Article

Lydia Huang · Mar 14, 2016  6m read

# Get your MDX query plan, MDX Query Trace and more from the DeepSee Shell

## The DeepSee Shell Best Practices Series - Get your MDX query plan, MDX Query Trace and more from the DeepSee Shell

To process a query, DeepSee performs the following steps:

1. **Preparation**, which occurs in process (that is, this step is not launched as a background process). In this phase:

- DeepSee parses the query and converts it to an object representation, the parse tree. In the parse tree, each axis of the query is represented separately. One axis represents the overall filtering of the query.
- DeepSee converts the parse tree to a normalized version of the query text. In this normalized version, for example, all %FILTER clauses have been combined into a single, equivalent WHERE clause.
- DeepSee generates a hash that is based on the normalized query text. DeepSee uses this hash value as the query key. The query key enables DeepSee to look up results for this query in the globals discussed in this appendix.
- If DeepSee finds that it is possible to reuse previous results for this query (from ^DeepSee.Cache.Results), DeepSee does so and skips the following steps.

2. **Execute axes**, which also occurs in process. In this phase:

- DeepSee executes any subqueries.
- DeepSee examines the slicer axis (the WHERE clause), merges in any relevant filtering (such as from a subject area filter), and updates ^DeepSee.Cache.Axis with information about this axis.
- DeepSee examines each of the remaining axes and updates ^DeepSee.Cache.Axis.

3. **Execute cells**, which occurs in the background (in multiple parallel processes). In this phase, DeepSee obtains intermediate values for each cell of the results, separately for each bucket, as follows:

- First DeepSee checks to see if ^DeepSee.Cache.Cell contains a value for the cell for the given bucket.

 If so, DeepSee uses that value.

- Otherwise, DeepSee uses the applicable nodes of ^DeepSee.Index to obtain the bitmap indices that it needs. DeepSee combines these bitmap indices and then uses the result to find the applicable records in the source table.

 If the cache uses buckets, DeepSee adds nodes to ^DeepSee.Cache.Cell for use by later queries.

4. **Consolidation**, which occurs in process. In this phase:

- For each slicer axis, DeepSee examines each result cell for that axis.

For each result cell, DeepSee finds all the nodes in ^DeepSee.Cache.Cell that contain values for this cell.

It then combines those values.

- For each result cell, DeepSee then combines the results across the slicer axes and obtains a single value.

```
>>plan

-------------- Query Plan --------------------

**WITH MEMBER [MEASURES].[MEDIAN TEST SCORE] AS '%KPI("%DeepSee.Median","MEDIAN",1,"%
measure","Test Score","%CONTEXT")',FORMAT_STRING='#.##' SELECT {[MEASURES].[%COUNT],[
MEASURES].[ALLERGY COUNT],%KPI("%DeepSee.Median","MEDIAN",1,"%measure","Test Score","
%CONTEXT")} ON 0,NON EMPTY HEAD(NONEMPTYCROSSJOIN([AGED].[H1].[AGE GROUP].MEMBERS,[AG
ED].[H1].[AGE BUCKET].MEMBERS),2000) ON 1 FROM [PATIENTS] WHERE [GEND].[H1].[GENDER].
&[Female]**

**DIMENSION QUERY (%FindMemberByKey): SELECT %ID,DxGender FROM DeepSee_Model_Patients
Cube.DxGender WHERE DxGender =?**

**DIMENSION QUERY (%GetMembers): Apply memberList**

**DIMENSION QUERY (%GetMembers): SELECT %ID,DxAgeBucket MKEY,DxAgeBucket FROM DeepSee
_Model_PatientsCube.DxAgeBucket WHERE (DxAgeGroup = '2') ORDER BY DxAgeBucket**

**DIMENSION QUERY (%GetMembers): SELECT %ID,DxAgeBucket MKEY,DxAgeBucket FROM DeepSee
_Model_PatientsCube.DxAgeBucket WHERE (DxAgeGroup = '3') ORDER BY DxAgeBucket**

**DIMENSION QUERY (%GetMembers): SELECT %ID,DxAgeBucket MKEY,DxAgeBucket FROM DeepSee
_Model_PatientsCube.DxAgeBucket WHERE (DxAgeGroup = '1') ORDER BY DxAgeBucket**

**EXECUTE PARALLEL: 1x1 task(s) **

**CONSOLIDATE**

**DIMENSION QUERY (%FindMemberByKey): SELECT %ID,DxGender FROM DeepSee_Model_Patients
Cube.DxGender WHERE DxGender =?**

**DIMENSION QUERY (%GetMembers): Apply memberList**

**DIMENSION QUERY (%GetMembers): SELECT %ID,DxAgeBucket MKEY,DxAgeBucket FROM DeepSee
_Model_PatientsCube.DxAgeBucket WHERE (DxAgeGroup = '2') ORDER BY DxAgeBucket**

**DIMENSION QUERY (%GetMembers): SELECT %ID,DxAgeBucket MKEY,DxAgeBucket FROM DeepSee
_Model_PatientsCube.DxAgeBucket WHERE (DxAgeGroup = '3') ORDER BY DxAgeBucket**

**DIMENSION QUERY (%GetMembers): SELECT %ID,DxAgeBucket MKEY,DxAgeBucket FROM DeepSee
_Model_PatientsCube.DxAgeBucket WHERE (DxAgeGroup = '1') ORDER BY DxAgeBucket**

-------------- End of Plan ----------------

>>trace

Trace is: off

>>trace on
```

```
Trace is: on

>>g

SELECT {[Measures].[%COUNT],[Measures].[Allergy Count],[MEASURES].[MEDIAN TEST SCORE]
} ON 0,NON EMPTY HEAD(NONEMPTYCROSSJOIN([AgeD].[H1].[Age Group].Members,[AgeD].[H1].[
Age Bucket].Members),2000) ON 1 FROM [PATIENTS] %FILTER [GenD].[H1].[Gender].&[Female
]

--> %Prepare: PATIENTS: 2016-02-11 15:02:08

--> %ExecuteParameters: PATIENTS

--> %InitializeResultsCache: PATIENTS

--> %InitializeResultsCache: PATIENTS: QUERY IS IN CACHE: en142672151

--> %PreProcessQuery: PATIENTS

--> %ExecuteAsynch: PATIENTS

--> %ExecuteAxes: PATIENTS: Status: 10

--> %InitializeAxisCache: PATIENTS: slicer: MEMBER [MEASURES].[MEDIAN TEST SCORE] AS
'%KPI("%DeepSee.Median","MEDIAN",1,"%measure","Test Score","%CONTEXT")',FORMAT_STRING
='#.##':[GEND].[H1].[GENDER].&[Female]:[GEND].[H1].[GENDER].&[Female]

--> %InitializeAxisCache: PATIENTS: REUSING AXIS: slicer: MEMBER [MEASURES].[MEDIAN T
EST SCORE] AS '%KPI("%DeepSee.Median","MEDIAN",1,"%measure","Test Score","%CONTEXT")'
,FORMAT_STRING='#.##':[GEND].[H1].[GENDER].&[Female]:[GEND].[H1].[GENDER].&[Female]

--> %InitializeAxisCache: PATIENTS: axis: MEMBER [MEASURES].[MEDIAN TEST SCORE] AS '%
KPI("%DeepSee.Median","MEDIAN",1,"%measure","Test Score","%CONTEXT")',FORMAT_STRING='
#.##':[GEND].[H1].[GENDER].&[Female]:{[MEASURES].[%COUNT],[MEASURES].[ALLERGY COUNT],
%KPI("%DeepSee.Median","MEDIAN",1,"%measure","Test Score","%CONTEXT")}

--> %InitializeAxisCache: PATIENTS: REUSING AXIS: axis: MEMBER [MEASURES].[MEDIAN TES
T SCORE] AS '%KPI("%DeepSee.Median","MEDIAN",1,"%measure","Test Score","%CONTEXT")',F
ORMAT_STRING='#.##':[GEND].[H1].[GENDER].&[Female]:{[MEASURES].[%COUNT],[MEASURES].[A
LLERGY COUNT],%KPI("%DeepSee.Median","MEDIAN",1,"%measure","Test Score","%CONTEXT")}

--> %InitializeAxisCache: PATIENTS: axis: MEMBER [MEASURES].[MEDIAN TEST SCORE] AS '%
KPI("%DeepSee.Median","MEDIAN",1,"%measure","Test Score","%CONTEXT")',FORMAT_STRING='
#.##':[GEND].[H1].[GENDER].&[Female]:NON EMPTY HEAD(NONEMPTYCROSSJOIN([AGED].[H1].[AG
E GROUP].MEMBERS,[AGED].[H1].[AGE BUCKET].MEMBERS),2000)

--> %InitializeAxisCache: PATIENTS: CREATING AXIS: axis: MEMBER [MEASURES].[MEDIAN TE
ST SCORE] AS '%KPI("%DeepSee.Median","MEDIAN",1,"%measure","Test Score","%CONTEXT")',
FORMAT_STRING='#.##':[GEND].[H1].[GENDER].&[Female]:NON EMPTY HEAD(NONEMPTYCROSSJOIN(
[AGED].[H1].[AGE GROUP].MEMBERS,[AGED].[H1].[AGE BUCKET].MEMBERS),2000)

--> %ExecuteAxes: PATIENTS: CURRENT AXES ARE VALID

WITH MEMBER [MEASURES].[MEDIAN TEST SCORE] AS '%KPI("%DeepSee.Median","MEDIAN",1,"%me
asure","Test Score","%CONTEXT")',FORMAT_STRING='#.##' SELECT {[MEASURES].[%COUNT],[ME
ASURES].[ALLERGY COUNT],[MEASURES].[MEDIAN TEST SCORE]} ON 0,NON EMPTY HEAD(NONEMPTYC
ROSSJOIN([AGED].[H1].[AGE GROUP].MEMBERS,[AGED].[H1].[AGE BUCKET].MEMBERS),2000) ON 1
 FROM [PATIENTS] WHERE [GEND].[H1].[GENDER].&[Female]
```

```
Patient Count Allergy Count Median Test Scor

1 0 to 29->0 to 9 69 45 73.50

2 0 to 29->10 to 19 74 47 75.50

3 0 to 29->20 to 29
```

The DeepSee Shell Best Practices Series

Run MDX queries in the DeepSee Shell

Get your MDX query performance stats from the DeepSee Shell

Reset your DeepSee cache in the DeepSee Shell

Execute an MDX query in the DeepSee Shell with/without results cache

Get your MDX query plan, MDX query trace and more from the DeepSee Shell

Example of using the MDX subquery flag in the DeepSee Shell

Debugging measures in the DeepSee Shell

#InterSystems IRIS BI (DeepSee) #MDX

---