_VERSION   | 4.1 (1, 1)                               |
| bw4hana | 30,203 | 326,188       | APPLICATIONVERSION | 2020.2                                   |
| bw4hana | 30,203 | 326,188       | APPLICATIONSOURCE  | Initial SQL - Accounts Receivable - Live |

这样，会话监测或高成本语句跟踪等 SAP HANA 管理工具将显示此前不可用的关于会话来源的信息。

### a) 会话监测:

Overview | Landscape | Alerts | Performance | Volumes | Configuration | System Information | Diagnosis Files | Trace Configuration

Threads | Sessions | Blocked Transactions | SQL Plan Cache | Expensive Statements | Trace | Job Progress | Load

Summary

Enter your filter Visible rows: 13/13

| s | Application     | Application Source                       | Application Version | Application User | Database User | Client Host | Client IP    | Client Process ID |
|---|-----------------|--|---------------------|------------------|---------------|-------------|--------------|-------------------|
| 0 | Tableau Desktop | Initial SQL - Accounts Receivable - Live | 2020.2              | ?                | SRECH         | SRECH-LAP   | 192.195.4... | 9,796             |
| 0 | Tableau Desktop | Initial SQL - Accounts Receivable - Live | 2020.2              | ?                | SRECH         | SRECH-LAP   | 192.195.4... | 25,192            |
| 0 | Tableau Desktop | Initial SQL - Accounts Receivable - Live | 2020.2              | ?                | SRECH         | SRECH-LAP   | 192.195.4... | 14,052            |

### b) 日志和跟踪:

SQL Result

```
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_L...  | SRECH     | Tableau Desktop  | Initial SQL - Accounts Receivable - Live | SAPHANADB,/BIO/TAC... |
| 2 | bw4hana | SELECT *tableau_FIN_AR_DFIAR30_CV"."CREDIT_AMOUNT_L... | SRECH     | Tableau Desktop  | Initial SQL - Accounts Receivable - Live | SAPHANADB,/BIO/TAC... |
| 3 | bw4hana | SELECT TO_DATE(*tableau_FIN_AR_DFIAR30_CV"."CLEARIN... | SRECH     | Tableau Desktop  | Initial SQL - Accounts Receivable - Live | SAPHANADB,/BIO/TAC... |
| 4 | bw4hana | SELECT ADD_DAYS(CAST(TO_DATE(*tableau_FIN_AR_DFIAR...  | SRECH     | Tableau Desktop  | Initial SQL - Accounts Receivable - Live | SAPHANADB,/BIO/TAC... |
| 5 | bw4hana | SELECT *tableau_FIN_AR_DFIAR30_CV"."ACCOUNTING_DO...   | SRECH     | Tableau Desktop  | Initial SQL - Accounts Receivable - Live | SAPHANADB,/BIO/TAC... |
| 6 | bw4hana | SELECT TOP 32 EXTRACT(MONTH FROM TO_DATE(ADD_DA...     | SRECH     | Tableau Desktop  | Initial SQL - Accounts Receivable - Live | SAPHANADB,/BIO/TAC... |
| 7 | bw4hana | SELECT ADD_DAYS(CAST(TO_DATE(*tableau_FIN_AR_DFIAR...  | SRECH     | Tableau Desktop  | Initial SQL - Accounts Receivable - Live | SAPHANADB,/BIO/TAC... |
| 8 | bw4hana | SELECT *tableau_FIN_AR_DFIAR30_CV"."ACCOUNTING_DO...   | SRECH     | Tableau Desktop  | Initial SQL - Accounts Receivable - Live | SAPHANADB,/BIO/TAC... |
| 9 | bw4hana | SELECT *tableau_FIN_AR_DFIAR30_CV"."ACCOUNTING_DO...   | SRECH     | Tableau Desktop  | Initial SQL - Accounts Receivable - Live | SAPHANADB,/BIO/TAC... |

## 高成本语句跟踪：找出长时间运行的查询和内存密集型查询

SAP HANA 提供了可能的途径来跟踪执行时间超过配置阈值的 SQL 语句。这种广泛使用的功能称为“高成本语句跟踪”，通常用于找出需要进行性能优化的查询。

如果没有在 Tableau 的初始 SQL 中分配 SAP HANA 会话变量，我们可能很难找出哪些长时间运行的 SQL 语句来自 Tableau（取决于使用的 Tableau 版本），以及哪个 Tableau 工作簿引发了相关查询（这是更加重要的信息）。

在 Tableau 的初始 SQL 中分配会话变量后，这会变得容易很多。配置跟踪时，我们不仅可以在用户级别限制跟踪，还可以在应用程序级别对其进行限制。必须注意阈值持续时间（以微秒为单位：1  $\mu$ s = 百万分之一秒）。如果只应该跟踪长时间运行的查询，这个值应该很高（默认为 1,000,000）。但如果应该跟踪全部或大多数 Tableau 查询，则可以将其设为较低的值（例如 1,000，也就是 1 毫秒）。

| Overview   | Landscape | Alerts   | Performance | Volumes | Configuration | System Information | Diagnosis Files | Trace Configuration |
|--|-----------|--|-------------|---------|---------------|--------------------|-----------------|---------------------|
| If the database trace is configured, the traces for the trace components of the system (for example, INDEXSERVER and NAMESERVER) are written to files named <servicename>_<host>.<port_number>.<3_digit_file_counter>.trc. Some of these traces are always activated by default.   |           | If the SQL trace is active, the database calls for the specified database, or application users are traced. The trace data is stored in files starting with sqltrace_<host>.<port_number>.<3_digit_file_counter>.py. |             |         |               |                    |                 |                     |
| <b>User-Specific Trace</b>   |           | <b>Performance Trace</b>   |             |         |               |                    |                 |                     |
| Configuration: <b>Not Specified</b>  |           | Status: <b>Inactive</b>  |             |         |               |                    |                 |                     |
| If the user trace is configured, the traces for the trace components for a specific database or application user (for example, INDEXSERVER and NAMESERVER) are written to files named <servicename>_<host>.<port_number>.<context>.trc.  |           | If the performance trace is running, the system performance is traced. The trace data is saved to the file specified.  |             |         |               |                    |                 |                     |
| <b>End-to-End Traces</b>   |           | <b>Expensive Statements Trace</b>  |             |         |               |                    |                 |                     |
| The predefined end-to-end traces are used by applications to record the steps through all the available trace components (INDEXSERVER and NAMESERVER for example) in a configuration. When an end-to-end trace is used, the traces for the trace components are written to files named <servicename>_<host>.<port_number>.<end-to-end-trace_name>.trc. |           | Status: <b>Inactive</b>  |             |         |               |                    |                 |                     |
|  |           | If the expensive statements trace is active, all statements that last longer than the specified threshold are traced. You can analyze the results on the Performance tab under Expensive Statements Trace.           |             |         |               |                    |                 |                     |

### Expensive Statements Trace

Configure the expensive statements trace by specifying the necessary options

Trace Status:  Inactive  Active

Threshold Duration (µs):

User Filter

No user filter

Database User:

Application User:

Table/View:

Application:

Passport Trace Level:

Trace parameter values

跟踪完成后，在 Expensive Statements（高成本语句）视图上使用 select（选择）（包括 APPLICATION\_NAME 列上的筛选器）将返回源自 Tableau 的语句。

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

结果集包含每项数据库操作的重要信息，其中包括：用户名、工作簿名称、SQL 语句、开始时间、持续时间（以微秒为单位）、受影响的数据库表和记录数。可以在“where”子句中将此查询进一步限制为特定的用户、工作簿、开始时间等。

以下信息可能特别有用：

- 语句执行期间的操作类型 (OPERATION)
- 查询开始时间 (START\_TIME)
- 查询所用的时间 (DURATION\_MICROSEC)
- 用于计算语句的 CPU 时间（以微秒为单位）(CPU\_TIME)
- 访问的对象的名称 (OBJECT\_NAME)
- SQL 语句 (STATEMENT\_STRING)
- 执行语句期间的峰值内存使用量（以字节为单位）(MEMORY\_SIZE)



记录的一些重要操作类型包括:

| 操作                             | 说明            |
|--------------------------------|---------------|
| AGGREGATED_EXECUTION           | 单个数据库请求的总执行时间 |
| CALL                           | 过程调用的执行时间     |
| COMPILE                        | 准备/解析时间       |
| CURSOR_CLOSE                   | 光标关闭时间        |
| FETCH                          | 获取时间          |
| SELECT, INSERT, UPDATE, DELETE | 对应操作的执行时间     |

高成本语句跟踪结果示例:

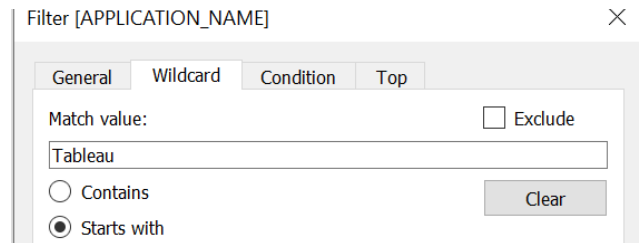
```
SQL Result
select * from M_EXPENSIVE_STATEMENTS where SESSION_VARIABLES like '%Tableau%'
```

| 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': 'Tableau Desktop'; 'APPLICATIONSOUR |
| 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': 'Tableau Desktop'; 'APPLICATIONSOUR |
| 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': 'Tableau Desktop'; 'APPLICATIONSOUR |
| 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': 'Tableau Desktop'; 'APPLICATIONSOUR |
| 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': 'Tableau Desktop'; 'APPLICATIONSOUR |
| 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': 'Tableau Desktop'; 'APPLICATIONSOUR |
| 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': 'Tableau Desktop'; 'APPLICATIONSOUR |
| 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': 'Tableau Desktop'; 'APPLICATIONSOUR |
| SELECT 'tableau_FIN_AR_DFIAR30_CV':'CREDIT_AMOUNT_L... | 14,600   | Jun 8, 2020 5:03:32.01658 AM  | Initial SQL - Accounts Receivable - Live | Tableau Desktop  | ('APPLICATION': 'Tableau Desktop'; 'APPLICATIONSOUR |
| SELECT 'tableau_FIN_AR_DFIAR30_CV':'CREDIT_AMOUNT_L... | 14,557   | Jun 8, 2020 5:03:32.01658 AM  | Initial SQL - Accounts Receivable - Live | Tableau Desktop  | ('APPLICATION': 'Tableau Desktop'; 'APPLICATIONSOUR |
| 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': 'Tableau Desktop'; 'APPLICATIONSOUR |
| SELECT 'tableau_FIN_AR_DFIAR30_CV':'CREDIT_AMOUNT_L... | 18,963   | Jun 8, 2020 5:03:31.838889 AM | Initial SQL - Accounts Receivable - Live | Tableau Desktop  | ('APPLICATION': 'Tableau Desktop'; 'APPLICATIONSOUR |

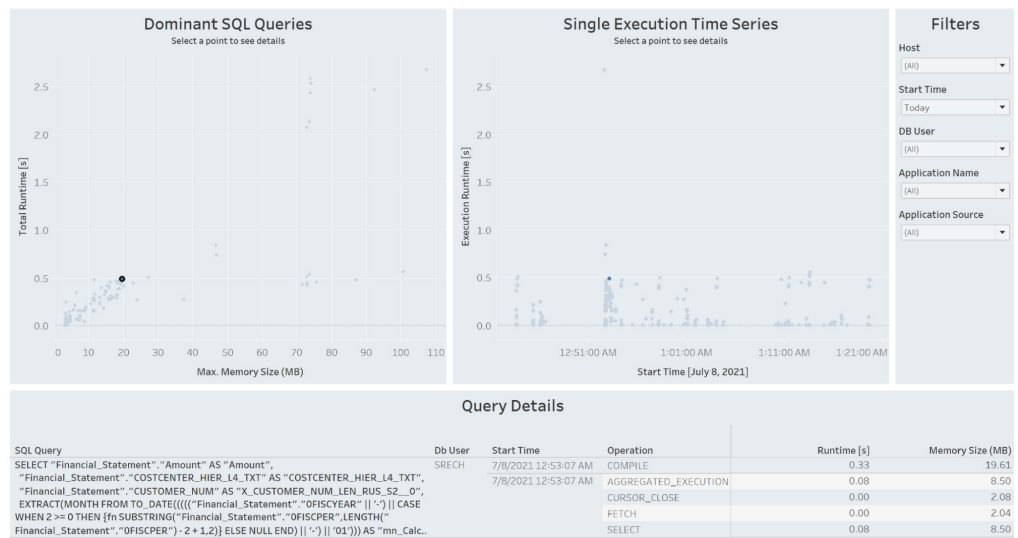
| DURATION_MICROSEC | OBJECT_NAME   | OPERATION            | RECORDS |
|-------------------|---|----------------------|---------|
| 14,646            | SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB... | AGGREGATED_EXECUTION | 35      |
| 14,604            | SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB... | SELECT               | 0       |
| 19,974            | SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB... | COMPILE              | 0       |
| 15,531            | SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB... | AGGREGATED_EXECUTION | 70      |
| 15,480            | SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB... | SELECT               | 0       |
| 20,365            | SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB... | COMPILE              | 0       |
| 22,328            | SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB... | AGGREGATED_EXECUTION | 1       |
| 22,317            | SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB... | SELECT               | 0       |
| 14,540            | SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB... | AGGREGATED_EXECUTION | 31      |
| 14,514            | SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB... | SELECT               | 0       |
| 21,458            | SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB... | COMPILE              | 0       |
| 18,961            | SAPHANADB/B1H/AD_FIAR302.SAPHANADB/B10/TAC_DOC_TYP.SAPHANADB... | COMPILE              | 0       |

## 在 Tableau 中分析 SAP HANA 高成本语句跟踪结果

Tableau 可用于分析高成本语句跟踪的结果。为此，需要建立与 Schema“SYS”中的“M\_EXPENSIVE\_STATEMENTS”视图的连接。要筛选为仅包含由 Tableau 触发的语句，可以使用数据源筛选器（例如，APPLICATION\_NAME 维度上的 Tableau 通配符筛选器）。



用于分析高成本查询的示例仪表板：



## SQL Plan Cache 统计数据：确定优化候选对象（长期运行和频繁执行）

SQL Plan Cache 是一款很有用的工具，可以帮助您了解 SAP HANA 数据库的 SQL 处理。它可以提供关于在系统中执行的语句的概要信息，还可以跟踪执行运行时间等统计信息。它让您了解频繁执行的查询和速度缓慢的查询，因此可用于确定优化候选对象（无需激活专用跟踪）。SQL Plan Cache 可以作为 SAP HANA 视图来进行查询，因此您可以使用 Tableau 分析这些信息。

在 SAP HANA 中，SQL 语句会在执行之前编译为计划。编译计划后，最好在下次执行相同语句时重新使用该计划，而不是每次都编译一个新计划。在 SAP HANA 中，SQL Plan Cache 会存储从先前的执行生成的计划。此外，它还会保留每个计划的统计信息以便进行监测。因此您可以分析执行次数、最短/最长/总计/平均运行时间、锁/等待统计数据等。

以下信息可能特别有用：

- 主要语句 (TOTAL\_EXECUTION\_TIME)
- 长时间运行的语句 (AVG\_EXECUTION\_TIME)
- 频繁执行的计划 (EXECUTION\_COUNT)
- 返回的记录数 (TOTAL\_RESULT\_RECORD\_COUNT)

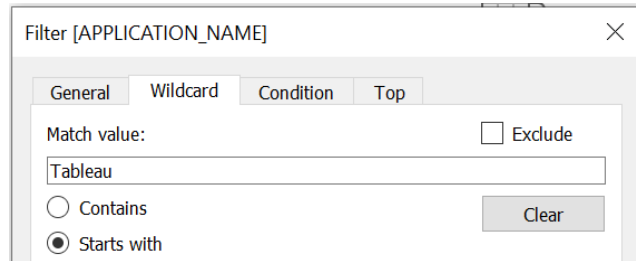
SAP Note 2000002 (常见问题: [SAP HANA SQL 优化](#)) 提供了如何解释 SQL Plan Cache 中的运行时统计数据的更多相关信息。举例来说, 操作可分解为以下类别:

| 操作              | 说明   |
|-----------------|--|
| CURSOR          | 包含总计光标时间, 包括 SAP HANA 服务器时间和客户端时间; 如果客户端在不同轮次的数据获取之间执行其他任务, 则光标时间可能远高于 SAP HANA 服务器时间。 |
| EXECUTION       | 包含 SAP HANA 服务器端的执行时间 (open + fetch + lock wait + close), 不包括表加载和准备时间。                 |
| EXECUTION_OPEN  | 包含 SAP HANA 服务器端的打开时间; 包括对数据的实际检索 (在早期具体化的列式存储访问的情况下)。                                 |
| EXECUTION_FETCH | 包含 SAP HANA 服务器端的获取时间; 包括对数据的实际检索 (在行式存储访问或延迟具体化的情况下)。                                 |
| EXECUTION_CLOSE | 包含 SAP HANA 服务器端的关闭时间。   |
| TABLE_LOAD      | 包含准备期间的表加载时间, 是准备时间的一部分。   |
| REPARATION      | 包含准备时间。  |
| LOCK_WAIT       | 包含事务锁等待时间, 不包含内部锁。   |

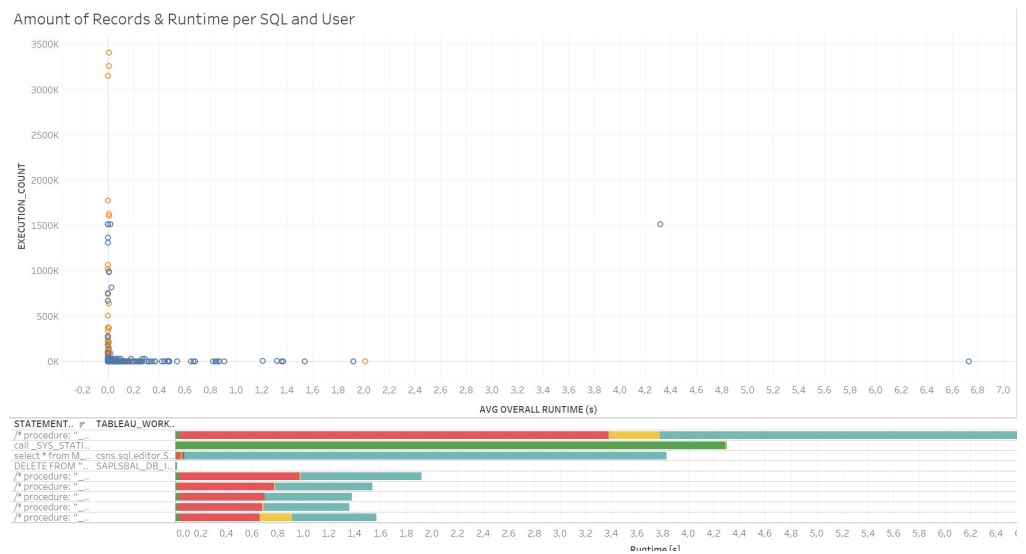
有关如何读取 SQL Plan Cache 的指南, 请参阅 SAP 帮助页面[示例: 读取 SQL Plan Cache](#)。如需关于如何分析 SQL Plan Cache 的建议, 请访问 SAP 帮助页面[SQL Plan Cache 分析](#)。

## 在 Tableau 中分析 SAP HANA SQL Plan Cache

Tableau 可用于分析 SQL Plan Cache 的结果。为此，需要建立与 Schema“SYS”中的“M\_SQL\_PLAN\_CACHE”视图的 SAP HANA 连接。要筛选为仅包含由 Tableau 触发的语句，可以使用数据源筛选器（例如，APPLICATION\_NAME 维度上的 Tableau 通配符筛选器）。



示例可视化可以包括执行计数和平均总体运行时间，以及运行时间的平均分布：

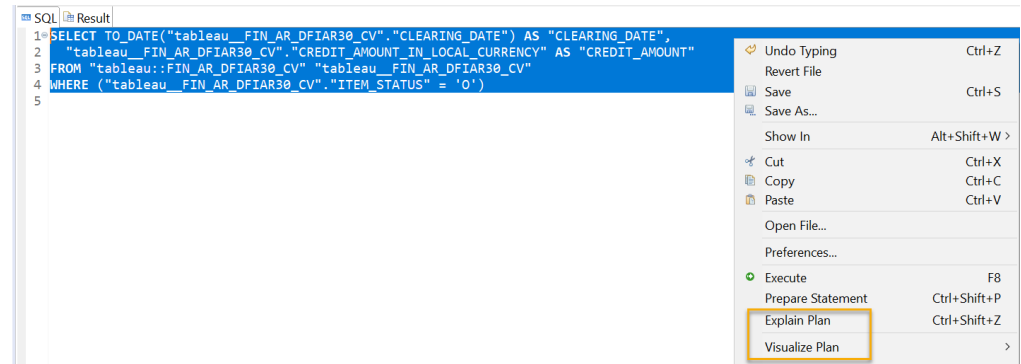


# 详细分析单个 SQL 查询： SAP HANA Explain Plan 和 Visualize Plan

使用高成本语句跟踪或 SQL Plan Cache 统计数据找出长时间运行的查询后，下一步是确定为什么会耗费如此长的执行时间。

要确定运行时间过长的根本原因，了解 SAP HANA 如何处理 SQL 语句执行会有所帮助。SAP HANA 中的“Explain Plan”和“Visualize Plan”功能提供了两种方式进行这项调查。

执行其中任何一个的最简单方法是将语句复制到 SQL 控制台并在上下文菜单中选择“Explain Plan”或“Visualize Plan”。



接下来的两个部分将更详细地介绍了每种功能。

## SAP HANA Plan Explanation

生成 SQL 语句的计划解释后，结果会显示查询执行的详细信息，以及处理过程中涉及的数据库操作。

```
= EXPLAIN PLAN FOR
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" = '0')
```

|   | OPERATOR_NAME | OPERATOR_DETAILS  | TABLE_NAME      | OUTPUT_SIZE | SUBTREE_COST           | OPERATOR_ID | LEVEL |
|---|---------------|---|-----------------|-------------|------------------------|-------------|-------|
| 1 | COLUMN SEARCH | TO_DATE(/B1H/AD_FIAR302.CLEARING_DATE), TO_DECIMAL(SUM(/B1H/AD_FIAR302.CRE... | ?               | 348.718     | 0.0007147089139647216  | 1           | 1     |
| 2 | AGGREGATION   | GROUPING: /B1H/AD_FIAR302.CLEAR_DATE, AGGREGATION: SUM(/B1H/AD_FIAR302.CRE... | ?               | 348.718     | 0.00008454636905806701 | 2           | 2     |
| 3 | COLUMN TABLE  | FILTER CONDITION: /B1H/AD_FIAR302.ITEM_STATUS = '0'                           | /B1H/AD_FIAR302 | 348.718     | ?                      | 3           | 3     |

此处以及随后的示例简要介绍了一些关键值。如需完整的详细信息，请参阅 [SAP HANA SQL 和系统视图参考](#) 中的 EXPLAIN\_PLAN\_TABLE 系统视图。

| 区域     | 详细信息  |
|--------|---|
| 操作详细信息 | OPERATOR_NAME 值显示执行的操作的类型，例如联接、并集、聚合等。操作取决于所用的引擎 — 简单地说就是行引擎或列引擎。依赖关系通过缩进来显示 — 见下方示例。 |
| 引擎     | 执行操作符的引擎类型显示在 EXECUTION_ENGINE 列中：ROW、COLUMN、OLAP、HEX、ESX。                            |
| 表详细信息  | 表详细信息包括表名称、类型、大小、表或访问的对象。   |
| 估计成本   | 成本值包括估计的输出行数 (OUTPUT_SIZE) 和以秒为单位的估计时间 (SUBTREE_COST)。                                |

以下几个页面描述了关于如何解释 Explain Plan 结果的示例：

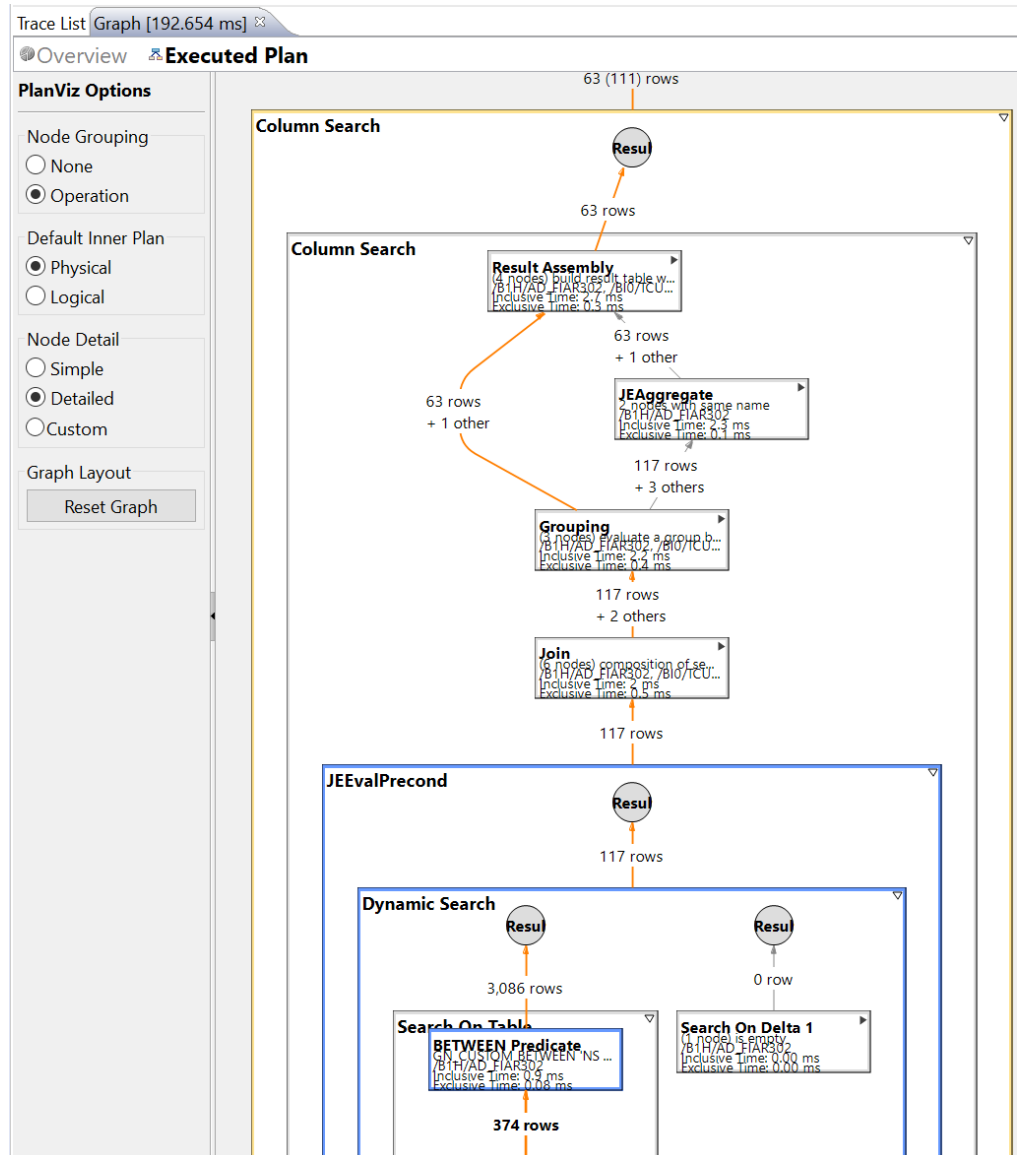
- [使用 Plan Explanation 分析 SQL 执行](#)
- [SAP HANA SQL 和系统视图参考： Explain Plan 语句](#)

## SAP HANA Plan Visualizer

SAP HANA Plan Visualizer 让您能够对 SQL 执行计划进行图形分析。因此您可以更轻松地了解处理过程中涉及的步骤，以及记录数量和运行时间如何随着时间的推移而变化。

运行时间以“Exclusive”（节点的执行时间）和“Inclusive”（包括后代节点的执行时间）值的形式给出。

| Time                        |                       | Context                            |   |
|-----------------------------|-----------------------|------------------------------------|---|
| Compilation                 | 20.92 ms              | SQL Query                          | SELECT TO_DATE("tableau_FIN_AR_DFIAR30... |
| Execution                   | 192.65 ms             | System                             | bw4hana:30203                             |
| <b>Dominant Operators</b>   |                       | System Version                     | 2.00.043.00.1569560581                    |
| <b>Name</b>                 | <b>Execution Time</b> | System Compile Type                | rel                                       |
| Basic Predicate             | 0.66 ms (0.34%)       | Memory Allocated                   | 11.7 MByte(s)                             |
| JEDistinctAttribute         | 0.25 ms (0.13%)       | <b>Data Flow</b>                   |   |
| Column Search               | 0.21 ms (0.11%)       | Number of Tables Used <sup>®</sup> | 2   |
| <b>Distribution</b>         |                       | Result Record Count                | 63  |
| Number of Nodes             | 1                     |                                    |   |
| Number of Network Transfers | 0                     |                                    |   |



Plan Visualizer 提供了额外的视图来进行分析，例如 Timeline View（时间线视图）和 Network View（网络视图）。如需这方面的更多信息，请参阅[使用 Plan Visualizer 分析 SQL 执行](#)一章中的“SAP HANA 故障排除和性能分析指南”。

此外，SAP 发布了几篇说明如何使用 PlanViz 的博文：

- [The HANA PlanVisualizer \(PlanViz\) – Quick and Easy](#)
- [Analyzing SQL Execution with the Plan Visualizer \(PlanViz\)](#)



一旦确定了 SAP HANA 查询处理中的瓶颈，就应该着手加以解决。要详细了解在 SAP HANA 计算视图中进行性能优化的潜在方法，请参阅“SAP HANA 开发人员性能指南”中的[计算视图中的优化功能](#)一章，以及 SAP 知识库文章 2000002 ([常见问题：SAP HANA SQL 优化](#))。

为了更改 Tableau 发送到 SAP HANA 的 SQL 查询，需要考虑对 Tableau 中的仪表板进行重新建模。您可以在以下资源中找到关于仪表板性能的最佳做法：[设计高效工作簿和仪表板性能最佳做法](#) (英文)。

## 结语

为了了解 SAP 上的 Tableau 仪表板的性能，请在 Tableau 端和 SAP HANA 端使用多个跟踪。

要优化单个仪表板，建议使用结合了 Tableau 的性能记录器和 SAP HANA 的 SQL 跟踪功能的端到端执行跟踪。这种方法可以揭示运行时间如何在不同的执行步骤和处理层之间分布，并帮助您发现瓶颈。

如果要进行长期性能监测，SAP HANA 高成本语句跟踪是最合适的方法，因为它产生的开销极小，并且可以永久保持激活状态。

一旦找出了长时间运行的或内存密集型 SQL 查询，就可以使用 SAP HANA 的 Plan Explanation 或 Plan Visualizer 对其做进一步分析，以此了解数据库执行步骤及其对运行时间的影响。

## 关于 Tableau

Tableau 是一个完整集成的企业级可视化分析平台，可帮助用户和组织提高数据驱动性。

无论是在本地还是在云端，在 Windows 还是 Linux 上，Tableau 都能够充分利用您现有的技术投资，随着您数据环境的变化和增长来进行扩展。让您最为宝贵的两项资产充分发挥价值：数据物尽其用，员工人尽其才。