Hyperion Focus 17

Monitoring EPM systems performance with Oracle Data Visualization

Alex Znyk

Ingersoll Rand

Disclaimer

This presentation is based on my experience. It does not reflect Ingersoll Rand or Oracle official positions. Any errors are my own.

This presentation is based on the following versions (on-premise):

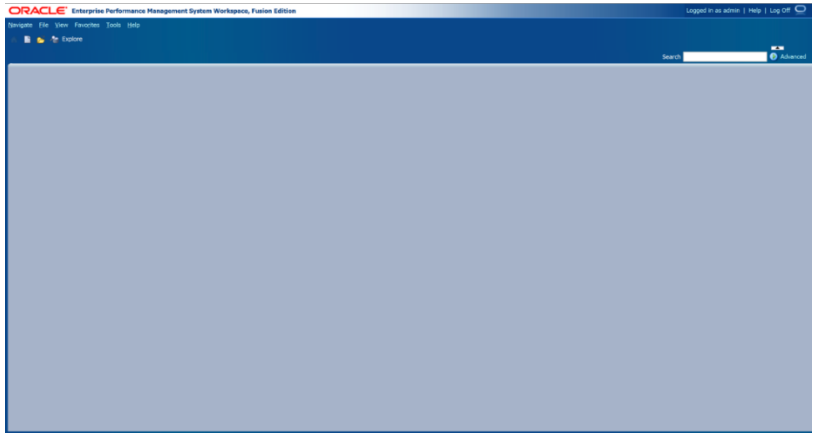
- HFM: 11.1.2.4.204.5267 [Workspace: Help => About]
- Planning: 11.1.2.4.005.08 [Workspace: Help => About]
- Essbase: 11.1.2.4.0 [EAS or MaxL]
- Databases on SQL Server (old version)
- Oracle Data Visualization Desktop: 12.2.3

Things may differ if you are using other versions.

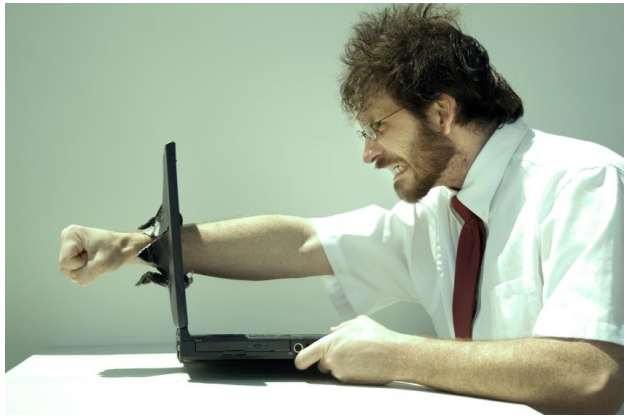
Introduction

This presentation is about getting from…

"If my system is not down,
it means it's working!"

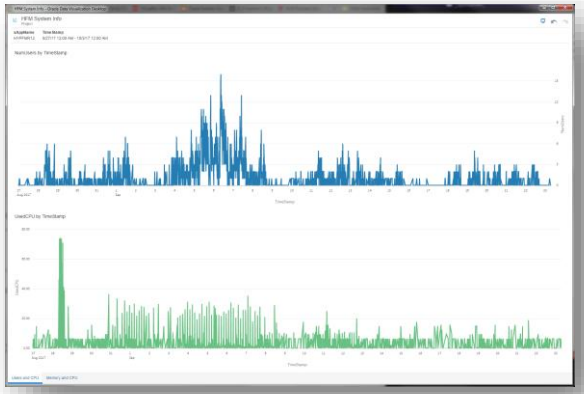


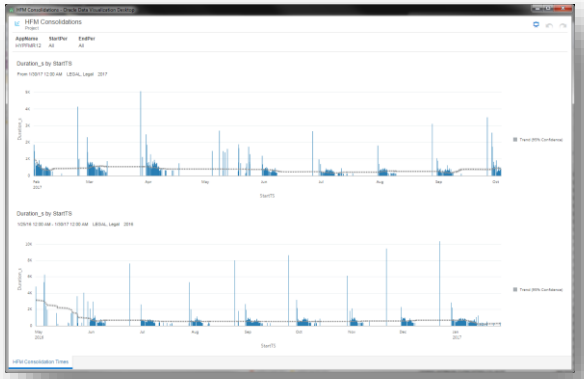
"If users are not screaming, surely performance is good?"

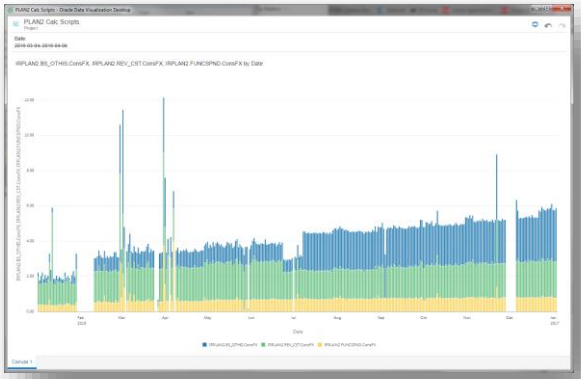


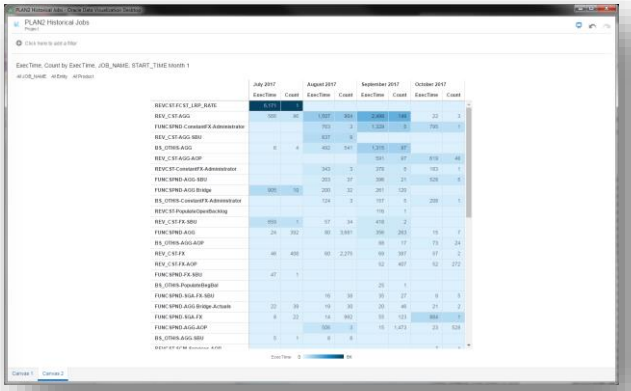
Introduction

To a place where you know how your systems are behaving, and you can identify performance trends:









Introduction

We will achieve this fine goal by:

- Exploring examples of information that can be gathered from Planning, Essbase and HFM; reviewing how to collect and prepare the data for consumption by Oracle DV
- Understanding how to connect Oracle DV to data sources
- Creating visualizations that allow to monitor EPM systems performance, and discussing some examples of how visualization can help understand system behavior and drive actions

Questions are welcome at any time !

Introduction

Who am I to talk about such things ?

Alex Znyk alexandre_znyk@irco.com

Sr Manager, Hyperion Systems Administration and Data Management at Ingersoll Rand, a global diversified manufacturing company
https://company.ingersollrand.com/

Prior to this, I oversaw global support of the Hyperion Systems, and had been on the team that performed the initial roll-out of the EPM systems.

My past experiences include:

- **Consulting (Hyperion integration, Financial Processes Optimization),**
- **I.T. Audit (internal and external),**
- **Financial Planning & Analysis (know thy enemy).**

At Ingersoll Rand, we use HFM, FDMEE, Planning, Essbase, EPM Maestro, DRM. Our user population is 1000+, located all over the world.

Hyperion Focus 17

Monitoring EPM systems performance with Oracle Data Visualization

Part 1: Where is my data?

Part 1 - Game plan

We will take a look at the different systems we are interested in, and where to find useful data for our purpose:

- **Planning: Job Status tables**
- **Essbase: using MaxL text log files to capture calculation execution times**
- **HFM: ErrorLog SysInfo records, Consolidation records**

Along the way, I will share different techniques that I have found useful to prepare the data for better analytics.

This list is not meant to be exhaustive. It is based on my experience, and is there to serve as a starting point.
If you find something useful and are willing to share, let me know, I am interested!

Hyperion Focus 17

Monitoring EPM systems performance with Oracle Data Visualization

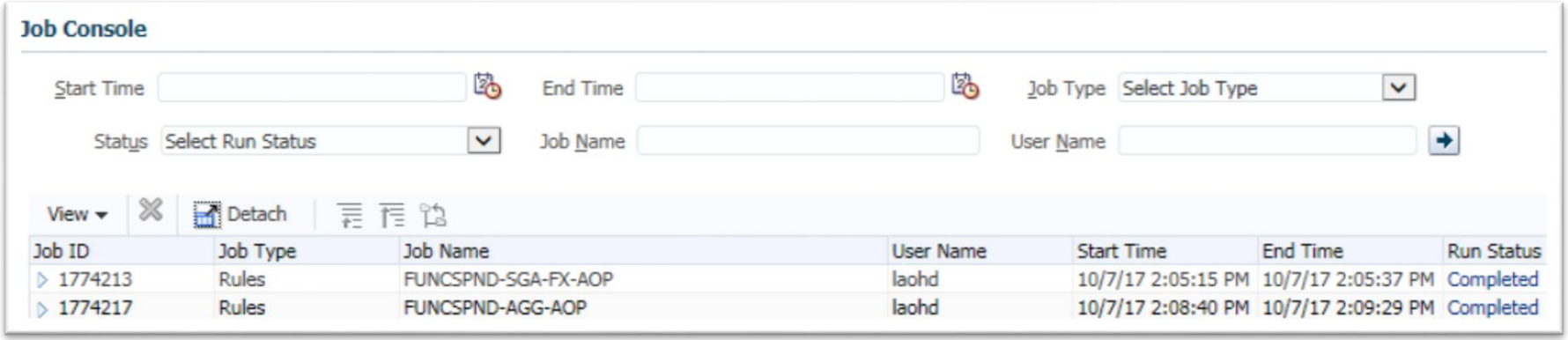
Part 1: Where is my data?

Hyperion Planning

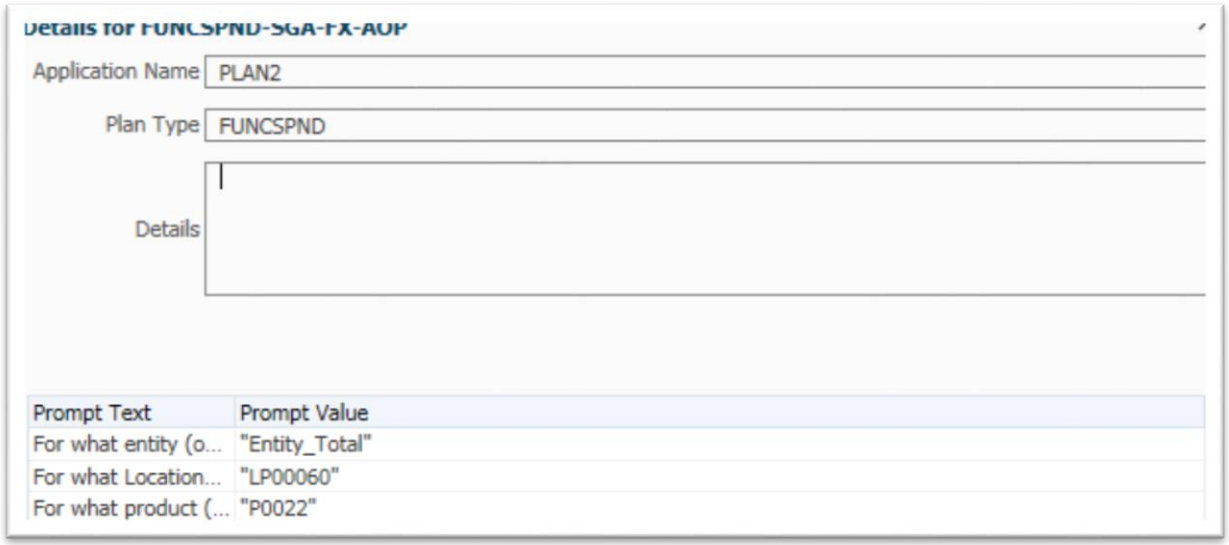
Job Status Tables

Part 1 – Hyperion Planning Job Status Tables

Information on business rule execution is available in the job console:



Detailed information about each job is available by clicking on the Run Status:



Part 1 – Hyperion Planning Job Status Tables

We can find this information in the database in the following tables:

HSP_JOB_STATUS: for recent jobs

HSP_HISTORICAL_JOB_STATUS: for older records

Let's take a look:

```
SELECT [JOB_ID]
      ,[PARENT_JOB_ID]
      ,[JOB_NAME]
      ,[JOB_TYPE]
      ,[USER_ID]
      ,[START_TIME]
      ,[END_TIME]
      ,[RUN_STATUS]
      ,[DETAILS]
      ,[ATTRIBUTE_1]
      ,[ATTRIBUTE_2]
      ,[SERVER_ID]
      ,[PARAMETERS]
      ,[JOB_SCHEDULER_NAME]
  FROM [HSP_JOB_STATUS]
```

Part 1 – Hyperion Planning Job Status Tables

The data has the same structure in the two tables:

- **A parent record for the rule execution**
- **Children record for each parameter**

We will use a query that joins the parent records and the parameters useful for analysis.

	JOB_ID	PARENT_JOB_ID	JOB_NAME	JOB_TYPE	USER_ID	START_TIME	END_TIME	RUN_STATUS	DETAILS	ATTRIBUTE_1
1	1774213	NULL	FUNCSPND-SGA-FX-AOP	1	202138	2017-10-07 14:05:15.340	2017-10-07 14:05:37.157	2		PLAN2
2	1774214	1774213	ENTITY	4	202138	2017-10-07 14:05:15.353	2017-10-07 14:05:15.353	2	"Entity_Total"	FUNCSPND-SGA-FX-AOP
3	1774215	1774213	Location	4	202138	2017-10-07 14:05:15.367	2017-10-07 14:05:15.367	2	"LP00060"	FUNCSPND-SGA-FX-AOP
4	1774216	1774213	Product	4	202138	2017-10-07 14:05:15.383	2017-10-07 14:05:15.383	2	"P0022"	FUNCSPND-SGA-FX-AOP
5	1774217	NULL	FUNCSPND-AGG-AOP	1	202138	2017-10-07 14:08:40.967	2017-10-07 14:09:29.843	2		PLAN2
6	1774218	1774217	Location	4	202138	2017-10-07 14:08:40.977	2017-10-07 14:08:40.977	2	"LP00060"	FUNCSPND-AGG-AOP
7	1774219	1774217	Product	4	202138	2017-10-07 14:08:40.990	2017-10-07 14:08:40.990	2	"P0022"	FUNCSPND-AGG-AOP
8	1774220	1774217	Entity	4	202138	2017-10-07 14:08:40.997	2017-10-07 14:08:40.997	2	"Entity_Total"	FUNCSPND-AGG-AOP

This will be useful when using the data to filter out records for specific business rules run with the same parameters.

The query is provided on the next page, and a sample of the output follows.

Part 1 – Hyperion Planning Job Status Tables

Query:

```
SELECT PJ.[JOB_ID]
      ,PJ.[JOB_NAME]
      ,PJ.[START_TIME]
      ,PJ.[END_TIME]
     ,DATEDIFF(ss,[START_TIME],[END_TIME]) AS [ExecTime]
      ,PJ.[RUN_STATUS]
      ,PJ.[ATTRIBUTE_1]
      ,PJ.[ATTRIBUTE_2]
     ,REPLACE(CAST(CJ1.[DETAILS] AS nvarchar(max)),'"', '') AS [Entity]
     ,REPLACE(CAST(CJ2.[DETAILS] AS nvarchar(max)),'"', '') AS [Product]

 INTO zHISTJOB

  FROM [HSP_HISTORICAL_JOB_STATUS] AS PJ

  LEFT OUTER JOIN (
  SELECT [JOB_ID]
       ,[PARENT_JOB_ID]
       ,[JOB_NAME]
       ,[DETAILS]
  FROM [HSP_HISTORICAL_JOB_STATUS]
  WHERE [PARENT_JOB_ID] IS NOT NULL
   AND [JOB_NAME] = 'lEntity' OR [JOB_NAME] = 'Entity')
  ) AS CJ1
  ON PJ.[JOB_ID] = CJ1.[PARENT_JOB_ID]

 LEFT OUTER JOIN (
  SELECT [JOB_ID]
       ,[PARENT_JOB_ID]
       ,[JOB_NAME]
       ,[DETAILS]
  FROM [HSP_HISTORICAL_JOB_STATUS]
  WHERE [PARENT_JOB_ID] IS NOT NULL
   AND ([JOB_NAME] = 'Product')
  ) AS CJ2
  ON PJ.[JOB_ID] = CJ2.[PARENT_JOB_ID]

  WHERE PJ.[PARENT_JOB_ID] IS NULL
   AND PJ.[JOB_TYPE] = 1

  ORDER BY [START_TIME] DESC
```

Part 1 – Hyperion Planning Job Status Tables

Sample output:

JOB_ID	JOB_NAME	START_TIME	END_TIME	ExecTime	RUN_STATUS	ATTRIBUTE_1	ATTRIBUTE_2	Entity	Product
1774209	FUNCSPND-AGG-...	2017-10-07 13:16:48.007	2017-10-07 13:16:55.507	7	2	PLAN2	FUNCSPND	Entity_Total	P11000
1774205	FUNCSPND-SGA-F...	2017-10-07 13:16:35.357	2017-10-07 13:16:37.983	2	2	PLAN2	FUNCSPND	Entity_Total	P11000
1774155	REV_CST-AGG-AOP	2017-10-07 13:00:48.507	2017-10-07 13:01:04.250	16	2	PLAN2	REV_CST	Entity_Total	P0022
1774150	REV_CST-FX-AOP	2017-10-07 12:57:45.120	2017-10-07 12:58:55.553	70	2	PLAN2	REV_CST	Entity_Total	P0022
1774147	REVCST-SCM-Cont...	2017-10-07 12:56:17.750	2017-10-07 12:56:18.813	1	2	PLAN2	REV_CST	E9258	NULL
1774144	REVCST-SCM-Serv...	2017-10-07 12:55:05.413	2017-10-07 12:55:12.683	7	2	PLAN2	REV_CST	E9258	NULL
1774140	FUNCSPND-AGG-...	2017-10-07 12:44:19.973	2017-10-07 12:44:37.207	18	2	PLAN2	FUNCSPND	Entity_Total	P11000
1774136	FUNCSPND-SGA-F...	2017-10-07 12:43:48.160	2017-10-07 12:44:00.393	12	2	PLAN2	FUNCSPND	Entity_Total	P11000

Hyperion Focus 17

Monitoring EPM systems performance with Oracle Data Visualization

Part 1: Where is my data?

Hyperion Essbase

MaxL Script logs

Part 1 – Hyperion Essbase MaxL Script Logs

A common practice for Essbase cubes is to run scheduled calculation scripts. For example, perform a nightly calculation or aggregation.

This can take the form of a MaxL script similar to this:

```
set timestamp on;
SPOOL ON to 'PLAN-ConsFX\PLANConsFX.log';
LOGIN $1 $2 on $3;
execute calculation $5.'BS_OTHIS'.'ConsFX';
execute calculation $5.'REV_CST'.'ConsFX';
execute calculation $5.'FUNCSPND'.'ConsFX';
execute calculation $5.'BS_OTHIS'.'ConsFXAg';
execute calculation $5.'REV_CST'.'ConsFXAg';
execute calculation $5.'FUNCSPND'.'ConsFXAg';
SPOOL OFF;
EXIT;
```

Part 1 – Hyperion Essbase MaxL Script Logs

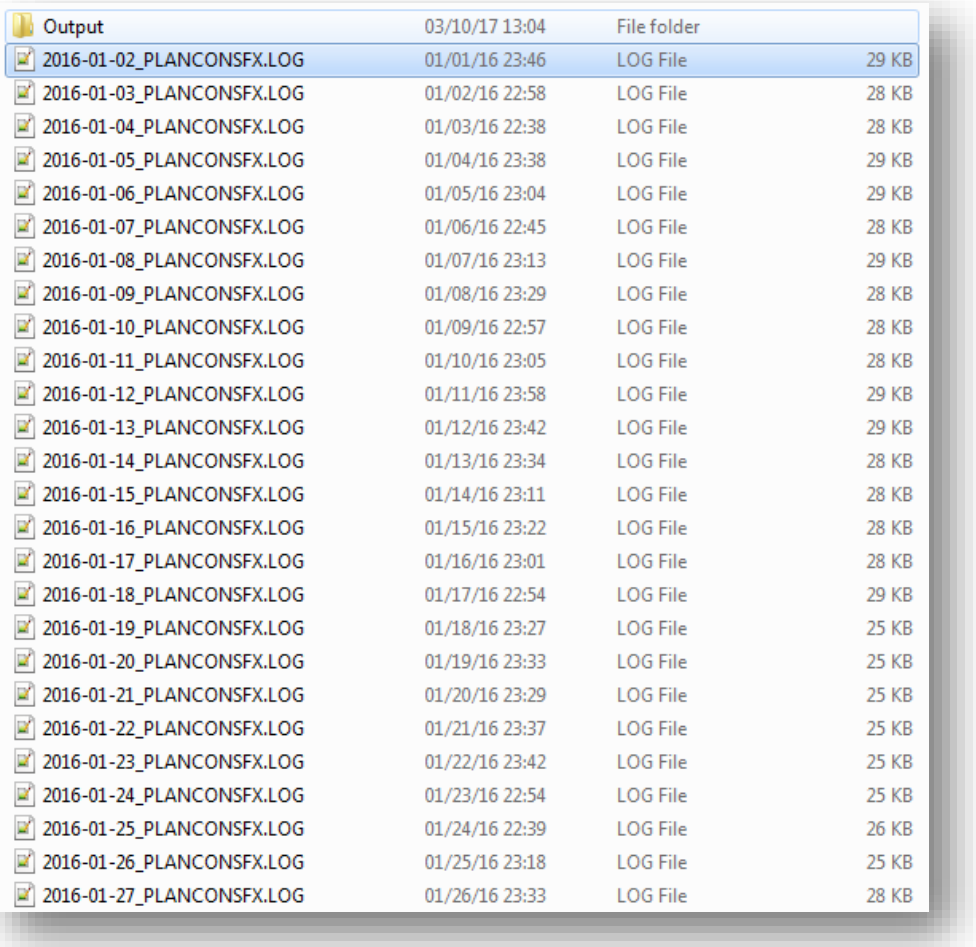
The output into the log file will be similar to this:

```
     essmsh timestamp: Sun Oct 08 22:30:00 2017

MAXL> execute calculation PLAN2.'BS_OTHIS'.'ConsFX';

 OK/INFO - 1200481 - Formula for member [Entity_Currency] will be executed in [TOPDC
 OK/INFO - 1200481 - Formula for member [Entity_Currency] will be executed in [TOPDC
 OK/INFO - 1012558 - Clearing all data blocks from [(ALL)] partition with fixed memb
 OK/INFO - 1012675 - Commit Blocks Interval for the calculation is [[0]].
 OK/INFO - 1017023 - No message for message #1017023 in message database.
 OK/INFO - 1012675 - Commit Blocks Interval for the calculation is [[0]].
 OK/INFO - 1012668 - Calculating [ Currency(Entity_Currency)] with fixed members [En
 OK/INFO - 1012678 - Calculating in parallel with [6] threads.
 OK/INFO - 1012679 - Calculation task schedule [1].
 OK/INFO - 1012680 - Parallelizing using [1] task dimensions. .
 OK/INFO - 1012553 - Copying data from [Working->FCST_09_03->Entity_Currency] to [Wo
 OK/INFO - 1012675 - Commit Blocks Interval for the calculation is [[0]].
 OK/INFO - 1012675 - Commit Blocks Interval for the calculation is [[0]].
 OK/INFO - 1012574 - Datacopy command copied [57849] source data blocks to [57849] t
 OK/INFO - 1012675 - Commit Blocks Interval for the calculation is [[0]].
 OK/INFO - 1012569 - Formula on (or backward dependence from) member [Working] force
 OK/INFO - 1012668 - Calculating [ Version(Working)] with fixed members [Period(Jan,
 OK/INFO - 1012677 - Calculating in serial.
 OK/INFO - 1012675 - Commit Blocks Interval for the calculation is [[0]].
 OK/INFO - 1012668 - Calculating [ Currency(Entity_Currency)] with fixed members [En
 OK/INFO - 1012678 - Calculating in parallel with [6] threads.
 OK/INFO - 1012679 - Calculation task schedule [1].
 OK/INFO - 1012680 - Parallelizing using [1] task dimensions. .
 OK/INFO - 1012553 - Copying data from [Working->FCST_09_03->Entity_Currency] to [Wo
 OK/INFO - 1012675 - Commit Blocks Interval for the calculation is [[0]].
 OK/INFO - 1012675 - Commit Blocks Interval for the calculation is [[0]].
 OK/INFO - 1012574 - Datacopy command copied [57849] source data blocks to [57849] t
 OK/INFO - 1012675 - Commit Blocks Interval for the calculation is [[0]].
 OK/INFO - 1012569 - Formula on (or backward dependence from) member [Working] force
 OK/INFO - 1012668 - Calculating [ Version(Working)] with fixed members [Period(Jan,
 OK/INFO - 1012677 - Calculating in serial.
 OK/INFO - 1012579 - Total Calc Elapsed Time for [ConsFX.csc] : [154.155] seconds.
 OK/INFO - 1013274 - Calculation executed.

     essmsh timestamp: Sun Oct 08 22:32:35 2017
```

Part 1 – Hyperion Essbase MaxL Script Logs

In my case, our process creates a separate log file each time the script is executed (on a daily basis).

I collected the files into a folder:



Output	03/10/17 13:04	File folder	
2016-01-02_PLANCONSFX.LOG	01/01/16 23:46	LOG File	29 KB
2016-01-03_PLANCONSFX.LOG	01/02/16 22:58	LOG File	28 KB
2016-01-04_PLANCONSFX.LOG	01/03/16 22:38	LOG File	28 KB
2016-01-05_PLANCONSFX.LOG	01/04/16 23:38	LOG File	29 KB
2016-01-06_PLANCONSFX.LOG	01/05/16 23:04	LOG File	29 KB
2016-01-07_PLANCONSFX.LOG	01/06/16 22:45	LOG File	28 KB
2016-01-08_PLANCONSFX.LOG	01/07/16 23:13	LOG File	29 KB
2016-01-09_PLANCONSFX.LOG	01/08/16 23:29	LOG File	28 KB
2016-01-10_PLANCONSFX.LOG	01/09/16 22:57	LOG File	28 KB
2016-01-11_PLANCONSFX.LOG	01/10/16 23:05	LOG File	28 KB
2016-01-12_PLANCONSFX.LOG	01/11/16 23:58	LOG File	29 KB
2016-01-13_PLANCONSFX.LOG	01/12/16 23:42	LOG File	29 KB
2016-01-14_PLANCONSFX.LOG	01/13/16 23:34	LOG File	28 KB
2016-01-15_PLANCONSFX.LOG	01/14/16 23:11	LOG File	28 KB
2016-01-16_PLANCONSFX.LOG	01/15/16 23:22	LOG File	28 KB
2016-01-17_PLANCONSFX.LOG	01/16/16 23:01	LOG File	28 KB
2016-01-18_PLANCONSFX.LOG	01/17/16 22:54	LOG File	29 KB
2016-01-19_PLANCONSFX.LOG	01/18/16 23:27	LOG File	25 KB
2016-01-20_PLANCONSFX.LOG	01/19/16 23:33	LOG File	25 KB
2016-01-21_PLANCONSFX.LOG	01/20/16 23:29	LOG File	25 KB
2016-01-22_PLANCONSFX.LOG	01/21/16 23:37	LOG File	25 KB
2016-01-23_PLANCONSFX.LOG	01/22/16 23:42	LOG File	25 KB
2016-01-24_PLANCONSFX.LOG	01/23/16 22:54	LOG File	25 KB
2016-01-25_PLANCONSFX.LOG	01/24/16 22:39	LOG File	26 KB
2016-01-26_PLANCONSFX.LOG	01/25/16 23:18	LOG File	25 KB
2016-01-27_PLANCONSFX.LOG	01/26/16 23:33	LOG File	28 KB

Part 1 – Hyperion Essbase MaxL Script Logs

To extract the data, I use an Excel VBA script which reads through the files, captures the execution time and outputs it to a spreadsheet page using different columns for the different cubes.

The output is the following:

Date	IRPLAN.BSCAP.ConsFX	IRPLAN.PANDL.ConsFX	IRPLAN2.BS_OTHIS.ConsFX	IRPLAN2.REV_CST.ConsFX	IRPLAN2.FUNCSPND.ConsFX
01/02/16	1.50	13.11	0.26	1.57	0.38
01/03/16	0.60	7.36	0.25	1.17	0.34
01/04/16	4.43	33.94	0.47	1.16	0.39
01/05/16	2.68	19.82	0.27	1.55	0.40
01/06/16	0.89	10.03	0.28	1.21	0.35
01/07/16	3.33	29.10	0.45	1.29	0.34
01/08/16	5.79	30.84	0.36	1.19	0.35
01/09/16	1.82	12.92	0.26	1.44	0.34
01/10/16	6.59	25.10	0.54	2.10	0.64
01/11/16	7.90	49.25	0.67	1.36	0.33
01/12/16	5.00	36.91	0.32	5.23	0.35
01/13/16	7.87	38.02	0.33	1.20	0.33
01/14/16	4.01	24.03	0.35	1.15	0.34
01/15/16	5.67	26.12	0.30	1.40	0.33
01/16/16	1.26	13.17	0.25	1.26	0.35
01/17/16	2.53	19.01	0.29	1.26	0.42
01/18/16	5.77	37.85	0.27	1.26	0.37
01/19/16	6.50	36.33	0.25	1.23	0.33
01/20/16	5.67	32.84	0.27	1.21	0.35
01/21/16	6.55	37.59	0.26	1.56	0.39
01/22/16	6.89	40.97	0.26	1.22	0.44
01/23/16	1.26	12.59	0.15	1.43	0.44
01/24/16	0.54	7.22	0.17	1.30	0.43
01/25/16	5.82	33.22	0.12	1.24	0.33
01/26/16	3.83	33.23	0.53	1.20	0.34
01/27/16	4.95	29.52	0.80	1.16	0.34
01/28/16	5.89	39.58	0.38	1.26	0.33

Part 1 – Hyperion Essbase MaxL Script Logs

The vba script:

```
Sub LoopThroughFiles()
  Dim MyObj As Object, MySource As Object, file As Variant
   Dim StrFile, strRange As String
   Dim i, j As Integer
   Dim fso As New FileSystemObject
   Dim dataArray
   Dim strScript, strExec As String

   i = 2
   strPath = "C:\Path\"
   StrFile = Dir(strPath & "*PLANCONSFX.LOG")
   Do While Len(StrFile) > 0
      strRange = "A" & i
      Sheets("LogLoad").Range(strRange) = Replace(StrFile, "_PLANCONSFX.LOG", "")
      StrFile = Dir

      If StrFile <> "" Then

         Set fso = CreateObject("Scripting.FileSystemObject")
         Set fil = fso.GetFile(strPath & StrFile)
         Set txt = fil.OpenAsTextStream(1)
         strtxt = txt.ReadAll
         txt.Close
         dataArray = Split(strtxt, vbLf)
         j = 1
         strExec = ""
         For Each strline In dataArray
            j = j + 1

            If Left(strline, 26) = "MAXL> execute calculation " Then
               strScript = Trim(Mid(strline, 27, 5000))
            End If

            If Left(strline, 18) = " OK/INFO - 1012579" Then
               strExec = Replace(strline, " OK/INFO - 1012579 - Total Calc Elapsed Time for ", "")

               If InStr(strScript, "BSCAP'.'ConsFX'") Then
                  strRange = "B" & i
                  Sheets("LogLoad").Range(strRange) = Replace(Replace(strExec, "] seconds.", ""),
"[ConsFX.csc] : [", "") * 1 / 60
               End If

               If InStr(strScript, "PANDL'.'ConsFX'") Then
                  strRange = "C" & i
                  Sheets("LogLoad").Range(strRange) = Replace(Replace(strExec, "] seconds.", ""),
"[ConsFX.csc] : [", "") * 1 / 60
               End If

               If InStr(strScript, "BS_OTHIS'.'ConsFX'") Then
                  strRange = "D" & i
                  Sheets("LogLoad").Range(strRange) = Replace(Replace(strExec, "] seconds.", ""),
"[ConsFX.csc] : [", "") * 1 / 60
               End If

               If InStr(strScript, "REV_CST'.'ConsFX'") Then
                  strRange = "E" & i
                  Sheets("LogLoad").Range(strRange) = Replace(Replace(strExec, "] seconds.", ""),
"[ConsFX.csc] : [", "") * 1 / 60
               End If

               If InStr(strScript, "FUNCSPND'.'ConsFX'") Then
                  strRange = "F" & i
                  Sheets("LogLoad").Range(strRange) = Replace(Replace(strExec, "] seconds.", ""),
"[ConsFX.csc] : [", "") * 1 / 60
               End If

            End If
         Next
         Set fso = Nothing
      End If
      i = i + 1
   Loop
End Sub
```

Hyperion Focus 17

Monitoring EPM systems performance with Oracle Data Visualization

Part 1: Where is my data?

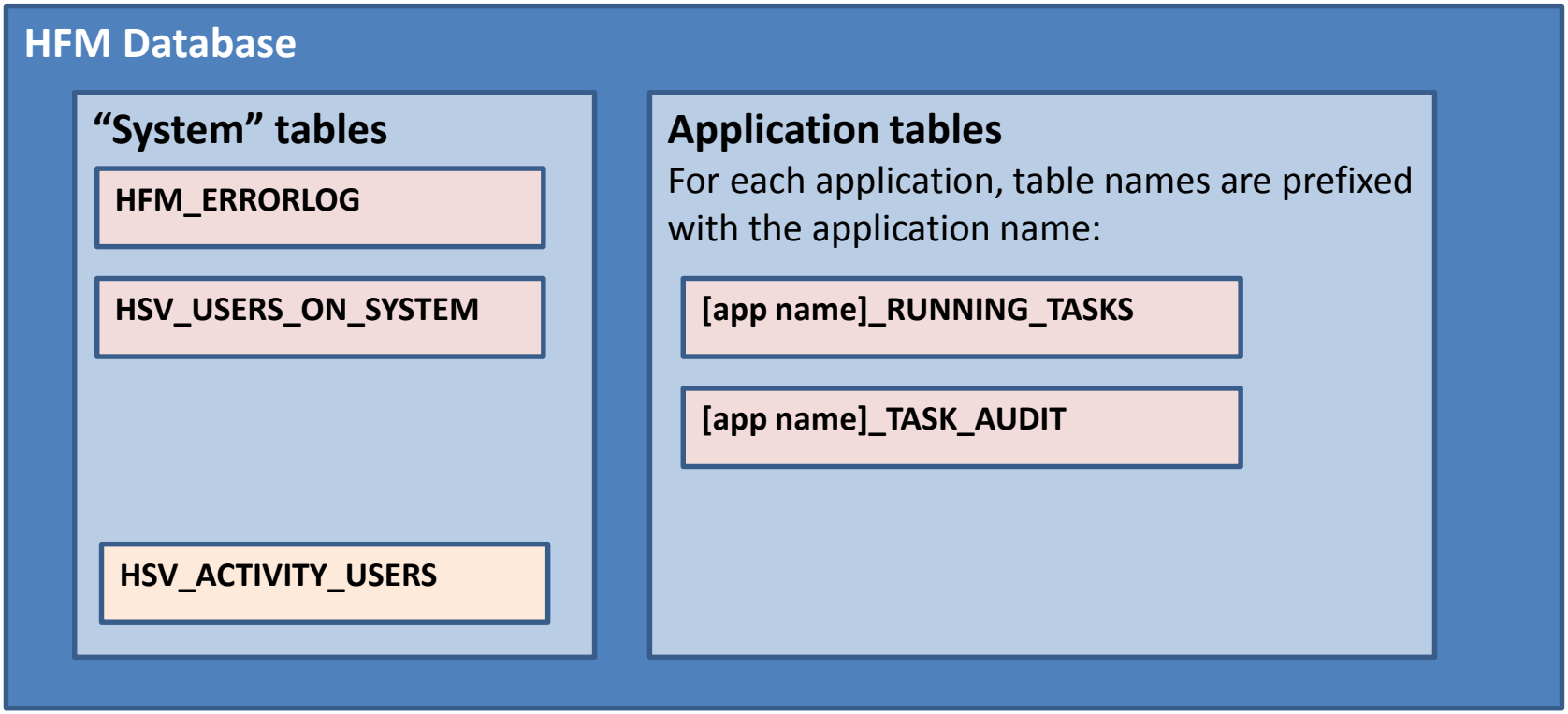
HFM

Overview

Part 1 – HFM Overview

HFM relies on a database that contains system tables, and application specific tables.

The tables we are interested in are:



Hyperion Focus 17

Monitoring EPM systems performance with Oracle Data Visualization

Part 1: Rounding up your source data

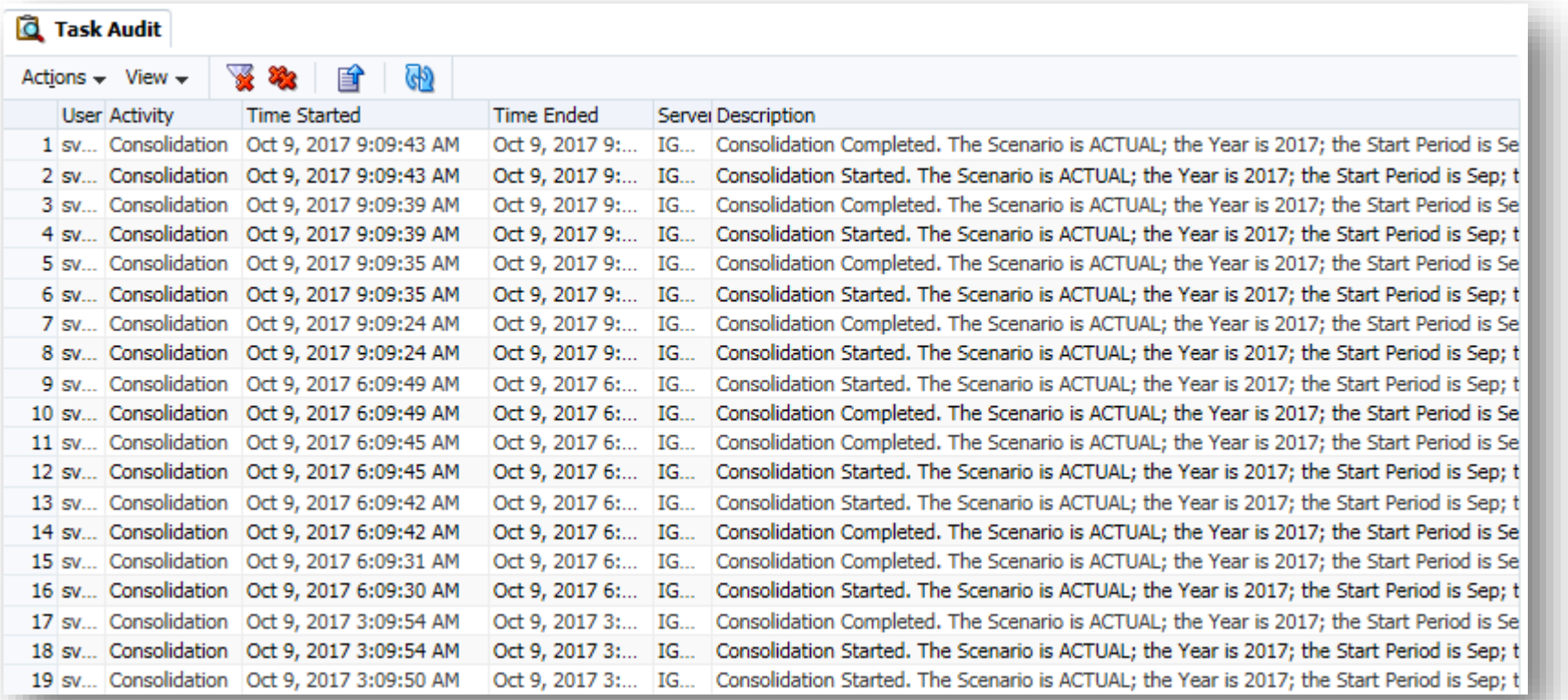
HFM

Consolidation Times

Part 1 – HFM Consolidation Times

HFM Consolidation information is captured by the system and stored in the database.

It can be accessed through Workspace in the Task Audit:



	User	Activity	Time Started	Time Ended	Server	Description
1	sv...	Consolidation	Oct 9, 2017 9:09:43 AM	Oct 9, 2017 9:...	IG...	Consolidation Completed. The Scenario is ACTUAL; the Year is 2017; the Start Period is Se
2	sv...	Consolidation	Oct 9, 2017 9:09:43 AM	Oct 9, 2017 9:...	IG...	Consolidation Started. The Scenario is ACTUAL; the Year is 2017; the Start Period is Sep; t
3	sv...	Consolidation	Oct 9, 2017 9:09:39 AM	Oct 9, 2017 9:...	IG...	Consolidation Completed. The Scenario is ACTUAL; the Year is 2017; the Start Period is Se
4	sv...	Consolidation	Oct 9, 2017 9:09:39 AM	Oct 9, 2017 9:...	IG...	Consolidation Started. The Scenario is ACTUAL; the Year is 2017; the Start Period is Sep; t
5	sv...	Consolidation	Oct 9, 2017 9:09:35 AM	Oct 9, 2017 9:...	IG...	Consolidation Completed. The Scenario is ACTUAL; the Year is 2017; the Start Period is Se
6	sv...	Consolidation	Oct 9, 2017 9:09:35 AM	Oct 9, 2017 9:...	IG...	Consolidation Started. The Scenario is ACTUAL; the Year is 2017; the Start Period is Sep; t
7	sv...	Consolidation	Oct 9, 2017 9:09:24 AM	Oct 9, 2017 9:...	IG...	Consolidation Completed. The Scenario is ACTUAL; the Year is 2017; the Start Period is Se
8	sv...	Consolidation	Oct 9, 2017 9:09:24 AM	Oct 9, 2017 9:...	IG...	Consolidation Started. The Scenario is ACTUAL; the Year is 2017; the Start Period is Sep; t
9	sv...	Consolidation	Oct 9, 2017 6:09:49 AM	Oct 9, 2017 6:...	IG...	Consolidation Started. The Scenario is ACTUAL; the Year is 2017; the Start Period is Sep; t
10	sv...	Consolidation	Oct 9, 2017 6:09:49 AM	Oct 9, 2017 6:...	IG...	Consolidation Completed. The Scenario is ACTUAL; the Year is 2017; the Start Period is Se
11	sv...	Consolidation	Oct 9, 2017 6:09:45 AM	Oct 9, 2017 6:...	IG...	Consolidation Completed. The Scenario is ACTUAL; the Year is 2017; the Start Period is Se
12	sv...	Consolidation	Oct 9, 2017 6:09:45 AM	Oct 9, 2017 6:...	IG...	Consolidation Started. The Scenario is ACTUAL; the Year is 2017; the Start Period is Sep; t
13	sv...	Consolidation	Oct 9, 2017 6:09:42 AM	Oct 9, 2017 6:...	IG...	Consolidation Started. The Scenario is ACTUAL; the Year is 2017; the Start Period is Sep; t
14	sv...	Consolidation	Oct 9, 2017 6:09:42 AM	Oct 9, 2017 6:...	IG...	Consolidation Completed. The Scenario is ACTUAL; the Year is 2017; the Start Period is Se
15	sv...	Consolidation	Oct 9, 2017 6:09:31 AM	Oct 9, 2017 6:...	IG...	Consolidation Completed. The Scenario is ACTUAL; the Year is 2017; the Start Period is Se
16	sv...	Consolidation	Oct 9, 2017 6:09:30 AM	Oct 9, 2017 6:...	IG...	Consolidation Started. The Scenario is ACTUAL; the Year is 2017; the Start Period is Sep; t
17	sv...	Consolidation	Oct 9, 2017 3:09:54 AM	Oct 9, 2017 3:...	IG...	Consolidation Completed. The Scenario is ACTUAL; the Year is 2017; the Start Period is Se
18	sv...	Consolidation	Oct 9, 2017 3:09:54 AM	Oct 9, 2017 3:...	IG...	Consolidation Started. The Scenario is ACTUAL; the Year is 2017; the Start Period is Sep; t
19	sv...	Consolidation	Oct 9, 2017 3:09:50 AM	Oct 9, 2017 3:...	IG...	Consolidation Started. The Scenario is ACTUAL; the Year is 2017; the Start Period is Sep; t

Part 1 – HFM Consolidation Times

Workspace allows to extract this information into a text file.

We could certainly use this as a source, however, this would require to re-export through Workspace each time we want to refresh our dataset.

We will see how to query the same information from the database.

The main table for this is [app name]_TASK_AUDIT:

```
SELECT TOP (100)
       [strGUID]
      ,[ActivityUserID]
      ,[ActivitySessionID]
      ,[ActivityCode]
      ,[ServerName]
      ,[AppName]
      ,[StartTime]
      ,[EndTime]
      ,[strDescription]
      ,[strModuleName]
  FROM [app name]_TASK_AUDIT
```

Results | Messages

	strGUID	ActivityUserID	ActivitySessionID	ActivityCode	ServerName	AppName	StartTime	EndTime	strDescription	strModuleName
1	000084D358D94346BCE9D1D30F831F80	2117695424	-2122683309	31	IGREPMNDC...	HYPFM...	42799.8822916667	42799.8822916667		#@RID@#25
2	0000D2A40AB445DA83553243E885972C	-2013834112	925633088	31	IGREPMNDC...	HYPFM...	42527.2025347222	42527.2025347222		#@RID@#25
3	0000E852800D4D76B02502ACB08B8A7D	579123516	-664451564	29	IGREPMNDC...	HYPFM...	42678.1124652778	42678.1124652778		#@RID@#25
4	0000F599292846DF922E1267C582CC55	2117695424	201469492	29	IGREPMNDC...	HYPFM...	42890.2954398148	42890.2954398148		#@RID@#25
5	00017A0EE05E4EAB98A64756DEB9ADE9	-142197528	897293013	29	IGREPMNDC...	HYPFM...	42506.5887731481	42506.5887731481		#@RID@#25
6	000263C2B7D64B15854367D262F96BC8	2117695424	8001506	31	IGREPMNDC...	HYPFM...	42985.9283449074	42985.9283449074		#@RID@#25

Part 1 – HFM Consolidation Times

There are a couple of transformations to perform to get the information in a more useable format.

UserID: the user IDs are identifiers and need to be matched to table [HSV_ACTIVITY_USERS] to display the username.

```
SELECT TOP (100)
      [strGUID]
     ,[ActivityUserID]
     ,[sUserName]
     ,[ActivitySessionID]
     ,[ActivityCode]
     ,[ServerName]
     ,[AppName]
     ,[StartTime]
     ,[EndTime]
     ,[strDescription]
     ,[strModuleName]
FROM [HYP-HFM].[dbo].[HYPFMR12_TASK_AUDIT]

LEFT OUTER JOIN [HSV_ACTIVITY_USERS]
ON [HSV_ACTIVITY_USERS].[lUserID] = [ActivityUserID]
```

Part 1 – HFM Consolidation Times

Timestamps: Start and End time timestamps are stored as numbers (floats), representing the number of days since January 1st, 1900. Casting to datetime and adjusting for timezone difference makes this much nicer:

```
,CAST([StartTime] - 2.166666666606034 as datetime) AS StartTimestamp
,CAST([EndTime] - 2.166666666606034 as datetime) AS EndTimestamp
```

StartTime	StartTimestamp	EndTime	EndTimestamp
43010.927025463	2017-10-02 18:14:55.000	43010.927025463	2017-10-02 18:14:55.000
43010.9270138889	2017-10-02 18:14:54.000	43010.9270138889	2017-10-02 18:14:54.000
43010.9270138889	2017-10-02 18:14:54.000	43010.9270138889	2017-10-02 18:14:54.000
43010.9237962963	2017-10-02 18:10:16.000	43010.9237962963	2017-10-02 18:10:16.000
43010.9236689815	2017-10-02 18:10:05.000	43010.9236689815	2017-10-02 18:10:05.000
43010.9236689815	2017-10-02 18:10:05.000	43010.9236689815	2017-10-02 18:10:05.000
43010.9235763889	2017-10-02 18:09:57.000	43010.9235763889	2017-10-02 18:09:57.000
43010.9235648148	2017-10-02 18:09:56.000	43010.9235648148	2017-10-02 18:09:56.000
43010.9234606481	2017-10-02 18:09:47.000	43010.9234606481	2017-10-02 18:09:47.000
43010.9234606481	2017-10-02 18:09:47.000	43010.9234606481	2017-10-02 18:09:47.000
43010.9234606481	2017-10-02 18:09:47.000	43010.9234606481	2017-10-02 18:09:47.000

Part 1 – HFM Consolidation Times

Filtering records for Consolidation only:

The table contains all types of events. In this case, we are only interested in Consolidation (other events could be used for further analysis), which Activity Code is 4.

(Refer to next page for the table of Activity Codes)

Furthermore, two records are created, one when Consolidation start, the other when it completes. We will only use the one marking completion.

```
WHERE ActivityCode = 4
  AND strDescription Like 'Consolidation Completed%'
```

Part 1 – HFM Consolidation Times

Table of HFM Activity codes:

Code	Description
0	Idle
1	RulesLoad
2	RulesScan
3	RulesExtract
4	Consolidation
5	ChartLogic
6	Translation
7	CustomLogic
8	Allocate
9	DataLoad
10	DataExtract
11	DataExtractviaHAL
12	DataEntry
13	DataRetrieval
14	DataClear
15	DataCopy
16	JournalEntry
17	JournalRetrieval
18	JournalPosting
19	JournalUnposting
20	JournalTemplateEntry

Code	Description
21	MetadataLoad
22	MetadataExtract
23	MemberListLoad
24	MemberListScan
25	MemberListExtract
26	SecurityLoad
27	SecurityScan
28	SecurityExtract
29	Logon
30	LogonFailure
31	Logoff
32	External
33	MetadataScan
34	DataScan
35	ExtendedAnalyticsExport
36	ExtendedAnalyticsSchemaDelete
37	TransactionsLoad
38	TransactionsExtract
39	DocumentAttachments
40	DocumentDetachments
41	CreateTransactions

Code	Description
42	EditTransactions
43	DeleteTransactions
44	PostTransactions
45	UnpostTransactions
46	DeleteInvalidRecords
47	DataAuditPurged
48	TaskAuditPurged
49	PostAllTransactions
50	UnpostAllTransactions
51	DeleteAllTransactions
52	UnmatchAllTransactions
53	AutoMatchbyID
54	AutoMatchbyAccount
55	IntercompanyMatchingReportbyID
56	IntercompanyMatchingReportbyAcct
57	IntercompanyTransactionReport
58	ManualMatch
59	UnmatchSelected
60	ManageICPeriods
61	Lock/UnlockICEntities
62	ManageICReasonCodes
63	Null

Part 1 – HFM Consolidation Times

Remove unnecessary text from the Description field:

strDescription
Consolidation Completed. The Scenario is ACTUAL; the Year is 2017; the Start Period is Sep; the End Period is Sep; the Entity is CB...

```
(replace(
 replace(
 replace(
 replace(
 replace(
 replace(
 replace(
 replace(
 replace(strDescription,'The Parent is ','')
         ,'ConsolidationType: ','')
         ,'The Entity is ','')
         ,'The End Period is ','')
         ,'The Start Period is ','')
         ,'The Year is ','')
         ,'The Scenario is ','')
         ,'Consolidation Completed. ','')
         ,'Consolidation Completed','')
) AS POV
```

POV
ACTUAL; 2017; Sep; Sep; CBC_TAX.; IMPACTED

Part 1 – HFM Consolidation Times

Transform the POV field into separate columns:

We will perform this parsing by using the following trick (the old version of T-SQL I use does not have string parsing):

- Transform the Error Info field by replacing ';' with XML tags

- Convert it to an xml field

- Extract the different values by XML lookups

This generates an XML field with multiple records, so will need to join tables to build our output.

The final query is provided on the next page, and the sample output follows

Part 1 – HFM Consolidation Times

Query:

```
SELECT [sUserName]
      ,[ServerName]
      ,[AppName]
      ,StartTS
      ,EndTS
      ,Duration_s
      ,Scenario=n.v.value('/e[1]','varchar(20)')
      ,Year=n.v.value('/e[2]','varchar(20)')
      ,StartPer=n.v.value('/e[3]','varchar(20)')
      ,EndPer=n.v.value('/e[4]','varchar(20)')
      ,Entity=Replace(n.v.value('/e[5]','varchar(20)'),'.','')
      ,TypeOrPar=Replace(n.v.value('/e[6]','varchar(20)'),'.','')
      ,Type=Replace(n.v.value('/e[7]','varchar(20)'),'.','')
FROM
((SELECT [sUserName]
        ,[ServerName]
        ,[AppName]
        ,CONVERT(datetime, (StartTime - 2.166666666693345)) AS StartTS
        ,CONVERT(datetime, (EndTime - 2.166666666693345)) AS EndTS
        ,DATEDIFF(SECOND,CONVERT(datetime, (StartTime - 2.166666666693345)),CONVERT(datetime, (EndTime - 2.166666666693345))) AS Duration_s
        ,(replace(replace(replace(replace(replace(replace(replace(replace(strDescription,'The Parent is ',''),'ConsolidationType: ',''),'The Entity is ',''),'The End Period is ',''),
                'The Start Period is ',''),'The Year is ',''),'The Scenario is ',''),'Consolidation Completed. ',''),'Consolidation Completed','')) AS POV
 FROM [HYPFMR12_TASK_AUDIT]

 LEFT OUTER JOIN [HSV_ACTIVITY_USERS] ON [HSV_ACTIVITY_USERS].[lUserID] = [ActivityUserID]

 WHERE ActivityCode = 4 AND strDescription Like 'Consolidation Completed%') AS tt
 CROSS APPLY
 (SELECT CAST('<e>'+Replace(tt.POV,'; ','</e><e>')+'</e>' AS XML) AS [POV_XML]) AS i
  CROSS APPLY
  i.[POV_XML].nodes('/') AS n(v))

ORDER BY StartTS DESC
```

Part 1 – HFM Consolidation Times

Sample output:

sUserName	ActivityCode	ServerName	AppName	StartTS	EndTS	Duration_s	POV
svc-hyper...	4	IGREPMNDC...	HYPFM...	2017-10-09 15:09:58.997	2017-10-09 15:09:58.997	0	ACTUAL; 2017; Sep; Sep; CBC_TAX.; IMPACTED
svc-hyper...	4	IGREPMNDC...	HYPFM...	2017-10-09 15:09:54.997	2017-10-09 15:09:54.997	0	ACTUAL; 2017; Sep; Sep; TAX.; IMPACTED
svc-hyper...	4	IGREPMNDC...	HYPFM...	2017-10-09 15:09:51.997	2017-10-09 15:09:51.997	0	ACTUAL; 2017; Sep; Sep; DGFN.; IMPACTED
svc-hyper...	4	IGREPMNDC...	HYPFM...	2017-10-09 15:09:39.997	2017-10-09 15:09:39.997	0	ACTUAL; 2017; Sep; Sep; LEGAL.; IMPACTED
irglhc@I...	4	IGREPMNDC...	HYPFM...	2017-10-09 12:37:57.997	2017-10-09 12:42:26.000	269	ACTUAL; 2017; Sep; Sep; DGFN.; IMPACTED
svc-hyper...	4	IGREPMNDC...	HYPFM...	2017-10-09 12:19:21.000	2017-10-09 12:19:54.997	33	ACTUAL; 2017; Sep; Sep; CBC_TAX.; IMPACTED
svc-hyper...	4	IGREPMNDC...	HYPFM...	2017-10-09 12:19:04.997	2017-10-09 12:19:15.997	11	ACTUAL; 2017; Sep; Sep; TAX.; IMPACTED
svc-hyper...	4	IGREPMNDC...	HYPFM...	2017-10-09 12:11:52.997	2017-10-09 12:18:46.997	414	ACTUAL; 2017; Sep; Sep; DGFN.; IMPACTED
svc-hyper...	4	IGREPMNDC...	HYPFM...	2017-10-09 12:09:29.997	2017-10-09 12:11:31.997	122	ACTUAL; 2017; Sep; Sep; LEGAL.; IMPACTED
svc-hyper...	4	IGREPMNDC...	HYPFM...	2017-10-09 09:09:43.000	2017-10-09 09:09:43.000	0	ACTUAL; 2017; Sep; Sep; CBC_TAX.; IMPACTED

Hyperion Focus 17

Monitoring EPM systems performance with Oracle Data Visualization

Part 1: Where is my data?

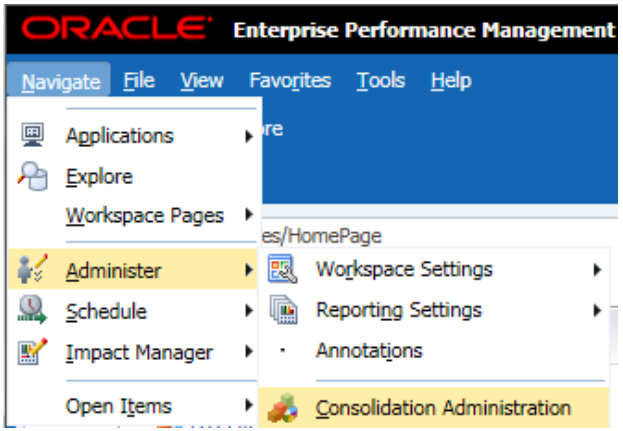
HFM

System Information

Part 1 – HFM System Information

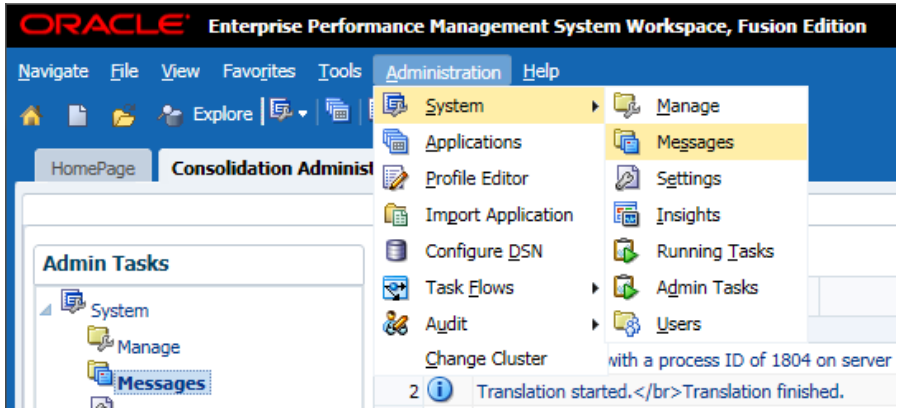
HFM records automatically System Information.

This information can be accessed from the workspace:



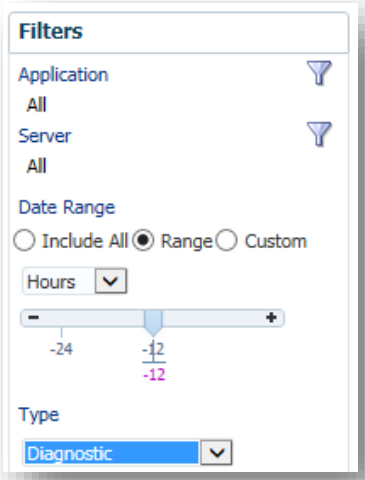
Navigate => Administer
=> Consolidation Administration

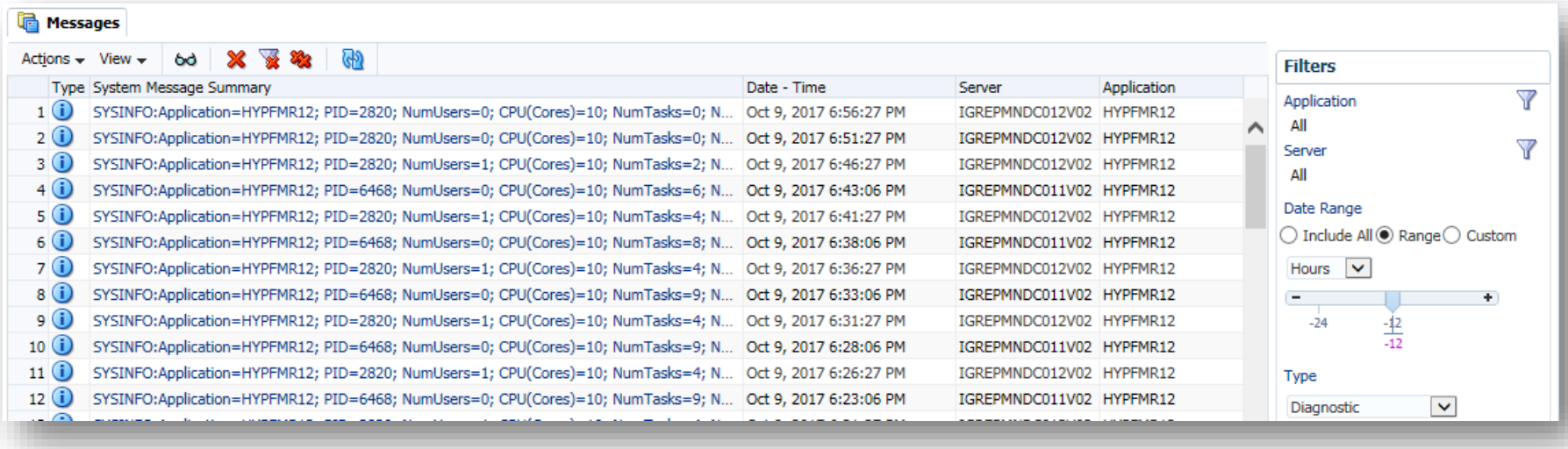
Once in Consol. Administration, access:
Administration => System
=> Messages



Part 1 – HFM System Information

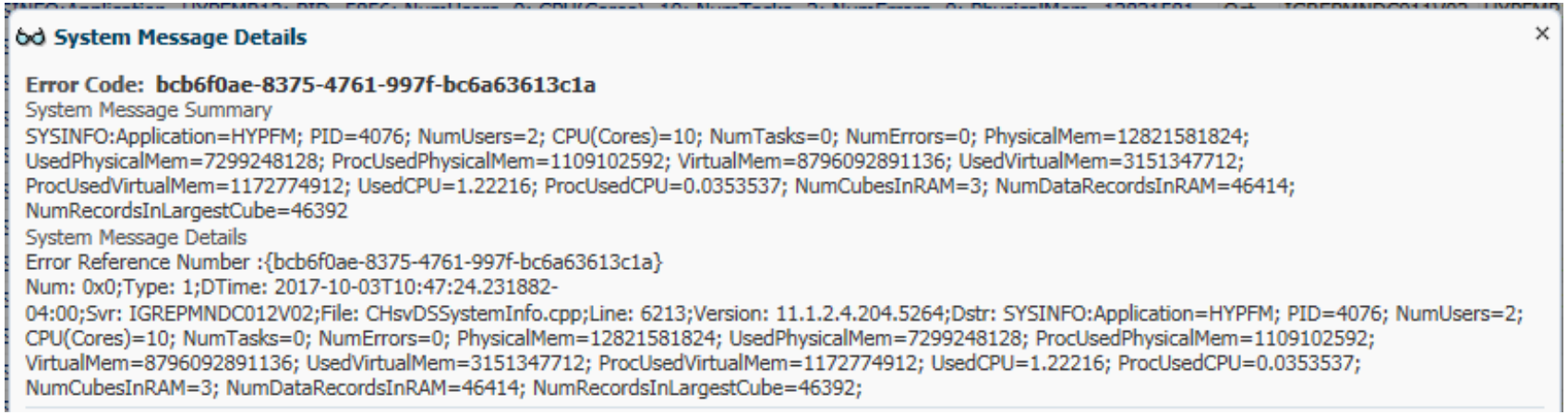
Select Diagnostic messages:





Part 1 – HFM System Information

Clicking on a message displays its full contents:



That's what I call some good information!

Now, if only it could be fed into a data visualization tool…

Part 1 – HFM System Information

Where can we find that?

Let's take a look in the database:

```
SELECT TOP (10)
       [sGUID]
      ,[lLogType]
      ,[dTimeStamp]
      ,[sServerName]
      ,[sAppName]
      ,[sXMLError]
  FROM [HFM_ERRORLOG]
  WHERE lLogType = 6
  ORDER BY dTimeStamp DESC
```

Results | Messages

	sGUID	lLogType	dTimeStamp	sServerName	sAppName	sXMLError
1	bcb6f0ae-8375-4761-997f-bc6a63613c1a	6	43011.6162526838	IGREPMNDC012V02	HYPFM	<?xml version="1.0" ?><EStr><Ref>{bcb6f0ae-8375-4761-997f-bc6a63613c1a}</Ref><AppNam
2	ef002645-48a3-4ed4-8cc9-12215c5587b4	6	43011.6159655889	IGREPMNDC012V02	HYPFMR12	<?xml version="1.0" ?><EStr><Ref>{ef002645-48a3-4ed4-8cc9-12215c5587b4}</Ref><AppNam
3	682a22be-1d0b-4e3e-b69c-88e6917bf681	6	43011.6157121643	IGREPMNDC011V02	HYPFMR12	<?xml version="1.0" ?><EStr><Ref>{682a22be-1d0b-4e3e-b69c-88e6917bf681}</Ref><AppNam
4	f08d55b7-75b6-4da3-a1ca-527cfa6dbe41	6	43011.612779825	IGREPMNDC012V02	HYPFM	<?xml version="1.0" ?><EStr><Ref>{f08d55b7-75b6-4da3-a1ca-527cfa6dbe41}</Ref><AppNam
5	355af811-7a01-4f2a-98be-9707469cc9ca	6	43011.6124928459	IGREPMNDC012V02	HYPFMR12	<?xml version="1.0" ?><EStr><Ref>{355af811-7a01-4f2a-98be-9707469cc9ca}</Ref><AppNam
6	96a8185f-7d08-4e7a-869d-14e0b0f318f0	6	43011.6122392476	IGREPMNDC011V02	HYPFMR12	<?xml version="1.0" ?><EStr><Ref>{96a8185f-7d08-4e7a-869d-14e0b0f318f0}</Ref><AppName
7	d6ddab4b-3d9c-4825-bf48-d7fb70896b5e	6	43011.6093068505	IGREPMNDC012V02	HYPFM	<?xml version="1.0" ?><EStr><Ref>{d6ddab4b-3d9c-4825-bf48-d7fb70896b5e}</Ref><AppNam
8	62e36f42-9ada-4c85-be7d-6682dbee4d7d	6	43011.6090198713	IGREPMNDC012V02	HYPFMR12	<?xml version="1.0" ?><EStr><Ref>{62e36f42-9ada-4c85-be7d-6682dbee4d7d}</Ref><AppNan
9	a3fa5358-6a1c-4588-a3d0-b05cb893bf74	6	43011.6087662152	IGREPMNDC011V02	HYPFMR12	<?xml version="1.0" ?><EStr><Ref>{a3fa5358-6a1c-4588-a3d0-b05cb893bf74}</Ref><AppNam
10	470fdb53-e222-40e3-a418-cf9d1ec3e710	6	43011.6058339338	IGREPMNDC012V02	HYPFM	<?xml version="1.0" ?><EStr><Ref>{470fdb53-e222-40e3-a418-cf9d1ec3e710}</Ref><AppNam

But will require a bit of work:

- **Reformating the Timestamps**
- **Transforming the "XMLError" message**

Part 1 – HFM System Information

Transforming the timestamp into a datetime format

Timestamps are stored as numbers (floats), representing the number of days since January 1st, 1900.
Casting to datetime and adjusting for timezone difference makes this much nicer:

```
CAST([dTimeStamp] - 2.166666666606034 as datetime) AS TimeStamp
```

dTimeStamp	TimeStamp
43011.7137156468	2017-10-03 13:07:45.030
43011.7132007162	2017-10-03 13:07:00.540
43011.7129523842	2017-10-03 13:06:39.083

Part 1 – HFM System Information

We can generate additional fields derived from the timestamp:

- the date in a text format (will allow us to do day averages)

- the month number (month averages)

- Week day number

- Hour

```
 CAST(CONVERT(date,[TimeStamp]) AS nvarchar(10)) AS strDate
,DATEPART(MONTH,[TimeStamp]) AS numMth
,DATEPART(dw,[TimeStamp]) numWeekDay
,DATEPART(HOUR,[TimeStamp]) AS numHour
```

These can be very useful to generate more analytics.

We will see later that Oracle DV also has the capability to derive different fields from the data source.

Part 1 – HFM System Information

The next step is to transform the message field.

This is an XML field, so let's first take a look at an example to understand the structure:

The good news is that it's well structured, and SQL provides functions to process XML

```
<?xml version="1.0" ?>
<EStr>
    <Ref>{7fbe1fba-b63b-4fde-9f7c-37b7dccd20e4}</Ref>
    <AppName>HYPFM</AppName>
    <User></User>
    <DBUpdate>1</DBUpdate>
    <ESec>
        <Num>0</Num>
        <Description></Description>
        <Type>1</Type>
        <DTime>2017-10-03T12:07:24.876882-04:00</DTime>
        <Svr>IGREPMNDC012V02</Svr>
        <File>CHsvDSSystemInfo.cpp</File>
        <Line>6213</Line>
        <Ver>11.1.2.4.204.5264</Ver>
        <DStr>SYSINFO:Application=HYPFM; PID=4076; NumUsers=0;
        <ExErr></ExErr>
        <PSec></PSec>
    </ESec>
</EStr>
```

This is the timezone we had to adjust for

Part 1 – HFM System Information

To use SQL XML functions, we need to transform this field from its text format to XML.

```
SELECT TOP 10
       [sServerName]
      ,[sAppName]
      ,dTimeStamp
      ,CAST([dTimeStamp] - 2.166666666606034 as datetime) AS TimeStamp
      ,(CAST([sXMLError] as XML)).value('(/EStr//AppName/node())[1]', 'varchar(10)')  AS XML_App
      ,(CAST([sXMLError] as XML)).value('(/EStr//ESec//Svr/node())[1]', 'varchar(20)')  AS XML_Svr
      ,(CAST([sXMLError] as XML)).value('(/EStr//ESec//DTime/node())[1]', 'varchar(50)')  AS XML_TS
      ,(CAST([sXMLError] as XML)).value('(/EStr//ESec//Ver/node())[1]', 'varchar(50)')  AS XML_Ver
      ,(CAST([sXMLError] as XML)).value('(/EStr//ESec//DStr/node())[1]', 'varchar(500)')  AS XML_Info
FROM [HFM_ERRORLOG]
  WHERE lLogType = 6
  ORDER BY dTimeStamp DESC
```

Part 1 – HFM System Information

Output:

Results | Messages

	sServerName	sAppName	dTimeStamp	TimeStamp	XML_App	XML_Svr	XML_TS	XML_Ver	XML_Info
1	IGREPMNDC012V02	HYPFM	43011.7171884477	2017-10-03 13:12:45.080	HYPFM	IGREPMNDC012V02	2017-10-03T13:12:45.081882-04:00	11.1.2.4.204.5264	SYSINFO:Application=HYPFM; PID=3912; NumUsers=1; CPU(Cores)=10...
2	IGREPMNDC012V02	HYPFMR12	43011.7166738644	2017-10-03 13:12:00.620	HYPFMR12	IGREPMNDC012V02	2017-10-03T13:12:00.621882-04:00	11.1.2.4.204.5264	SYSINFO:Application=HYPFMR12; PID=3536; NumUsers=5; CPU(Cores)...
3	IGREPMNDC011V02	HYPFMR12	43011.7164250694	2017-10-03 13:11:39.123	HYPFMR12	IGREPMNDC011V02	2017-10-03T13:11:39.125992-04:00	11.1.2.4.204.5264	SYSINFO:Application=HYPFMR12; PID=5856; NumUsers=4; CPU(Cores)...
4	IGREPMNDC011V02	HYPFM	43011.7162592129	2017-10-03 13:11:24.793	HYPFM	IGREPMNDC011V02	2017-10-03T13:11:24.795992-04:00	11.1.2.4.204.5264	SYSINFO:Application=HYPFM; PID=5780; NumUsers=1; CPU(Cores)=10...
5	IGREPMNDC012V02	HYPFM	43011.7137156468	2017-10-03 13:07:45.030	HYPFM	IGREPMNDC012V02	2017-10-03T13:07:45.031882-04:00	11.1.2.4.204.5264	SYSINFO:Application=HYPFM; PID=3912; NumUsers=1; CPU(Cores)=10...
6	IGREPMNDC012V02	HYPFMR12	43011.7132007162	2017-10-03 13:07:00.540	HYPFMR12	IGREPMNDC012V02	2017-10-03T13:07:00.541882-04:00	11.1.2.4.204.5264	SYSINFO:Application=HYPFMR12; PID=3536; NumUsers=7; CPU(Cores)...
7	IGREPMNDC011V02	HYPFMR12	43011.7129523842	2017-10-03 13:06:39.083	HYPFMR12	IGREPMNDC011V02	2017-10-03T13:06:39.085992-04:00	11.1.2.4.204.5264	SYSINFO:Application=HYPFMR12; PID=5856; NumUsers=5; CPU(Cores)...
8	IGREPMNDC011V02	HYPFM	43011.7127865856	2017-10-03 13:06:24.760	HYPFM	IGREPMNDC011V02	2017-10-03T13:06:24.760992-04:00	11.1.2.4.204.5264	SYSINFO:Application=HYPFM; PID=5780; NumUsers=0; CPU(Cores)=10...
9	IGREPMNDC012V02	HYPFM	43011.7102430195	2017-10-03 13:02:44.997	HYPFM	IGREPMNDC012V02	2017-10-03T13:02:44.996882-04:00	11.1.2.4.204.5264	SYSINFO:Application=HYPFM; PID=3912; NumUsers=1; CPU(Cores)=10...
10	IGREPMNDC012V02	HYPFMR12	43011.709727626	2017-10-03 13:02:00.467	HYPFMR12	IGREPMNDC012V02	2017-10-03T13:02:00.466882-04:00	11.1.2.4.204.5264	SYSINFO:Application=HYPFMR12; PID=3536; NumUsers=6; CPU(Cores)...

Much better!

But we are not quite there yet: the information is bundled in a single field.

Let's parse this into separate fields, so that we have the data we want to graph nicely laid out.

Part 1 – HFM System Information

We will perform this parsing by using the following trick (the old version of T-SQL I use does not have string parsing):

- Transform the Error Info field by replacing ';' with XML tags

- Convert it to an xml field

- Extract the different values by XML lookups

This generates an XML field with multiple records, so will need to join tables to build our output.

Once this is done, we will remove the extra text from each field, and cast the results to the appropriate numeric type.

Part 1 – HFM System Information

Here is the resulting query with select output fields (the query with all fields is provided later for reference):

```
SELECT [sServerName]
      ,[sAppName]
      ,[TimeStamp]
      ,CAST(CONVERT(date,[TimeStamp]) AS nvarchar(10)) AS strDate
      ,DATEPART(MONTH,[TimeStamp]) AS numMth
      ,DATEPART(dw,[TimeStamp]) numWeekDay
      ,DATEPART(HOUR,[TimeStamp]) AS numHour
      ,[NumUsers]=CAST(REPLACE(n.v.value('/e[3]','varchar(50)'),'NumUsers=','') AS int)
      ,[UsedPhysicalMem]=CAST(REPLACE(n.v.value('/e[8]','varchar(50)'),'UsedPhysicalMem=','') AS bigint)
      ,[UsedVirtualMem]=CAST(REPLACE(n.v.value('/e[11]','varchar(50)'),'UsedVirtualMem=','') AS bigint)
      ,[UsedCPU]=CAST(REPLACE(n.v.value('/e[13]','varchar(50)'),'UsedCPU=','') AS float)
      ,[NumDataRecordsInRAM]=CAST(REPLACE(n.v.value('/e[16]','varchar(50)'),'NumDataRecordsInRAM=','') AS bigint)
      ,[NumRecordsInLargestCube]=CAST(REPLACE(n.v.value('/e[17]','varchar(50)'),'NumRecordsInLargestCube=','') AS bigint)
FROM
(
    (SELECT
          [sServerName]
         ,[sAppName]
         ,CAST([dTimeStamp] - 2.166666666606034 as datetime) AS [TimeStamp]
         ,(CAST([sXMLError] as XML)).value('(/EStr//ESec//DStr/node())[1]', 'varchar(1000)') AS [XML_Info]
    FROM [HFM_ERRORLOG]
    WHERE [lLogType] = 6
    ORDER BY dTimeStamp DESC) AS tt
    CROSS APPLY
    (SELECT CAST('<e>'+Replace(tt.XML_Info,'; ','</e><e>')+'</e>' AS XML) AS [XML_Info_XML]) AS i
    CROSS APPLY
    i.[XML_Info_XML].nodes('/') AS n(v)
)
```

Part 1 – HFM System Information

And the output:

sServerName	sAppName	Timestamp	strDate	numMth	numWe...	numHour	NumU...	UsedPhysicalMem	UsedVirtualM...	UsedCPU	NumDataRecordsInRAM	NumRecordsInLargestCube
IGREPMN...	HYPFM...	2017-10-07 11:26:24.277	2017-10-07	10	7	11	0	4867387392	3095322624	0.322513	301571	66623
IGREPMN...	HYPFM...	2017-10-07 11:21:24.233	2017-10-07	10	7	11	0	4868861952	3095322624	0.354949	301571	66623
IGREPMN...	HYPFM...	2017-10-07 11:16:24.177	2017-10-07	10	7	11	0	4870545408	3095322624	0.590429	301571	66623
IGREPMN...	HYPFM...	2017-10-07 11:11:24.363	2017-10-07	10	7	11	0	4746604544	2387615744	6.3994	0	0
IGREPMN...	HYPFM...	2017-10-07 10:59:23.980	2017-10-07	10	7	10	0	4256387072	3093807104	0.312431	301571	66623
IGREPMN...	HYPFM...	2017-10-07 10:54:23.950	2017-10-07	10	7	10	0	4255072256	3093807104	0.32595	301571	66623
IGREPMN...	HYPFM...	2017-10-07 10:49:23.910	2017-10-07	10	7	10	0	4246134784	3093807104	0.63095	301571	66623
IGREPMN...	HYPFM...	2017-10-07 10:44:24.127	2017-10-07	10	7	10	0	4138749952	2394890240	6.3994	0	0
IGREPMN...	HYPFM...	2017-10-07 10:00:23.940	2017-10-07	10	7	10	0	4275634176	3080830976	0.441155	301571	66623
IGREPMN...	HYPFM...	2017-10-07 09:55:23.890	2017-10-07	10	7	9	0	4271968256	3088891904	0.295171	301571	66623
IGREPMN...	HYPFM...	2017-10-07 09:50:23.860	2017-10-07	10	7	9	0	4283547648	3088891904	0.376585	301571	66623
IGREPMN...	HYPFM...	2017-10-07 09:45:23.820	2017-10-07	10	7	9	0	4269178880	3088891904	0.613588	301571	66623
IGREPMN...	HYPFM...	2017-10-07 09:40:24.040	2017-10-07	10	7	9	0	4156055552	2388140032	6.3994	0	0
IGREPMN...	HYPFM...	2017-10-07 09:08:23.423	2017-10-07	10	7	9	0	4879261696	3097620480	0.357143	301571	66623
IGREPMN...	HYPFM...	2017-10-07 09:03:23.367	2017-10-07	10	7	9	0	4916715520	3097620480	0.51989	301571	66623
IGREPMN...	HYPFM...	2017-10-07 08:58:23.277	2017-10-07	10	7	8	0	4860772352	3097620480	0.363298	301571	66623
IGREPMN...	HYPFM...	2017-10-07 08:53:23.177	2017-10-07	10	7	8	0	4861227008	3097620480	0.421818	301571	66623
IGREPMN...	HYPFM...	2017-10-07 08:48:23.133	2017-10-07	10	7	8	0	4854956032	3091259392	0.614938	301571	66623
IGREPMN...	HYPFM...	2017-10-07 08:43:23.287	2017-10-07	10	7	8	0	4732264448	2388992000	9.5194	0	0

Part 1 – HFM System Information

For reference, this is the full final query that I use to generate the transformed data set for analytics:

```
SELECT
    [sServerName]
   ,[sAppName]
   ,TimeStamp
   ,CAST(CONVERT(date,[TimeStamp]) AS nvarchar(10)) AS strDate
   ,DATEPART(MONTH,[TimeStamp]) AS numMth
   ,DATEPART(dw,[TimeStamp]) numWeekDay
   ,DATEPART(HOUR,[TimeStamp]) AS numHour
   ,PID=REPLACE(n.v.value('/e[2]','varchar(50)'),'PID=','')
   ,NumUsers=CAST(REPLACE(n.v.value('/e[3]','varchar(50)'),'NumUsers=','') AS int)
   ,CPU_Cores=REPLACE(n.v.value('/e[4]','varchar(50)'),'CPU(Cores)=','')
   ,NumTasks=CAST(REPLACE(n.v.value('/e[5]','varchar(50)'),'NumTasks=','') AS int)
   ,NumErrors=CAST(REPLACE(n.v.value('/e[6]','varchar(50)'),'NumErrors=','') AS int)
   ,PhysicalMem=CAST(REPLACE(n.v.value('/e[7]','varchar(50)'),'PhysicalMem=','') AS bigint)
   ,UsedPhysicalMem=CAST(REPLACE(n.v.value('/e[8]','varchar(50)'),'UsedPhysicalMem=','') AS bigint)
   ,ProcUsedPhysicalMem=CAST(REPLACE(n.v.value('/e[9]','varchar(50)'),'ProcUsedPhysicalMem=','')
                                        AS bigint)
   ,VirtualMem=CAST(REPLACE(n.v.value('/e[10]','varchar(50)'),'VirtualMem=','') AS bigint)
   ,UsedVirtualMem=CAST(REPLACE(n.v.value('/e[11]','varchar(50)'),'UsedVirtualMem=','') AS bigint)
   ,ProcUsedVirtualMem=CAST(REPLACE(n.v.value('/e[12]','varchar(50)'),'ProcUsedVirtualMem=','')
                                        AS bigint)
   ,UsedCPU=CAST(REPLACE(n.v.value('/e[13]','varchar(50)'),'UsedCPU=','') AS float)
   ,ProcUsedCPU=CAST(REPLACE(n.v.value('/e[14]','varchar(50)'),'ProcUsedCPU=','') AS float)
   ,NumCubesinRAM=CAST(REPLACE(n.v.value('/e[15]','varchar(50)'),'NumCubesinRAM=','') AS bigint)
   ,NumDataRecordsInRAM=CAST(REPLACE(n.v.value('/e[16]','varchar(50)'),'NumDataRecordsInRAM=','')
                                        AS bigint)
   ,NumRecordsInLargestCube=CAST(REPLACE(n.v.value('/e[17]','varchar(50)'),
                                        'NumRecordsInLargestCube=','') AS bigint)
FROM
(
   (SELECT
      [sServerName],
      [sAppName],
      CAST([dTimeStamp] - 2.166666666606034 as datetime)
         AS TimeStamp,
      (CAST([sXMLError] as XML)).value('(/EStr//ESec//DStr/node())[1]',
         'varchar(1000)') AS XML_Info
   FROM [HFM_ERRORLOG]
   WHERE lLogType = 6
   ORDER BY dTimeStamp DESC) AS tt
   CROSS APPLY (SELECT CAST('<e>'+Replace(tt.XML_Info,';'
         ,'</e><e>')+'</e>' AS XML) AS XML_Info_XML) AS I
   CROSS APPLY i.XML_Info_XML.nodes('/') AS n(v)
)
ORDER BY TimeStamp DESC
```

Part 1 – HFM System Information

We now have a query that outputs the data we will use for visualizations.

We can do one of the following:

- **Export and save to a spreadsheet.**
  This makes sense for ad-hoc analysis that may require data 'post-processing'.

- **Write the output to a table.**
  This has the benefit of better performance, but requires a process to refresh the data to the output table.

- **Create a database view**
  At this time, I have not been able to connect to views in Oracle DV due to a technical issue. In process of investigating.

Part 1 – HFM System Information

To output to a view:

- **Add 'CREATE VIEW view_name AS (' before the query**
- **Remove the ORDER BY Statement at the end and close the parenthesis**

```
CREATE VIEW vSYSINFO AS
(SELECT [sServerName]
      ,[sAppName]
      ,TimeStamp
```

To output to a table:

- **Add 'INTO table_name' after the SELECT section and before the FROM**

```
SELECT [sServerName]
      ,[sAppName]
      ,[Timestamp]
      ,CAST(CONVERT(date,[TimeStamp])
      ,DATEPART(MONTH,[TimeStamp]) AS
      ,DATEPART(dw,[TimeStamp]) numWe
      ,DATEPART(HOUR,[TimeStamp]) AS
      ,[NumUsers]=CAST(REPLACE(n.v.va
      ,[UsedPhysicalMem]=CAST(REPLACE
      ,[UsedVirtualMem]=CAST(REPLACE(
      ,[UsedCPU]=CAST(REPLACE(n.v.val
      ,[NumDataRecordsInRAM]=CAST(REP
      ,[NumRecordsInLargestCube]=CAST
INTO zSYSINFO_Full
FROM
```

Part 1 – What next?

Some areas to explore if your appetite is not satiated:

- **Planning and Essbase log files:**
  These could be scanned and relevant information extracted using a script and written to a database table

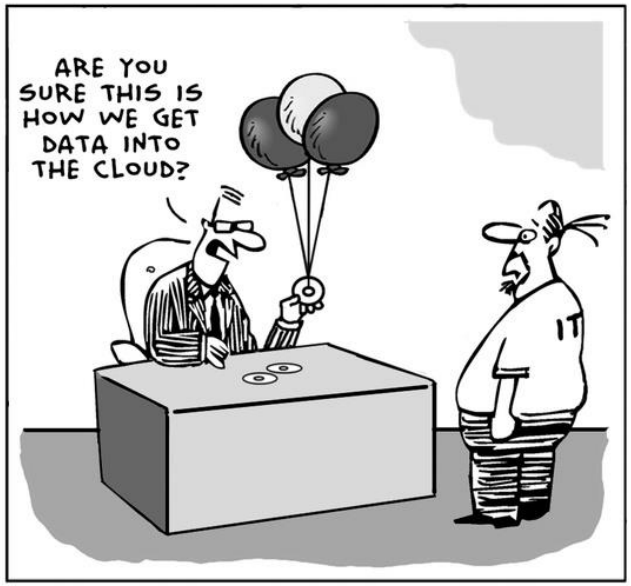
- **Server level metrics:**
  Depending on your technology, you can scan services, memory, CPU… and write that to a file or DB table
  (similar to what HFM SYSINFO provides for the HFM server)

If you are using other systems (ETL, Database), the same approach can be extended to them.

Hyperion Focus 17

Monitoring EPM systems performance with Oracle Data Visualization

Part 2: How do I connect to my data?



Part 2 – Game Plan

Now that we have our data, we want to connect to it in Oracle DV.

Because the process is streamlined and does not depend on the data content, we will look at two examples:

- **Connecting to a SQL Database table**
- **Connecting to an Excel Spreadsheet table**

My personal experience is primarily with these two types of connections.

Oracle DV offers a number of options, and keeps expanding available sources.
Some connectors are in beta release.
Please refer to Oracle DV documentation for further information.

Hyperion Focus 17

Monitoring EPM systems performance with Oracle Data Visualization

Part 2: How do I connect to my data?

Connecting to a SQL Table

Part 2 – Connecting to a SQL Table

Quick recap of what we will use:

- **A source HFM database that we have access to (in my case, on SQL Server, but Oracle DV can handle all kinds of databases)**
- **In this HFM database, a table named vSYSINFO_Full**

Results | Messages

	sServerName	sAppName	Timestamp	strDate	numMth	numWeekDay	numHour	NumUsers	UsedPhysicalMem	UsedVirtualMem	UsedCPU	NumDataRecordsInRAM	NumRecordsInLargestCube
1	IGREPMN...	HYPFM...	2017-10-07 11:26:24.277	2017-10-07	10	7	11	0	4867387392	3095322624	0.322513	301571	66623
2	IGREPMN...	HYPFM...	2017-10-07 11:21:24.233	2017-10-07	10	7	11	0	4868861952	3095322624	0.354949	301571	66623
3	IGREPMN...	HYPFM...	2017-10-07 11:16:24.177	2017-10-07	10	7	11	0	4870545408	3095322624	0.590429	301571	66623
4	IGREPMN...	HYPFM...	2017-10-07 11:11:24.363	2017-10-07	10	7	11	0	4746604544	2387615744	6.3994	0	0
5	IGREPMN...	HYPFM...	2017-10-07 10:59:23.980	2017-10-07	10	7	10	0	4256387072	3093807104	0.312431	301571	66623
6	IGREPMN...	HYPFM...	2017-10-07 10:54:23.950	2017-10-07	10	7	10	0	4255072256	3093807104	0.32595	301571	66623
7	IGREPMN...	HYPFM...	2017-10-07 10:49:23.910	2017-10-07	10	7	10	0	4246134784	3093807104	0.63095	301571	66623
8	IGREPMN...	HYPFM...	2017-10-07 10:44:24.127	2017-10-07	10	7	10	0	4138749952	2394890240	6.3994	0	0
9	IGREPMN...	HYPFM...	2017-10-07 10:00:23.940	2017-10-07	10	7	10	0	4275634176	3080830976	0.441155	301571	66623
10	IGREPMN...	HYPFM...	2017-10-07 09:55:23.890	2017-10-07	10	7	9	0	4271968256	3088891904	0.295171	301571	66623
11	IGREPMN...	HYPFM...	2017-10-07 09:50:23.860	2017-10-07	10	7	9	0	4283547648	3088891904	0.376585	301571	66623
12	IGREPMN...	HYPFM...	2017-10-07 09:45:23.820	2017-10-07	10	7	9	0	4269178880	3088891904	0.613588	301571	66623
13	IGREPMN...	HYPFM...	2017-10-07 09:40:24.040	2017-10-07	10	7	9	0	4156055552	2388140032	6.3994	0	0
14	IGREPMN...	HYPFM...	2017-10-07 09:08:23.423	2017-10-07	10	7	9	0	4879261696	3097620480	0.357143	301571	66623
15	IGREPMN...	HYPFM...	2017-10-07 09:03:23.367	2017-10-07	10	7	9	0	4916715520	3097620480	0.51989	301571	66623
16	IGREPMN...	HYPFM...	2017-10-07 08:58:23.277	2017-10-07	10	7	8	0	4860772352	3097620480	0.363298	301571	66623
17	IGREPMN...	HYPFM...	2017-10-07 08:53:23.177	2017-10-07	10	7	8	0	4861227008	3097620480	0.421818	301571	66623
18	IGREPMN...	HYPFM...	2017-10-07 08:48:23.133	2017-10-07	10	7	8	0	4854956032	3091259392	0.614938	301571	66623

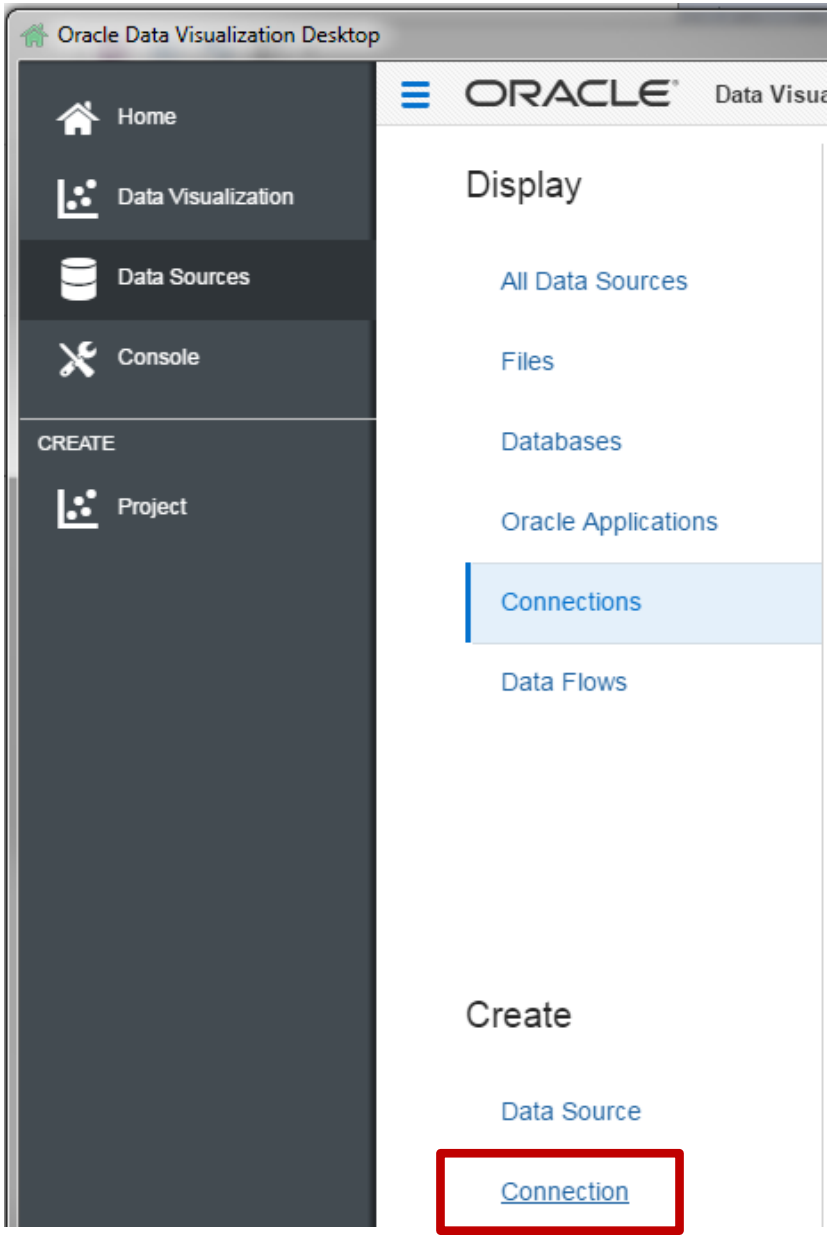
Over the next slides, we will go through the steps of creating the connection and setting it up for easy building of our visualizations.

Part 2 – Connecting to a SQL Table

Step 1: Creating the connection

In Oracle DV:

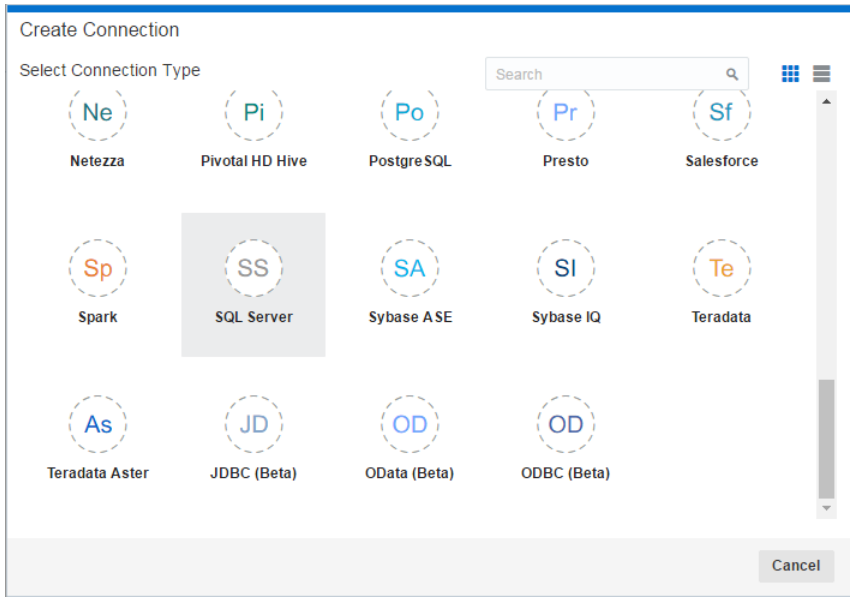
- **Navigate to “Data Sources”**
- **Click on Create Connection**



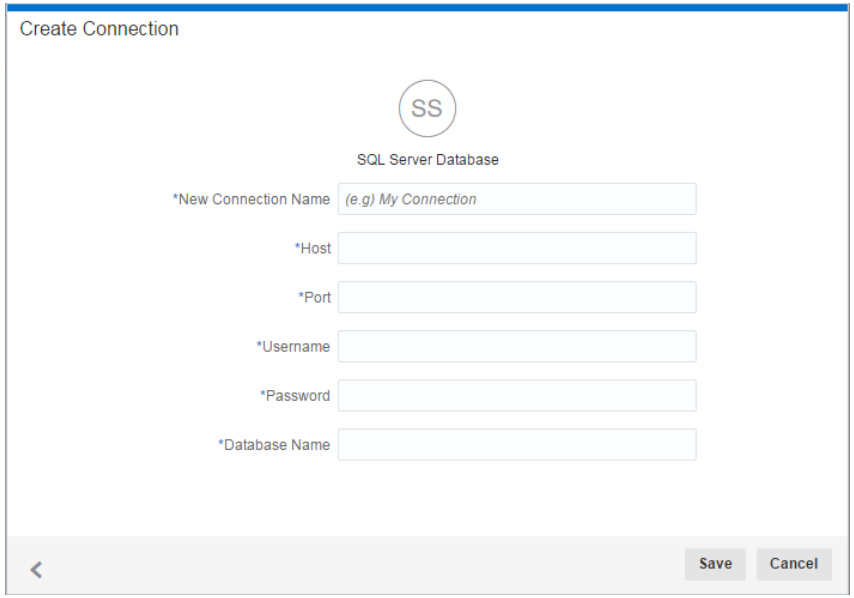
Part 2 – Connecting to a SQL Table

Step 1: Creating the connection

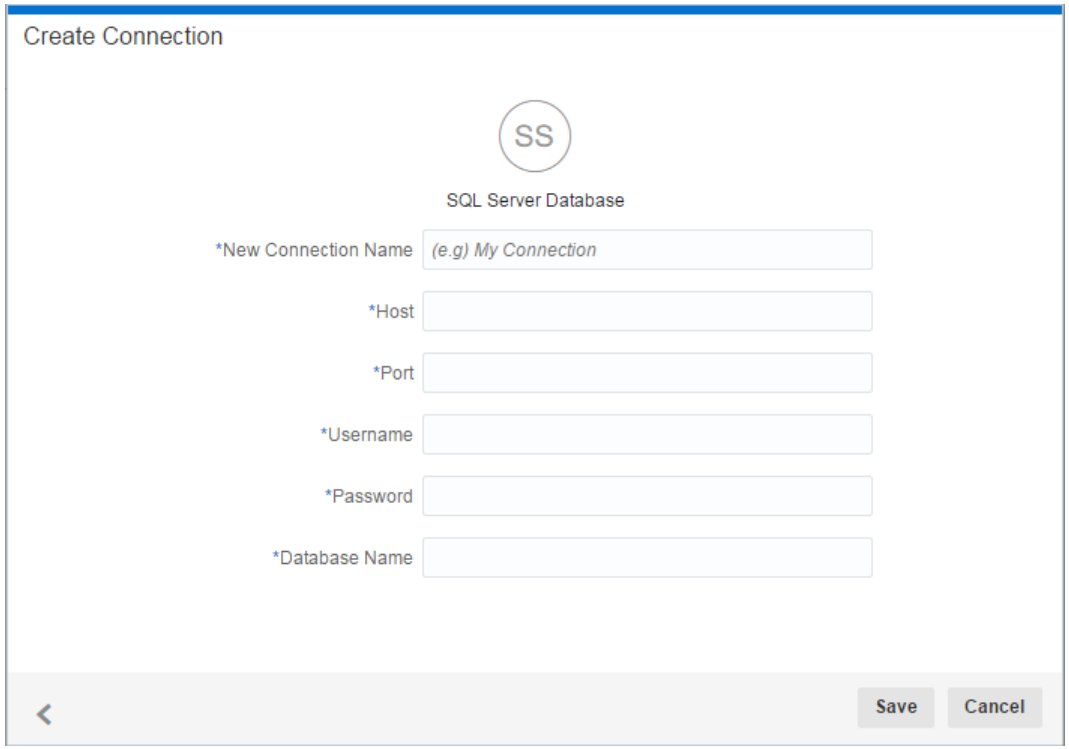
- **Select the Connection Type: (for me, SQL Server)**



- **Enter the connection settings:**



Part 2 – Connecting to a SQL Table



Connection name: this can be any name you want. Chose something that makes sense for this database. I chose “HFM”.

Host: the name of the database server.

Port: for SQL Server, this is typically 1433. Verify with your database administrator if this does not work.

Username and password: this must be a username and password with access to the database. In my experience, DB accounts work better.

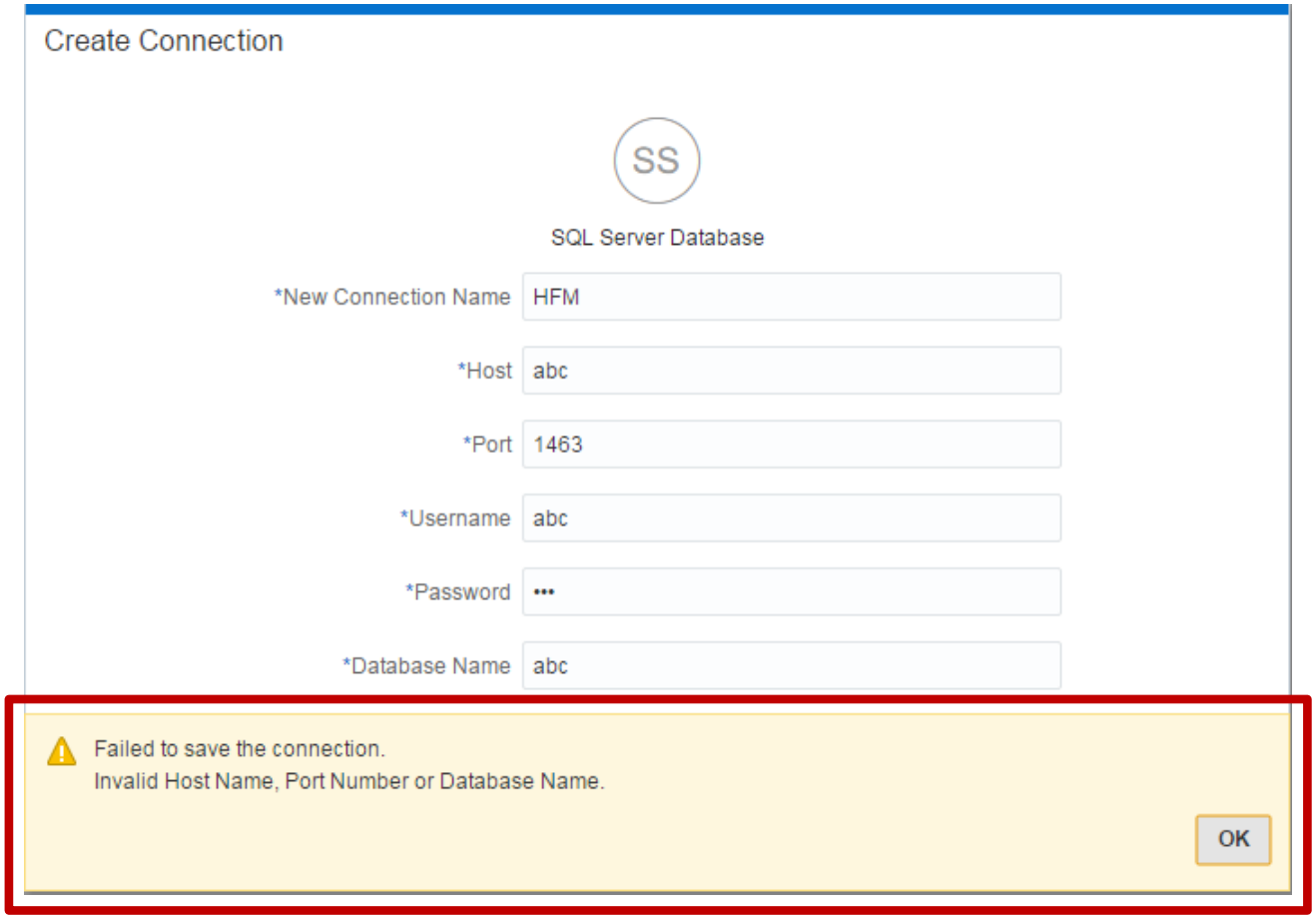
Database name: the name of the database.

If all works well, you will see a message “Connection created successfully”

Part 2 – Connecting to a SQL Table

Oracle DV validates the connection when saving.

If the settings are not correct, you will be notified and the connection will not be saved:



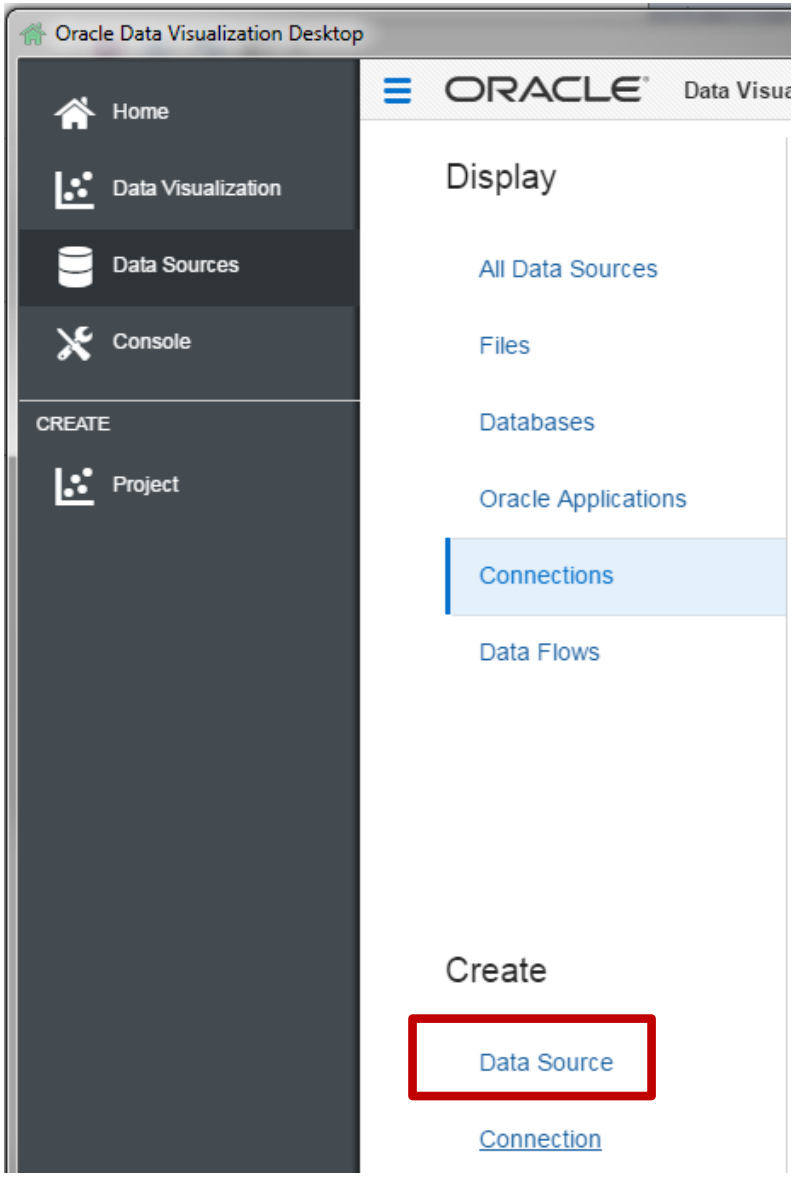
Part 2 – Connecting to a SQL Table

Step 2: Creating the data source

Once we have a connection, we can create a data source:

This can be done from the left pane by clicking on Create Data Source

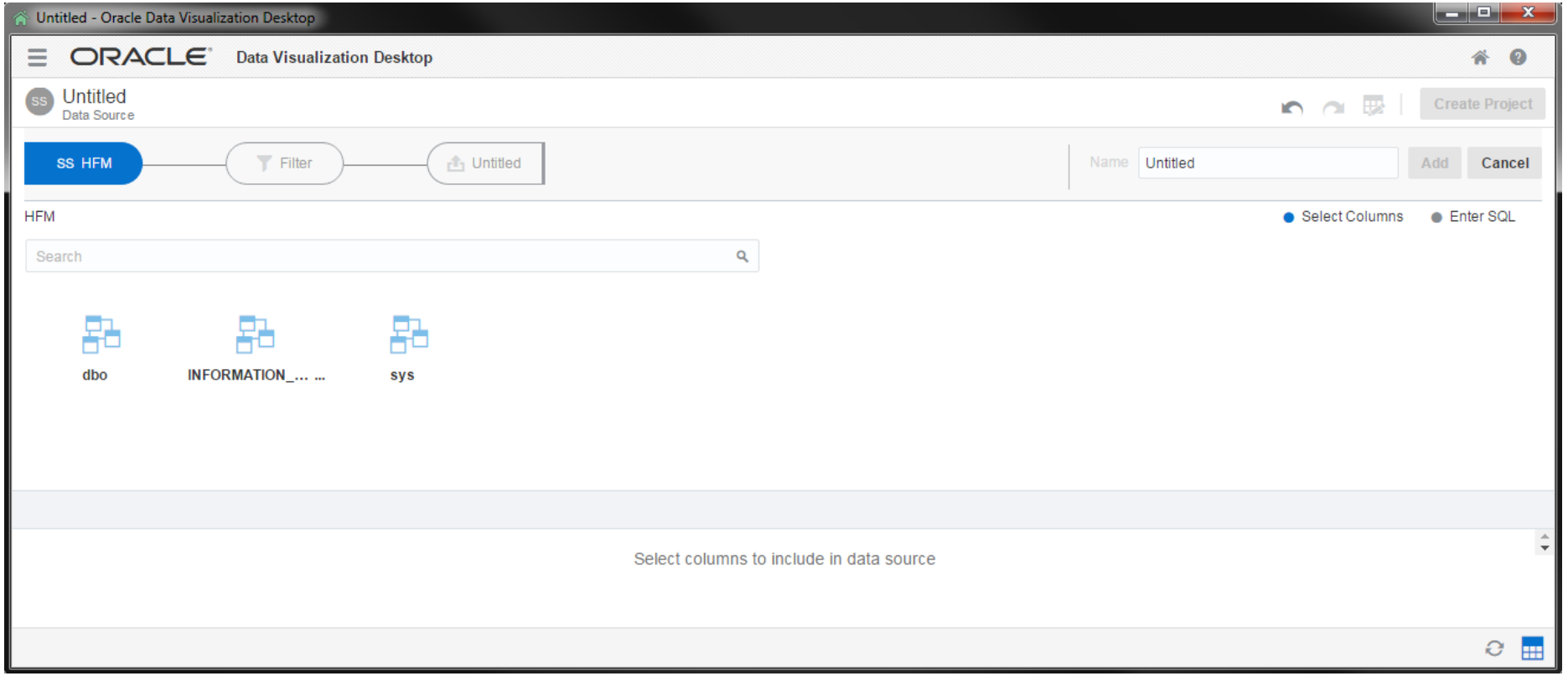
It can also be done from the connection list, by clicking on your database connection and selecting Create Data Source from the contextual menu, as well as by clicking on the database connection.



Part 2 – Connecting to a SQL Table

Step 2: Creating the data source

This will bring up the data source creation screen:

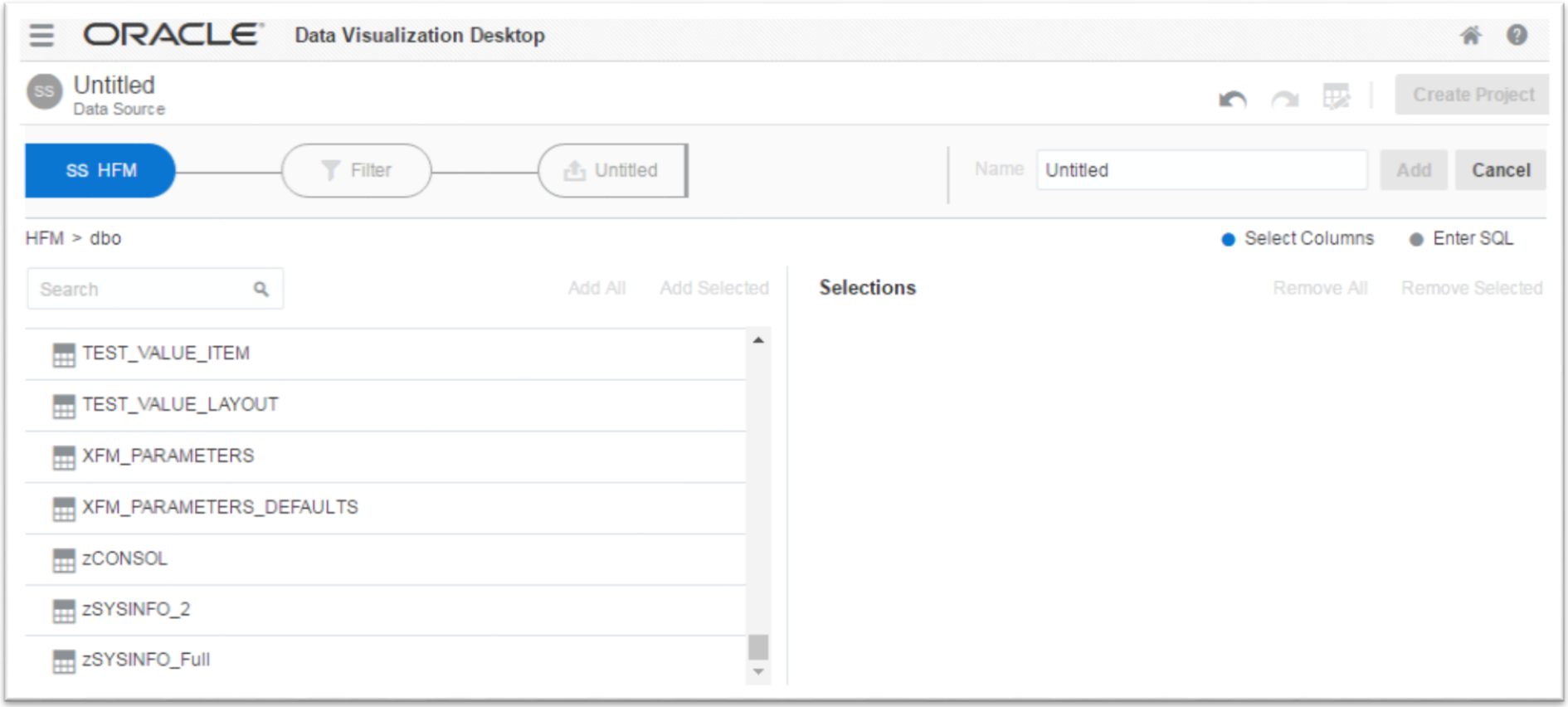


The data source creation screen shows the schemas and allows you to drill-down into tables.

Part 2 – Connecting to a SQL Table

Step 2: Creating the data source

By selecting the schema, you will access the list of tables, which will allow you to select the table you want to be a data source:

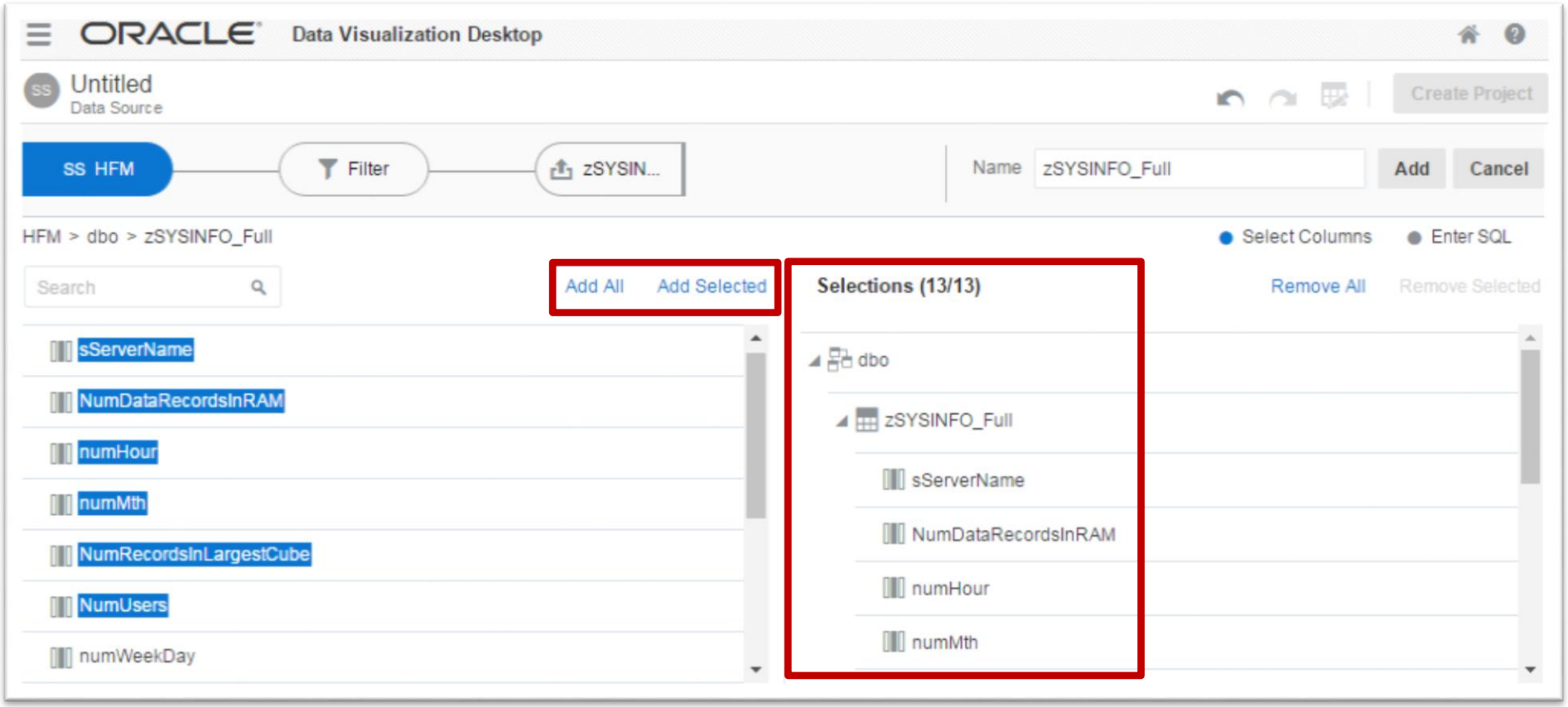


Part 2 – Connecting to a SQL Table

Step 2: Creating the data source

Once the table is selected, you will see the list of columns and be able to select the ones that are useful (in our case, all, since we use a specially designed table).

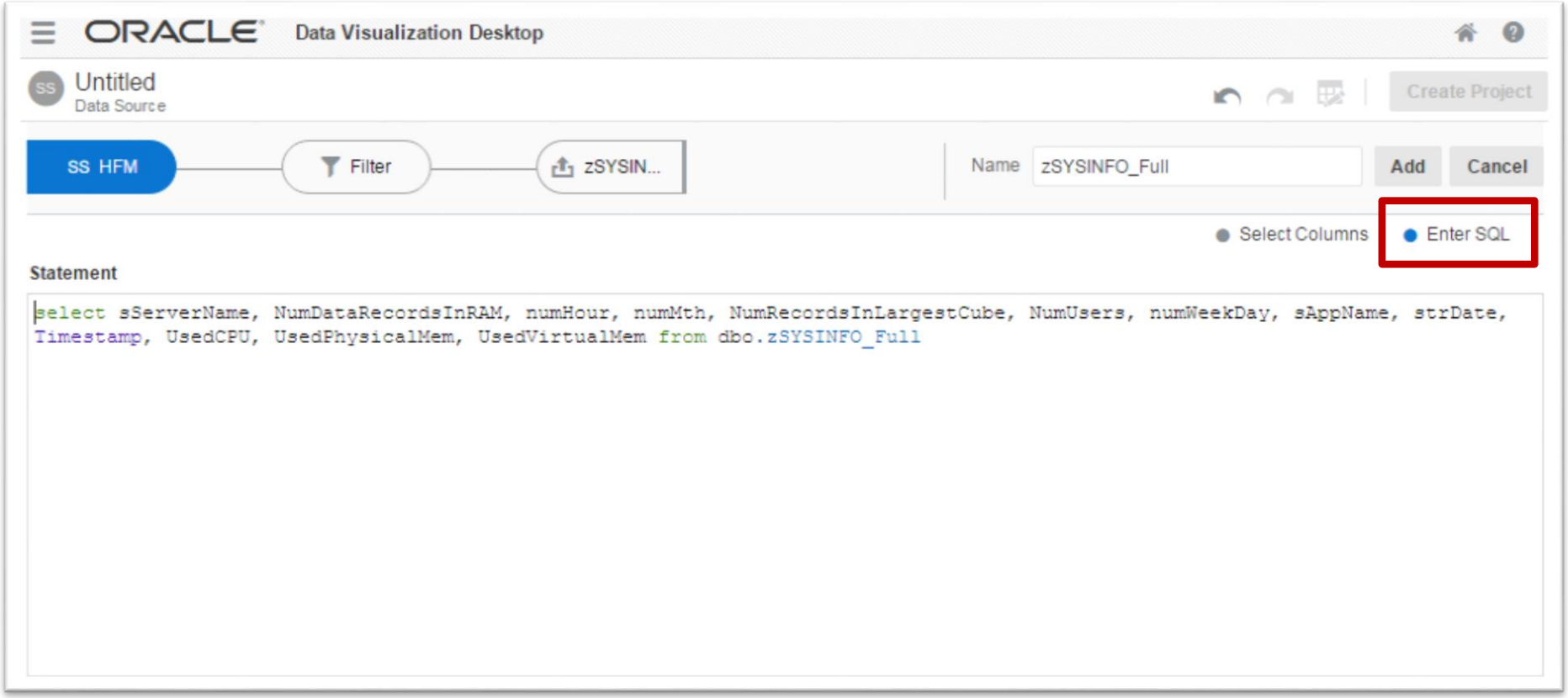
Selected columns will appear in the right pane



Part 2 – Connecting to a SQL Table

Step 2: Creating the data source

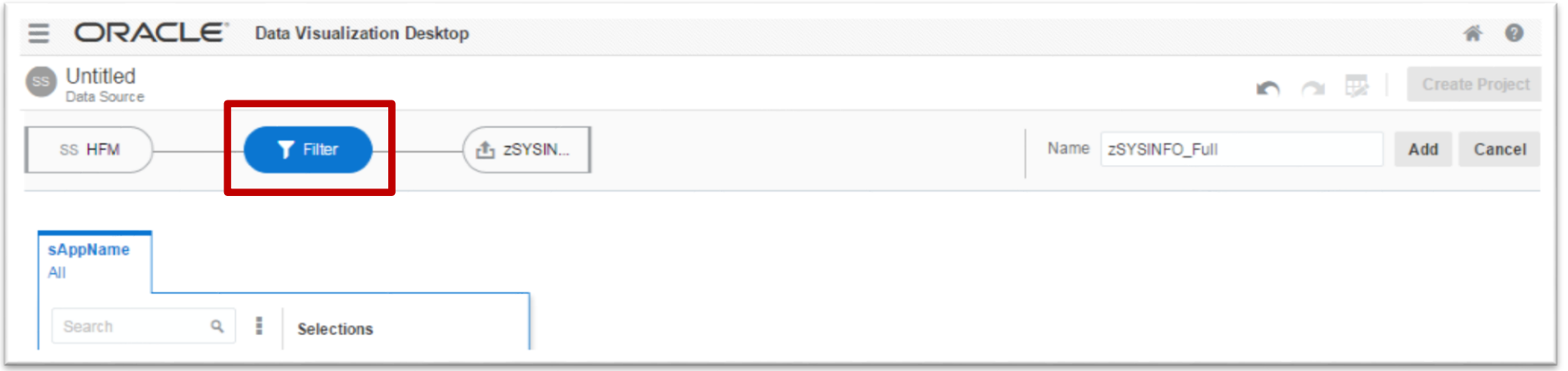
In the background, Oracle DV creates a SQL query. You can access it and update if you would like to do so:



Part 2 – Connecting to a SQL Table

Step 2: Creating the data source

You can add filters to the data from the filter screen:



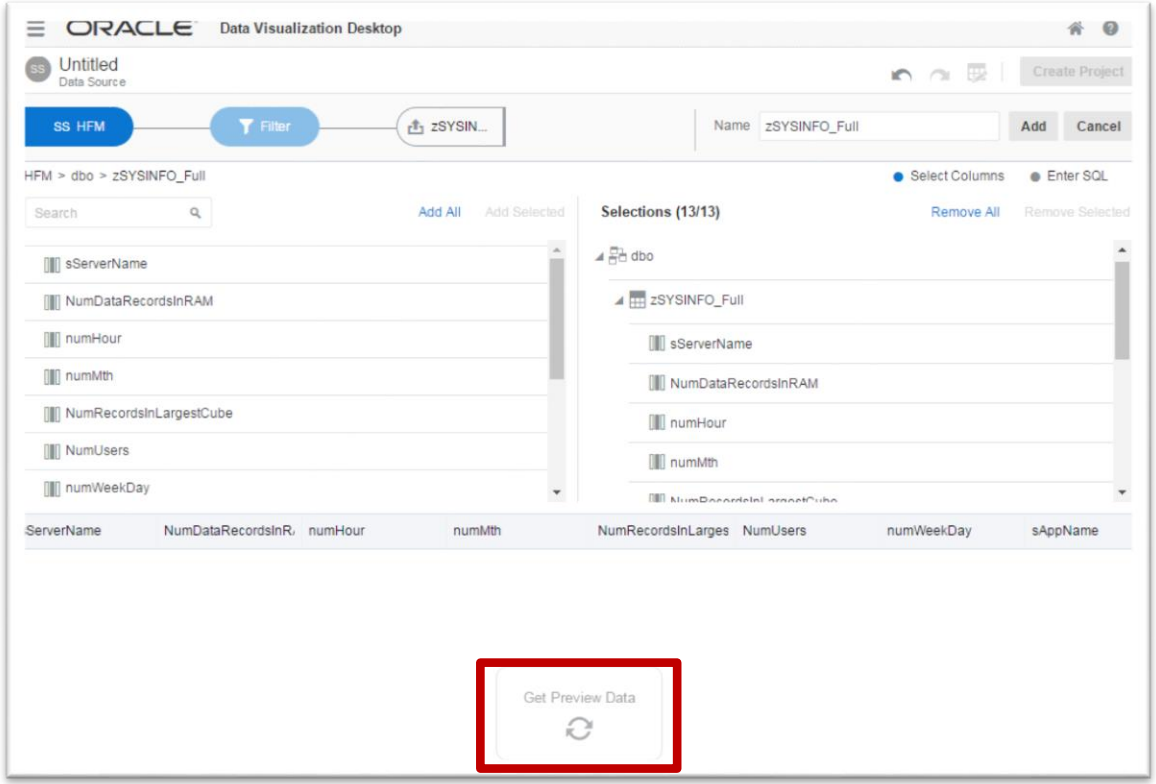
You can select any of the columns in scope, and set a filter through the Oracle DV interface.

Once again, this becomes part of the underlying SQL query and can be reviewed and edited there.

Part 2 – Connecting to a SQL Table

Step 2: Creating the data source

Once ready, click on Preview Data to verify the connection:

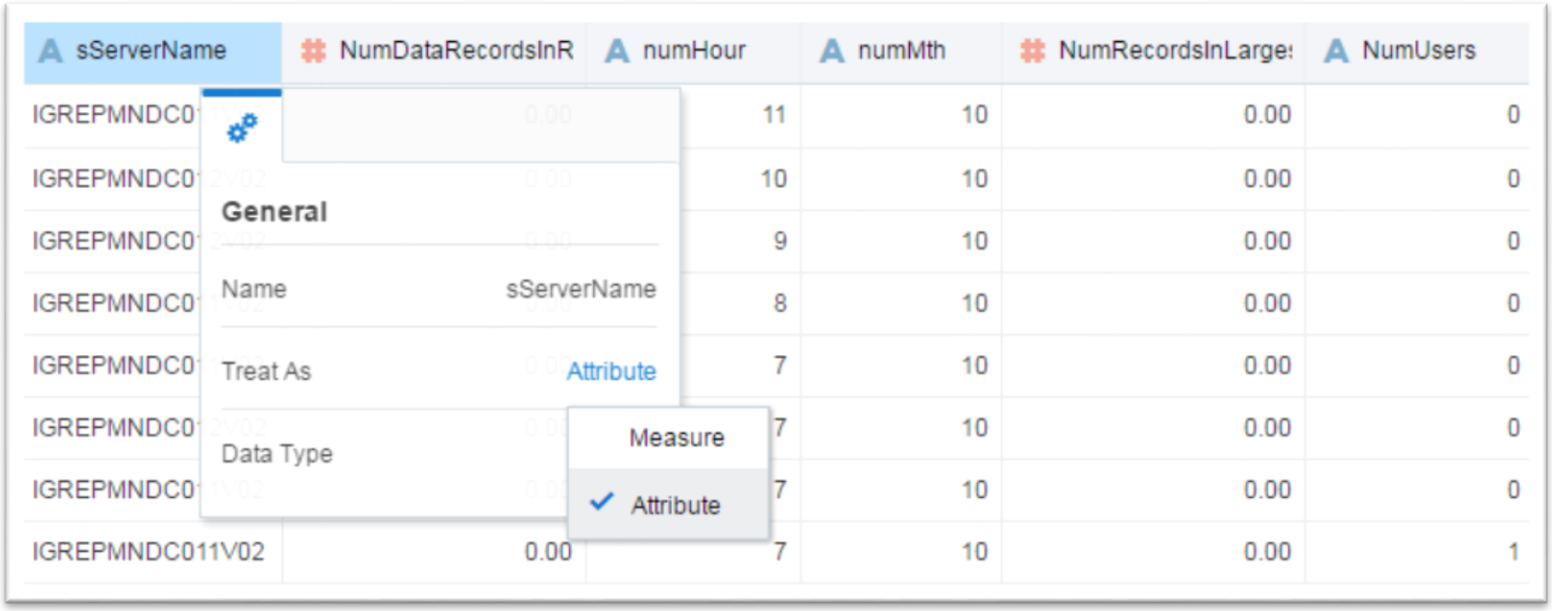


sServerName	NumDataRecordsInR	numHour	numMth	NumRecordsInLarge	NumUsers	numWeekDay	sAppName
IGREPMNDC011V02	0.00	11	10	0.00	0	7	HYPFMR12
IGREPMNDC012V02	0.00	10	10	0.00	0	7	HYPFMR12

Part 2 – Connecting to a SQL Table

Step 2: Creating the data source

The next step is to review the different columns settings:



Each column can be set to “Attribute” (a property that can be used for filtering, data grouping…) or a “Measure” (a value that will be graphed).

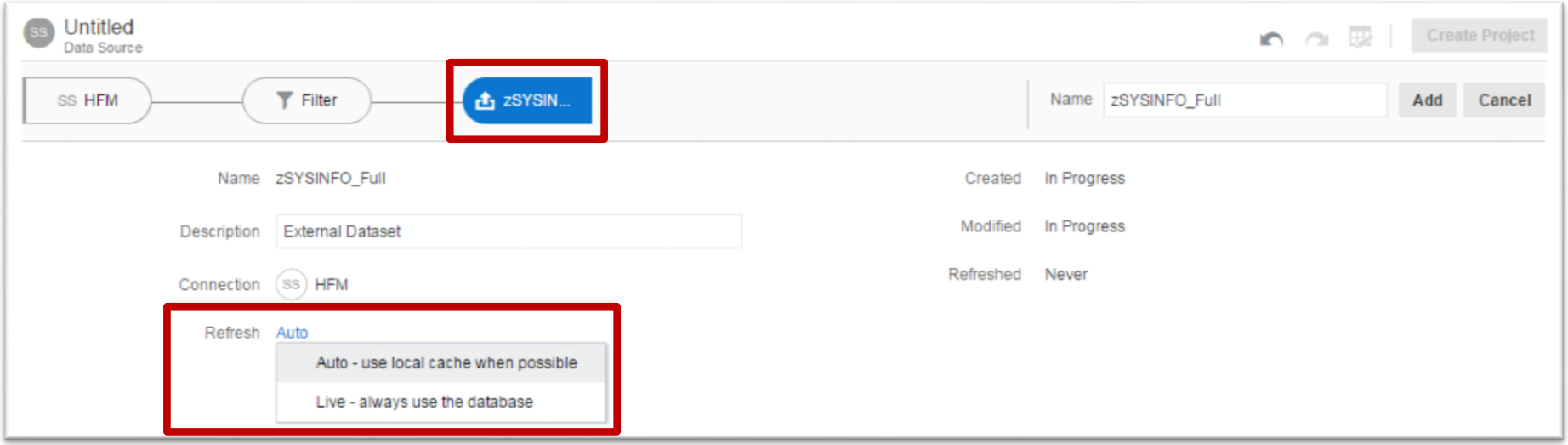
A measure can be assigned an Aggregation type (Sum, Average, Count…).

(This can be later updated).

Part 2 – Connecting to a SQL Table

Step 2: Creating the data source

Finally, set the data source parameters and save the data source:



In particular, you define on this screen whether the data will be refreshed Live, or "Auto":

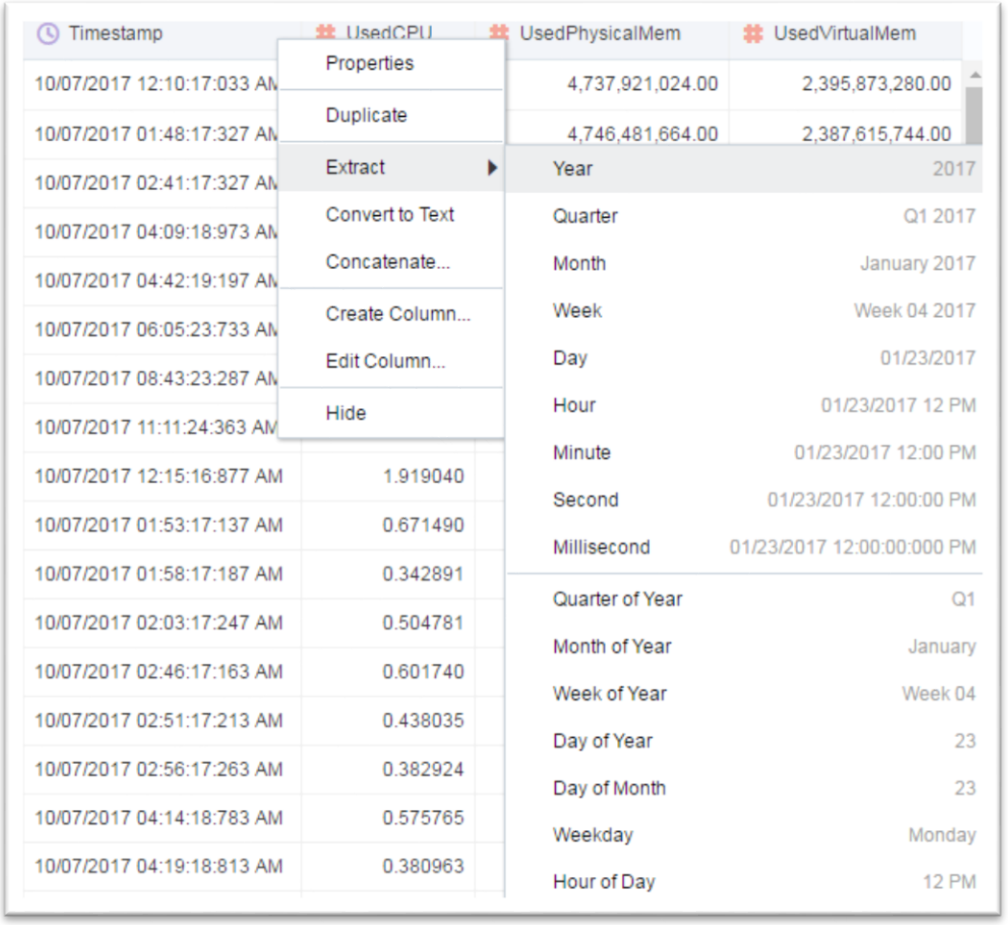
- **Live ensures that data is always pulled from the Database, but can impact performance**
- **Auto relies on a cache. This provides better performance but requires refreshes**

Part 2 – Connecting to a SQL Table

Step 2: Creating the data source

After saving the data source, you can add derived and calculated columns.

One set of derived columns that can be very useful are the derivations from a timestamp column:



There are more advanced functionalities for creating derived columns, which are not in scope of this presentation.

Hyperion Focus 17

Monitoring EPM systems performance with Oracle Data Visualization

Part 2: How do I connect to my data?

Connecting to a Spreadsheet file

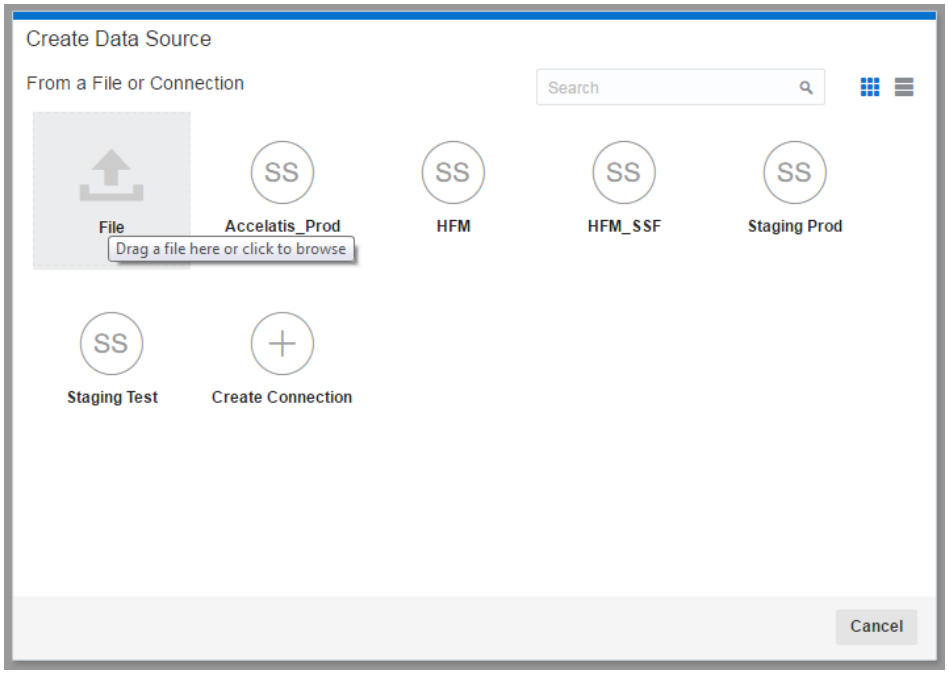
Part 2 – Connecting to a Spreadsheet file

This process is simpler than connecting to a database.

Essentially, we only perform step 2:

In Oracle DV:

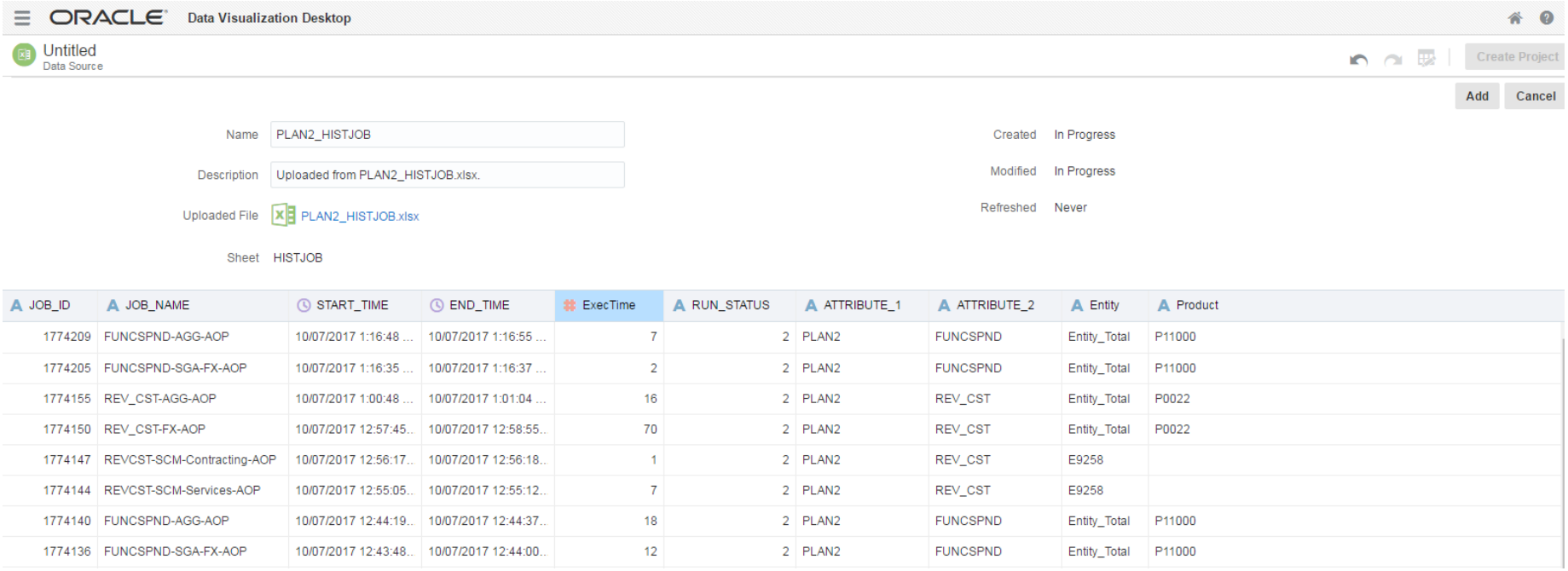
- **Navigate to “Data Sources”**
- **Click on Create Data Source**
- **Select “File”:**
  You can also drag an drop a file on this tile



Part 2 – Connecting to a Spreadsheet file

The spreadsheet is loaded and interpreted based on column titles and formats.

As with a database table, you have to review and where needed adjust the column types:



JOB_ID	JOB_NAME	START_TIME	END_TIME	ExecTime	RUN_STATUS	ATTRIBUTE_1	ATTRIBUTE_2	Entity	Product
1774209	FUNCSPND-AGG-AOP	10/07/2017 1:16:48 ...	10/07/2017 1:16:55 ...	7	2	PLAN2	FUNCSPND	Entity_Total	P11000
1774205	FUNCSPND-SGA-FX-AOP	10/07/2017 1:16:35 ...	10/07/2017 1:16:37 ...	2	2	PLAN2	FUNCSPND	Entity_Total	P11000
1774155	REV_CST-AGG-AOP	10/07/2017 1:00:48 ...	10/07/2017 1:01:04 ...	16	2	PLAN2	REV_CST	Entity_Total	P0022
1774150	REV_CST-FX-AOP	10/07/2017 12:57:45..	10/07/2017 12:58:55..	70	2	PLAN2	REV_CST	Entity_Total	P0022
1774147	REVCST-SCM-Contracting-AOP	10/07/2017 12:56:17..	10/07/2017 12:56:18..	1	2	PLAN2	REV_CST	E9258	
1774144	REVCST-SCM-Services-AOP	10/07/2017 12:55:05..	10/07/2017 12:55:12..	7	2	PLAN2	REV_CST	E9258	
1774140	FUNCSPND-AGG-AOP	10/07/2017 12:44:19..	10/07/2017 12:44:37..	18	2	PLAN2	FUNCSPND	Entity_Total	P11000
1774136	FUNCSPND-SGA-FX-AOP	10/07/2017 12:43:48..	10/07/2017 12:44:00..	12	2	PLAN2	FUNCSPND	Entity_Total	P11000

Hyperion Focus 17

Monitoring EPM systems performance with Oracle Data Visualization

Part 3: Let's visualize!

Hyperion Focus 17

Monitoring EPM systems performance with Oracle Data Visualization

Part 3: Let's visualize!

Creating a visualization

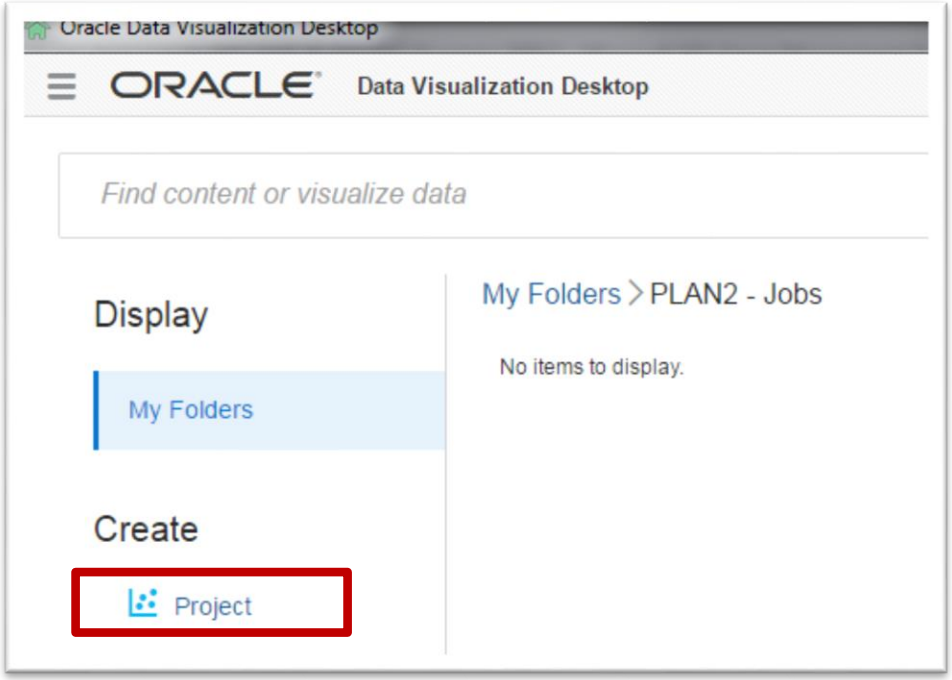
Part 3 – Creating a visualization

With data sources created, we can build visualizations.

We will first go through the steps of creating one, then discuss how the different visualizations can provide us insights into EPM systems performance.

To begin, from the Home page,

Click on Create Project



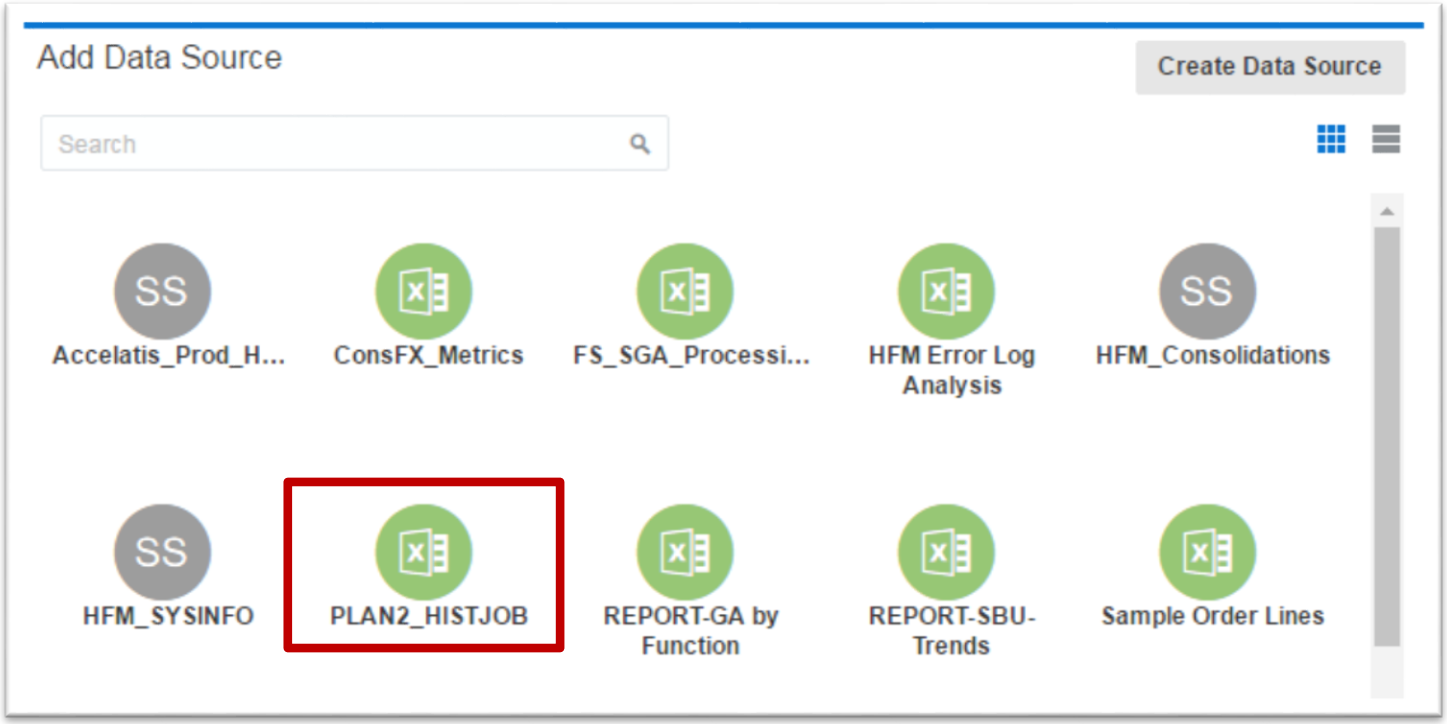
Part 3 – Creating a visualization

The first step is to add a data source.

DV will present the list of defined data sources, select the source or sources that you want to use for the project.

You can change sources later.

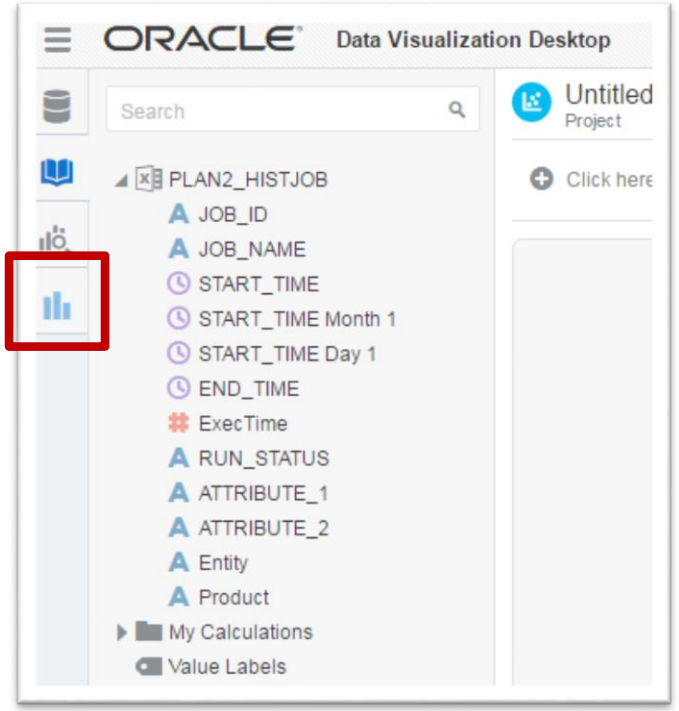
For the purpose of this walkthrough, I will use the example of the Planning Job History



Part 3 – Creating a visualization

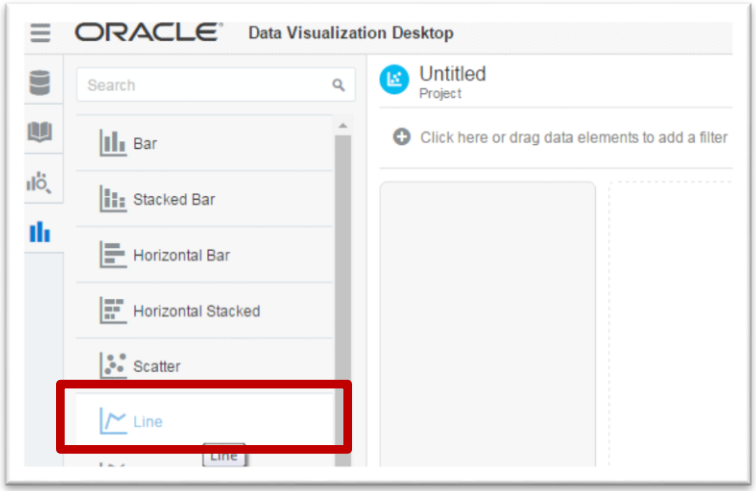
Once you have selected a data source, you are brought to the project screen.

Click on Visualizations to start creating one:



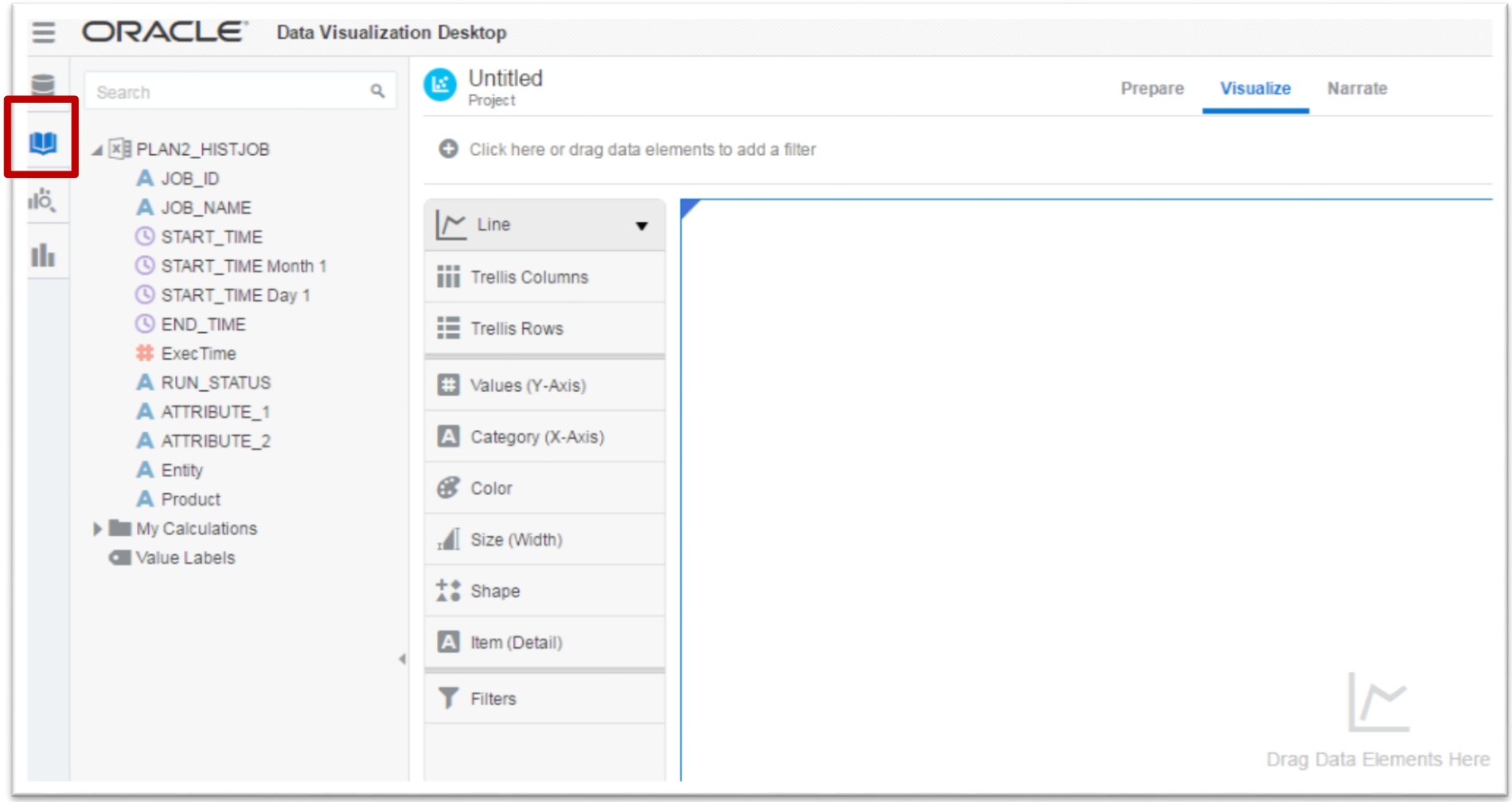
This will bring you to the list of visualizations and allow to select the one you want.

Let's select a Line graph:



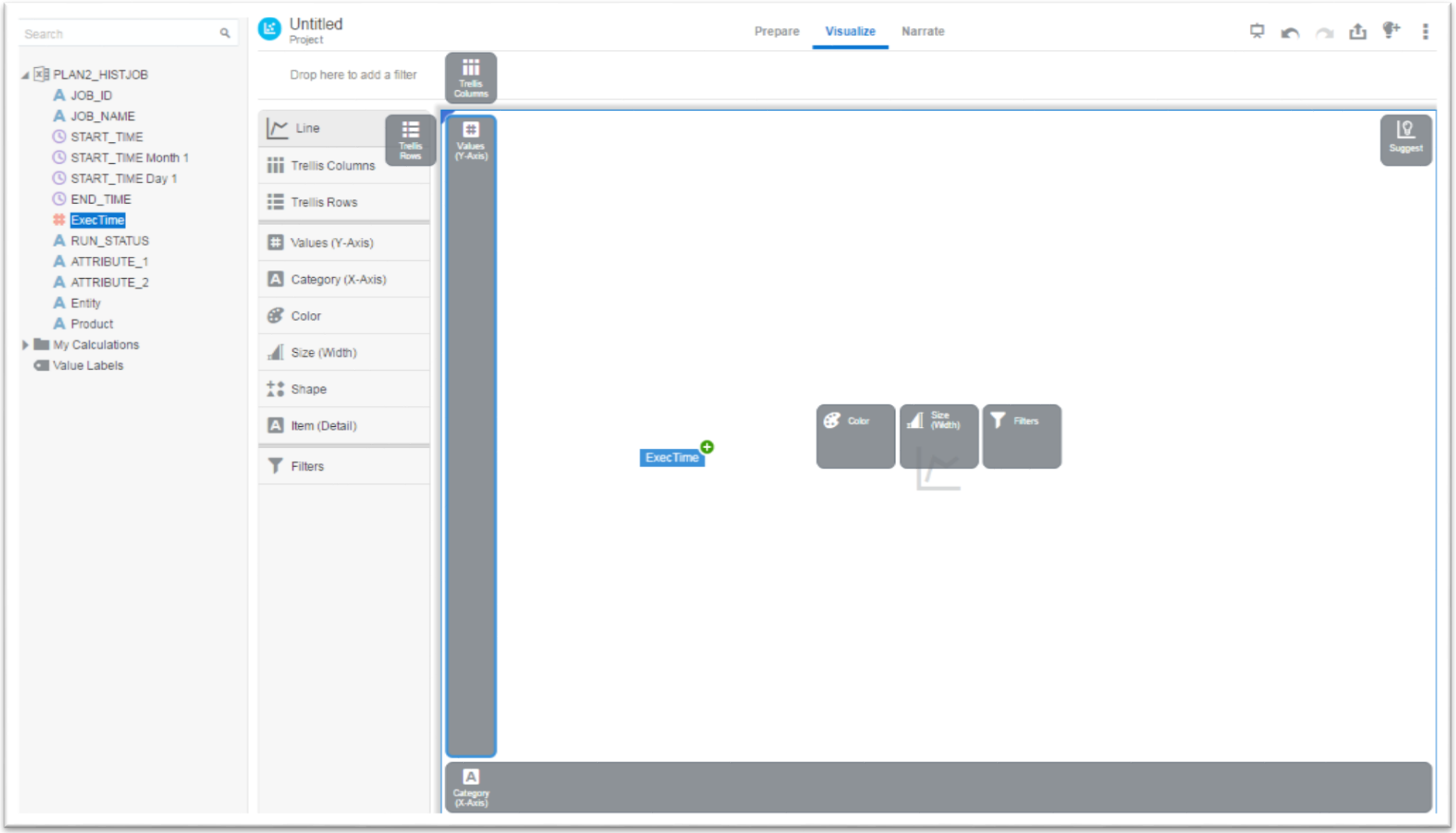
Part 3 – Creating a visualization

With the visualization created, select data elements, and set up your visualization by adding data elements by dragging and dropping into the main pane.



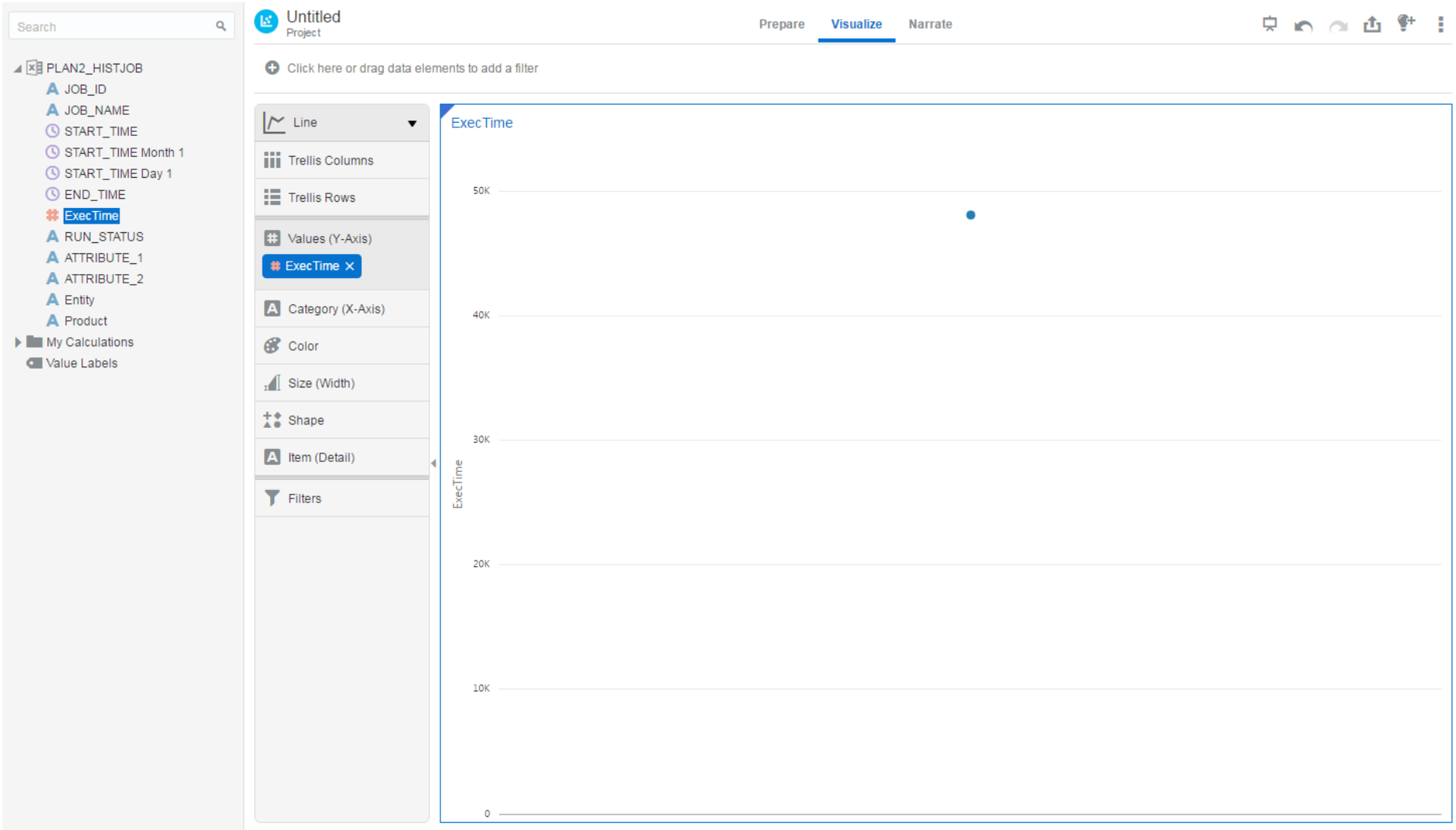
Part 3 – Creating a visualization

As you drag and drop, you will see different areas allowing you to define how the data element is used:



Part 3 – Creating a visualization

Once you have added a data element, it will appear in the left column. You can drag and drop or remove it from there:

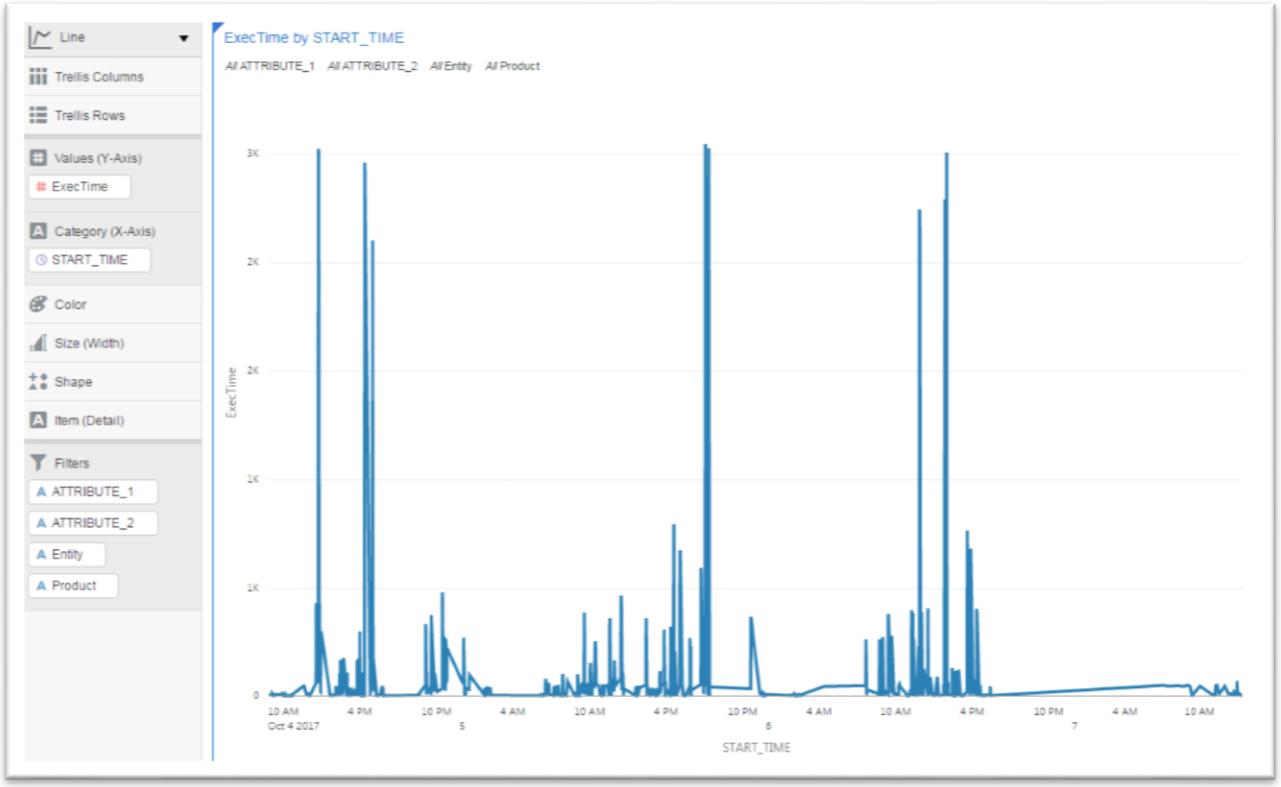


Part 3 – Creating a visualization

We will use the following for this Line graph:

- **Execution Time (in seconds. Integer) as Values (Y-Axis)**
- **START_TIME as Category (X-Axis)**
- **JOB_NAME, ATTRIBUTE_1, ATTRIBUTE_2, Entity, Product as Filters**

With my data set, the output is the following:



Part 3 – Creating a visualization

While this gives a line graph, let's consider what it shows:

All business rules execution times by date, for all our applications, and all parameters.

We can change a bit the presentation by putting JOB_NAME as Color:

Part 3 – Creating a visualization

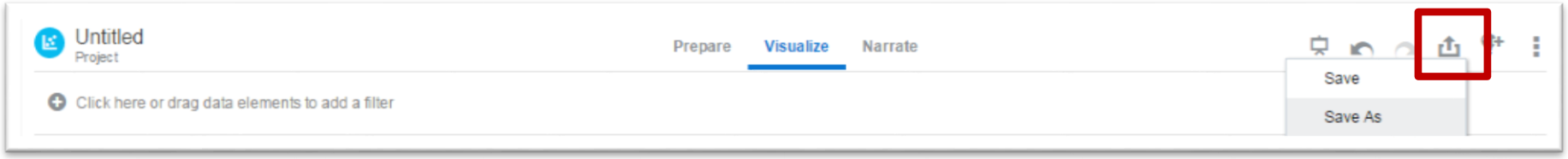
We are now getting into the analytics: this new view allows to identify which business rules have the longest execution time.

We can use filters to focus on specific datasets, identify areas of concerns, or patterns.

It is useful to first spend some time exploring your dataset to find the most useful selections, filters and visualization.

As you identify them, create multiple Canvases set to the views that are most useful.

Remember to save your project before closing, or you will lose your work



Hyperion Focus 17

Monitoring EPM systems performance with Oracle Data Visualization

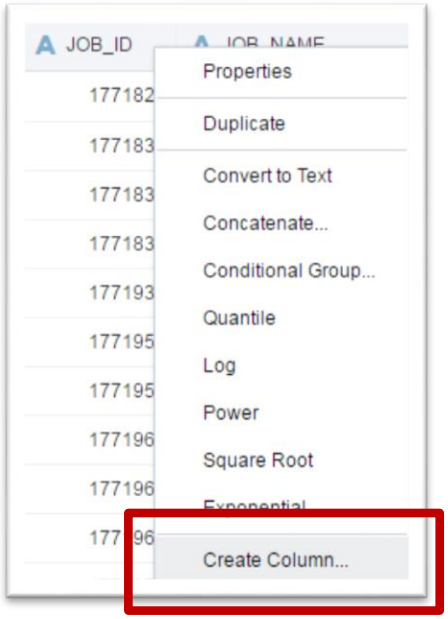
Part 3: Let's visualize!

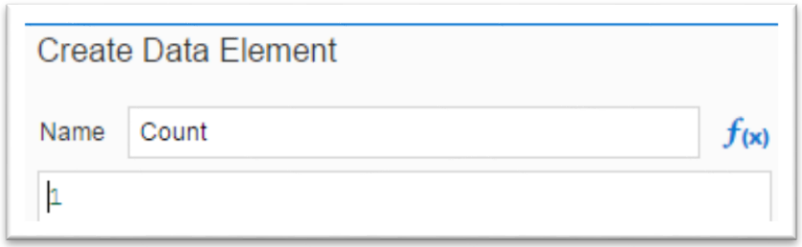
Visualizing data to understand EPM performace

Part 3 – Visualizing data to understand EPM performance

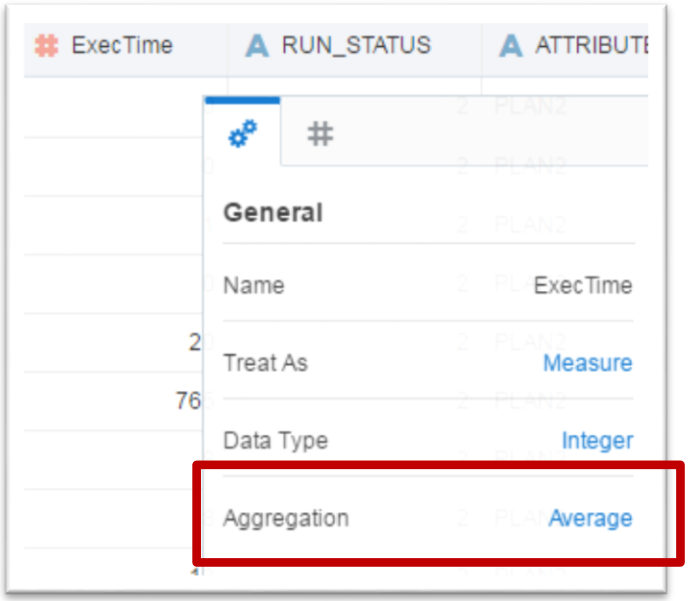
Count of rules execution and Average time by Application and Cube.

In order to design this, we will add a calculated column to the data source, named Count, containing a fixed value of 1. From the data source, click on the contextual menu of an existing column, and select Create column:



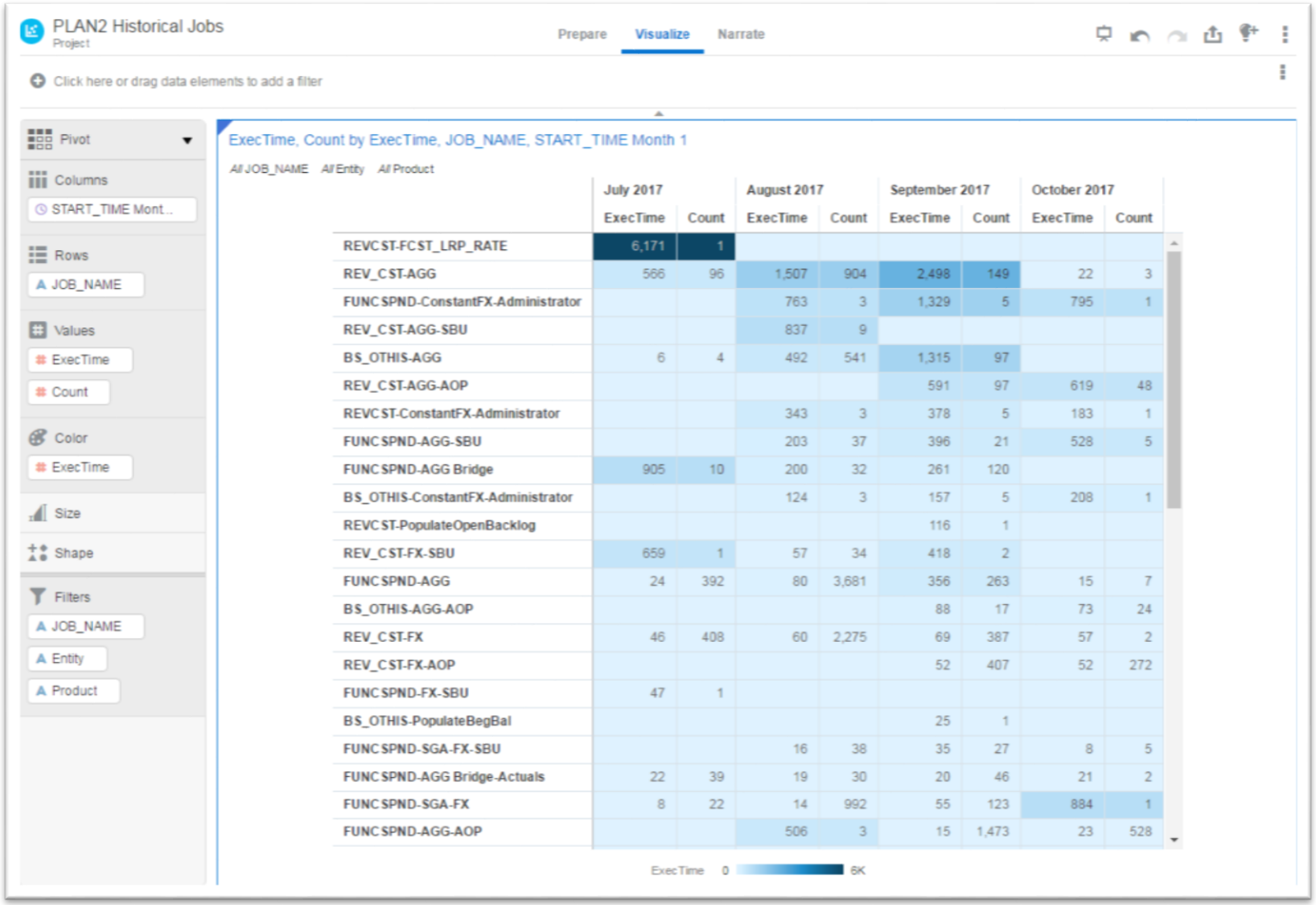


Also verify that the measure Execution time has its aggregation property set to Average:



Part 3 – Visualizing data to understand EPM performance

With the count created, we will create a Pivot time visualization with the following settings:

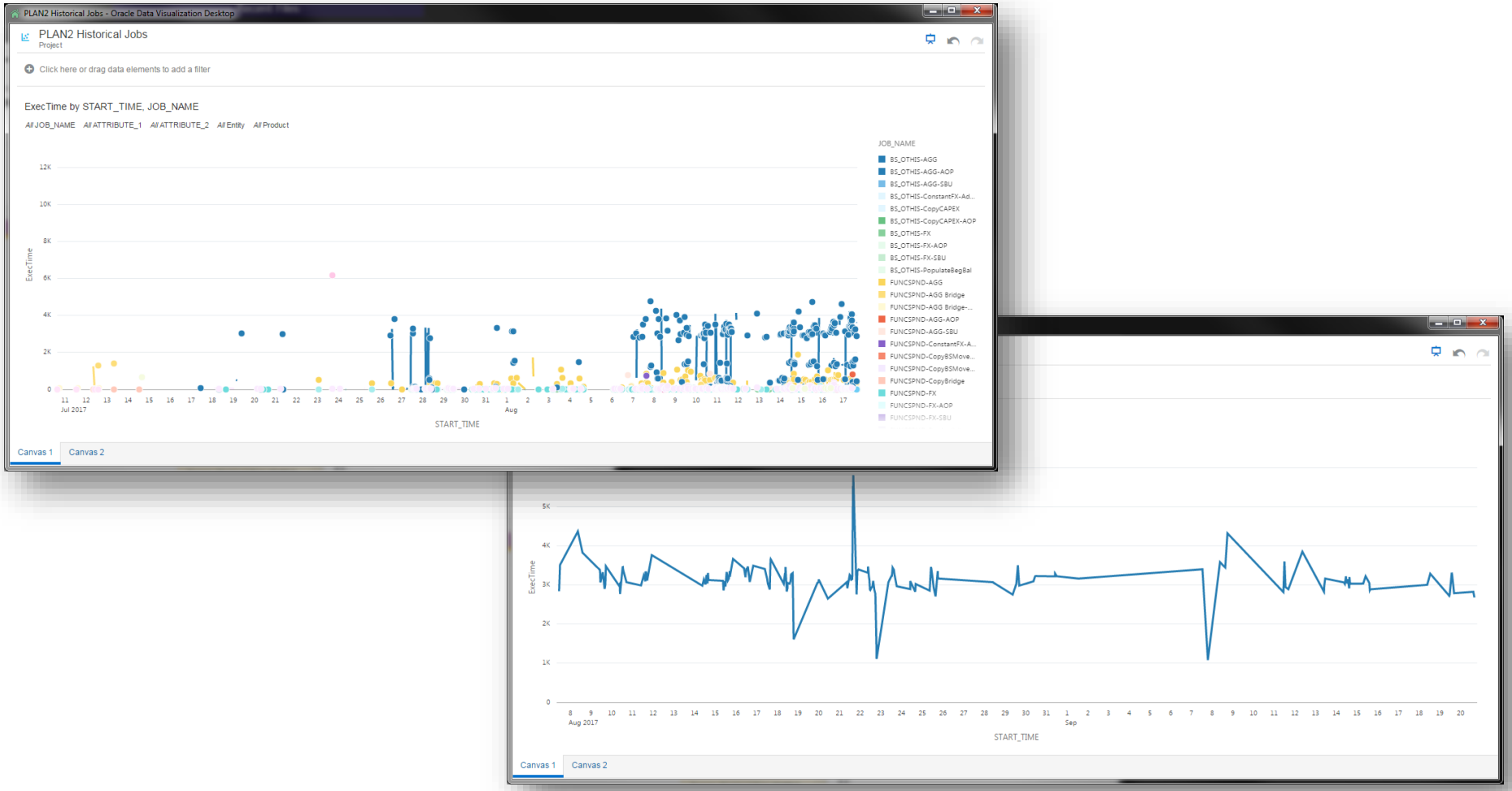


	July 2017		August 2017		September 2017		October 2017	
	ExecTime	Count	ExecTime	Count	ExecTime	Count	ExecTime	Count
REVCST-FCST_LRP_RATE	6,171	1						
REV_CST-AGG	566	96	1,507	904	2,498	149	22	3
FUNCSPND-ConstantFX-Administrator			763	3	1,329	5	795	1
REV_CST-AGG-SBU			837	9				
BS_OTHIS-AGG	6	4	492	541	1,315	97		
REV_CST-AGG-AOP					591	97	619	48
REVCST-ConstantFX-Administrator			343	3	378	5	183	1
FUNCSPND-AGG-SBU			203	37	396	21	528	5
FUNCSPND-AGG Bridge	905	10	200	32	261	120		
BS_OTHIS-ConstantFX-Administrator			124	3	157	5	208	1
REVCST-PopulateOpenBacklog					116	1		
REV_CST-FX-SBU	659	1	57	34	418	2		
FUNCSPND-AGG	24	392	80	3,681	356	263	15	7
BS_OTHIS-AGG-AOP					88	17	73	24
REV_CST-FX	46	408	60	2,275	69	387	57	2
REV_CST-FX-AOP					52	407	52	272
FUNCSPND-FX-SBU	47	1						
BS_OTHIS-PopulateBegBal					25	1		
FUNCSPND-SGA-FX-SBU			16	38	35	27	8	5
FUNCSPND-AGG Bridge-Actuals	22	39	19	30	20	46	21	2
FUNCSPND-SGA-FX	8	22	14	992	55	123	884	1
FUNCSPND-AGG-AOP			506	3	15	1,473	23	528

Part 3 – Visualizing data to understand EPM performance

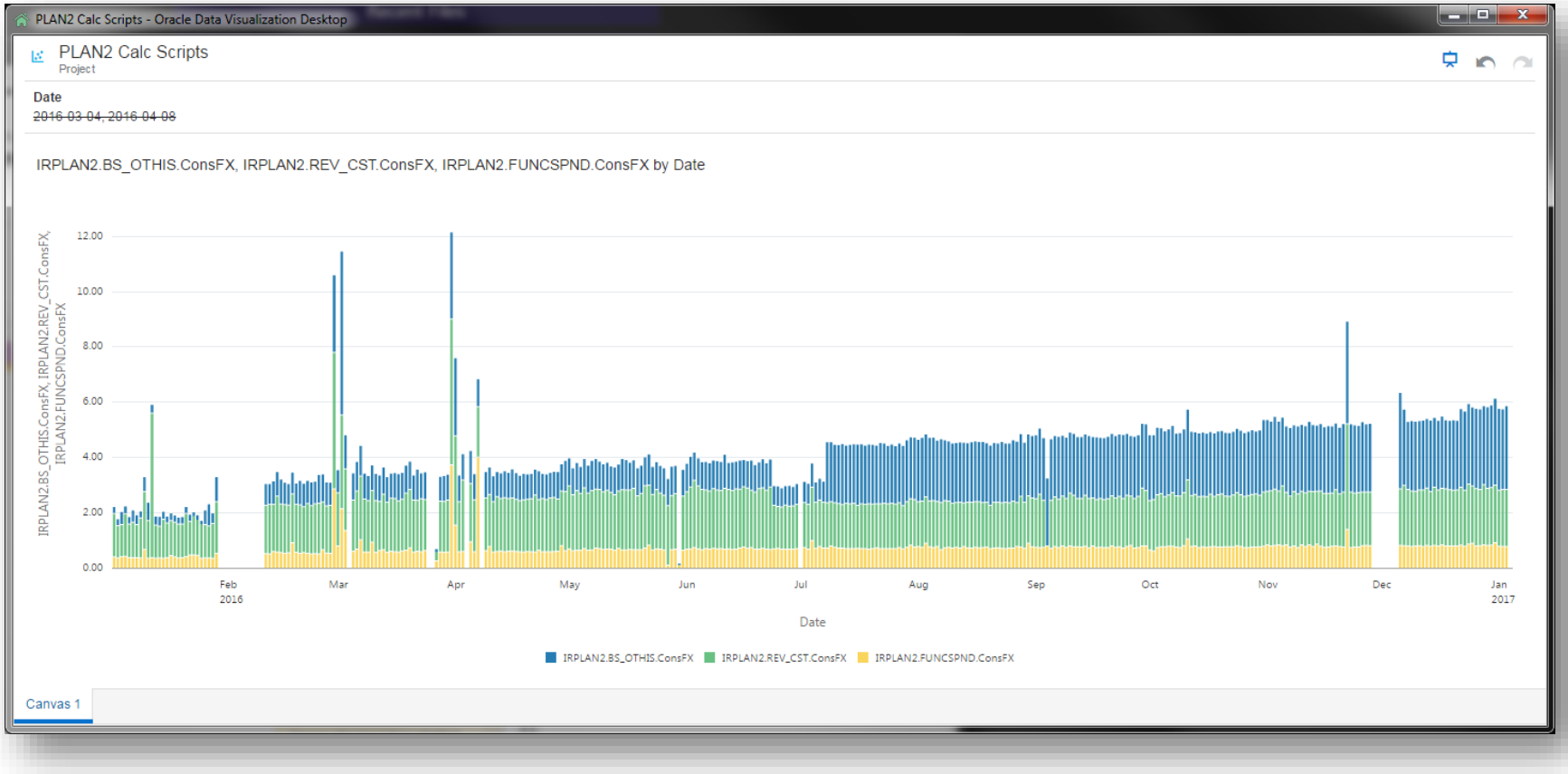
Understanding which rules are executed most often as well as which rules have the highest average time can provide insights.

This is particularly useful as an input to further analyze execution times over time, i.e. as a filter for the chronological view:



Part 3 – Visualizing data to understand EPM performance

Execution time of Calculation scripts over time:



This type of visualization allows to identify oddities, as well as the general evolution of the process.

This can allow to take actions before the process evolution causes adverse effects.

Part 3 – Visualizing data to understand EPM performance

Analyzing HFM System Information:

Looking at the correlation, or lack of, between different system metrics can provide insights. For example, in the following comparison, we can see memory usage grow during month end close and correlate to number of records in memory.

In this case, we can see how the limit of physical memory on the server is reached during peak phases and the process relies on Virtual memory.

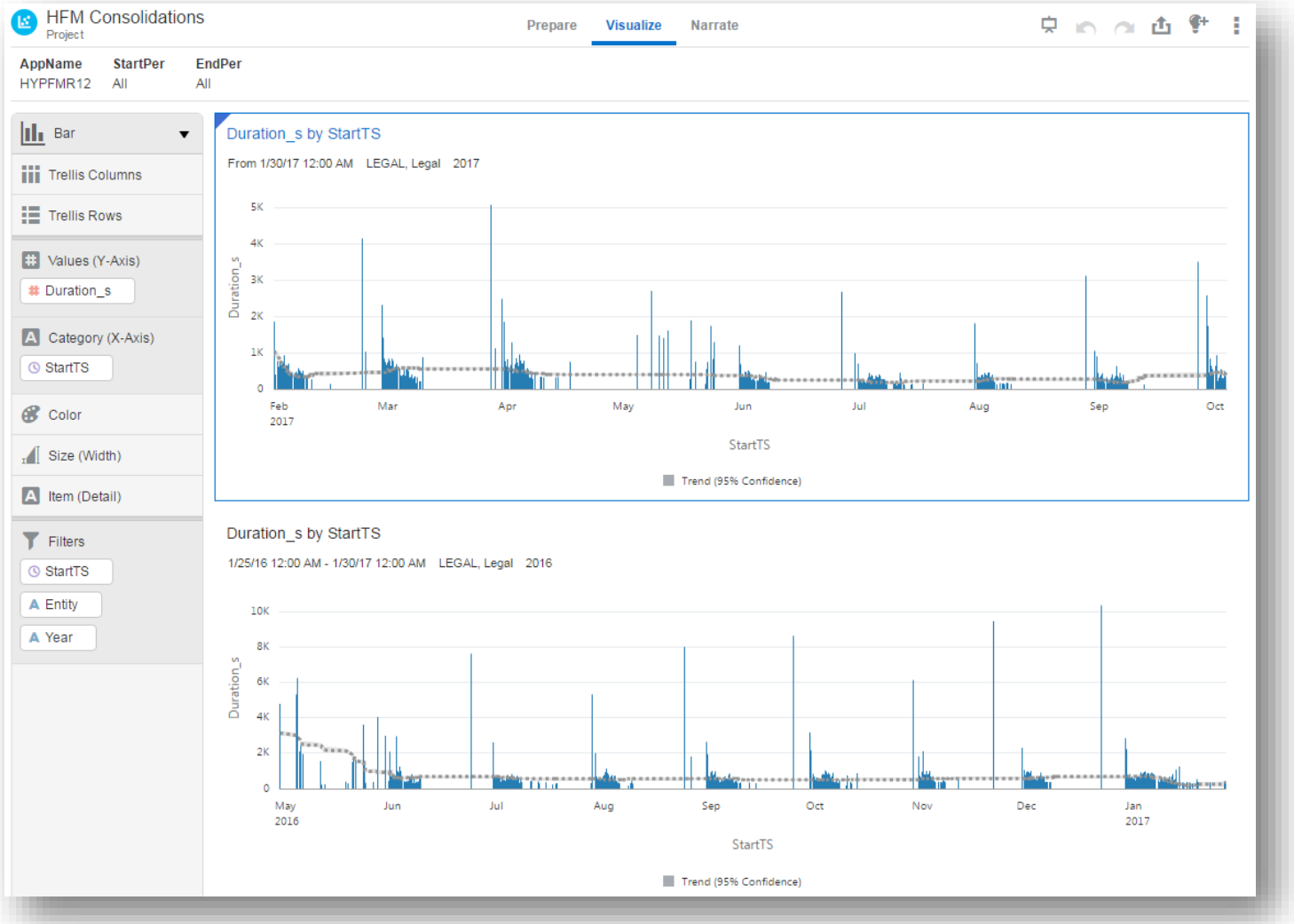
Part 3 – Visualizing data to understand EPM performance

The definitions of the graphs are as follows:

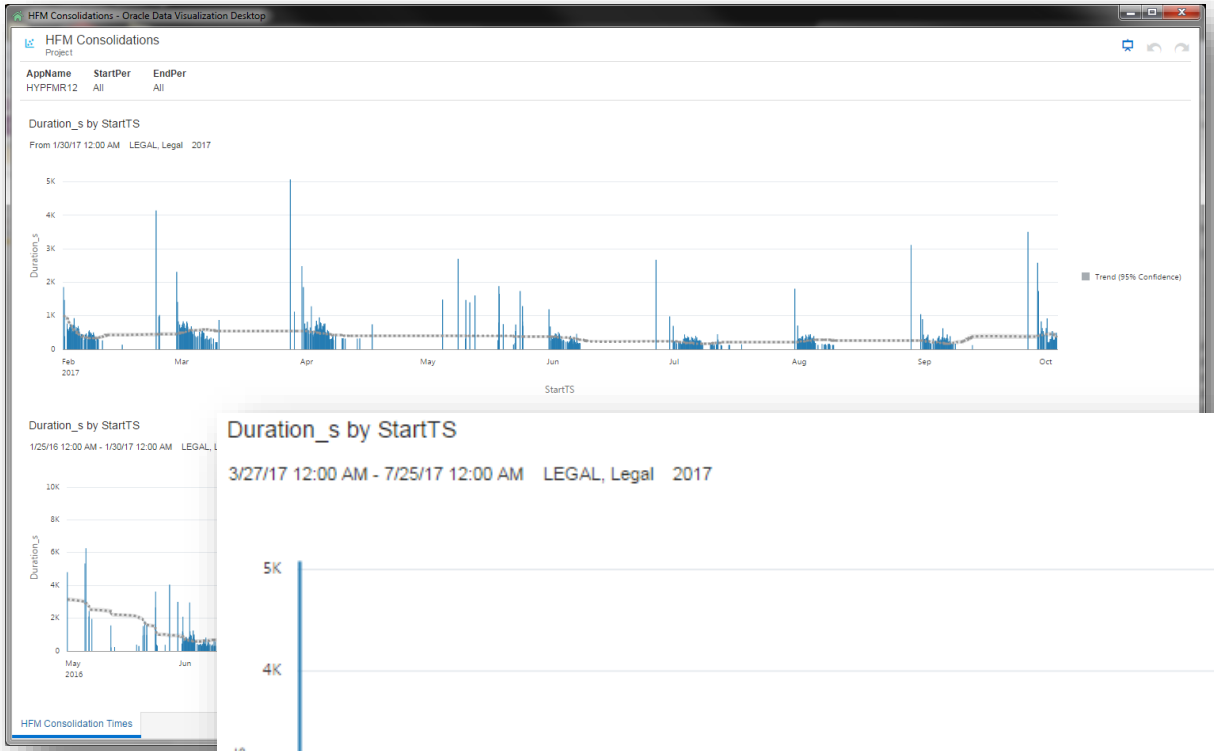
Part 3 – Visualizing data to understand EPM performance

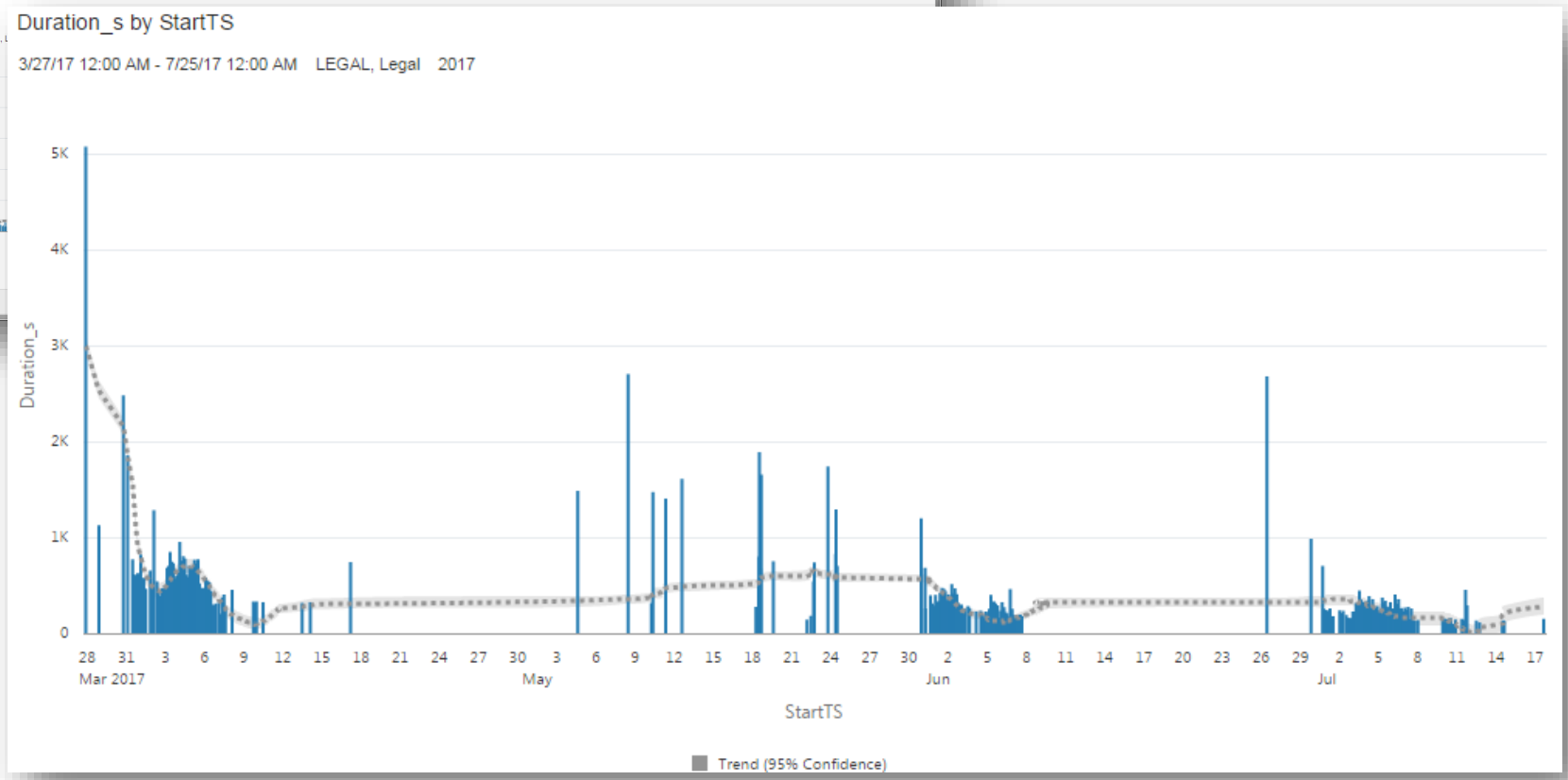
HFM Consolidation times

In this example, the two visualizations are the same, but for different years, allowing a year over year comparison.



Part 3 – Visualizing data to understand EPM performance





Ending Comments

There are many possible visualizations, and which one to use depends on your data and the analysis you are performing.

In real life cases, the order of steps may be the following:

- **Identify an area of interest or concern**
- **Identify datasets that may be relevant to this area**
- **Obtain data and use visualizations to understand patterns and trends, gaining insights that can lead to appropriate actions or provide evidence in support of requests for resources**

Hyperion Focus 17

Thank you