ODTUG Kscope15
HOLLYWOOD, FLORIDA · JUNE 21-25

US-Analytics

Query Addendum

Voiding Your Warranty

The Planning Repository Exposed

SQL Server Query Addendum

Authors: **Brian Marshall**

Last Updated: **6/26/14 2:31 AM**

Table of Contents

1 Introduction

The purpose of this document is to provide an introduction to the use of the Hyperion Planning repository. Rather than highlighting specific table definitions, this document provides real-world application of the data found in the repository.

This document covers a wide range of necessary queries that should be useful to all Planning administrators ranging from a variety of standard dimensions to completion of task lists. These queries are meant as a starting point and many queries will require a certain level of updates to work for your environment.

Any questions related to these queries can be directed to Brian Marshall (bmarshall@us-analytics.com).

2 Dimensions

While virtually all objects in Planning exist in the HSP_OBJECT table, standard dimensions such as Accounts, Scenario, and Entity all have individual tables containing additional information.   Also, the HSP_DIMENSION table contains information regarding all dimensions and their associations with specific plan types.

2.1 *Dimension Information*

The dimension information query below provides a high-level view of all dimensions in your Planning Application. Each row describes what plan types a dimension is associated with, whether or not the dimension is secured, the density of the dimension, the type of dimension, and last modified information (when and by whom).

```
SELECT
       o.[OBJECT_NAME]
      ,REPLACE(REPLACE(REPLACE((SELECT pt.[TYPE_NAME]
        FROM HSP_PLAN_TYPE pt
        WHERE d.USED_IN & pt.PLAN_TYPE <> 0FOR XML Raw)
        , '"/><row TYPE_NAME="', ', '), '<row TYPE_NAME="', ''), '"/>', '' )
        AS Plan_Types
    ,CASE d.ENFORCE_SECURITY
            WHEN 1 THEN 'Secured' ELSE 'Not Secured'
       END AS Secured
    ,CASE d.DENSITY1
            WHEN 0 THEN 'Dense' WHEN 1 THEN 'Sparse'
       END AS Density
    ,CASE d.DIM_TYPE
            WHEN 0 THEN 'None'
            WHEN 1 THEN 'Account'
            WHEN 2 THEN 'Time'
            WHEN 3 THEN 'Entity'
            WHEN 6 THEN 'Attribute'
            ELSE 'None'
       END AS [Dimension_Type]
    ,o.MODIFIED AS [Last_Modified]
    ,o.MODIFIED_BY AS [Modified_By]
FROM
      HSP_DIMENSION d
INNER JOIN
      HSP_OBJECT o ON o.[OBJECT_ID] = d.DIM_ID
WHERE
      d.USED_IN <> 0
      AND d.DIM_TYPE <> 7
ORDER BY
      d.POSITION1
```

2.2 The Account Dimension

While the HSP_MEMBER table contains a variety of member-related information, some dimensions contain additional information. One such member is the account dimension. This dimension contains a variety of fields specific to accounts such as time balance, weekly spread, skip value, account type, variance reporting and more. The query below provides a link between the HSP_MEMBER, HSP_ACCOUNT, and HSP_OBJECT to provide the most important account-related information.

```
SELECT
       po.[OBJECT_NAME] AS ParentName
      ,o.[OBJECT_NAME] AS MemberName
      ,REPLACE(REPLACE(REPLACE((SELECT pt.[TYPE_NAME]
        FROM HSP_PLAN_TYPE pt
        WHERE a.USED_IN & pt.PLAN_TYPE <> 0 FOR XML Raw)
        , '"/><row TYPE_NAME="', ', '), '<row TYPE_NAME="', ''), '"/>', '' )
        AS PlanTypes
    ,pt.[TYPE_NAME] AS SourcePlanType
            ,CASE m.DATA_STORAGE
                  WHEN 0 THEN 'Store Data'
                  WHEN 1 THEN 'Never Share'
                  WHEN 2 THEN 'Label Only'
                  WHEN 3 THEN 'Shared Member'
                  WHEN 4 THEN 'Dynamic Calc and Store'
                  WHEN 5 THEN 'Dynamic'
             END AS DataStorage
            ,CASE m.CONSOL_OP
                  WHEN 0 THEN '+'
                  WHEN 1 THEN '-'
                  WHEN 2 THEN '*'
                  WHEN 3 THEN '/'
                  WHEN 4 THEN '%'
                  WHEN 5 THEN '~'
                  WHEN 6 THEN '^'
             END AS Consolidation
            ,REPLACE(REPLACE(REPLACE((
                  SELECT
                        ato.[OBJECT_NAME] + ': ' + ao.[OBJECT_NAME] AS Alias
                  FROM
                        HSP_ALIAS a
                  INNER JOIN
                        HSP_OBJECT ao ON a.MEMBER_ID = ao.[OBJECT_ID]
                  INNER JOIN
                        HSP_OBJECT ato ON a.ALIASTBL_ID = ato.[OBJECT_ID]
                  WHERE
                        a.MEMBER_ID = m.MEMBER_ID FOR XML Raw)
                        , '"/><row Alias="', ', '), '<row Alias="', ''), '"/>', ''
)
             AS MemberAlias
            ,REPLACE(REPLACE(REPLACE((
                  SELECT
                        uo.[OBJECT_NAME] AS UDA
                  FROM
                        HSP_MEMBER_TO_UDA mu
                  INNER JOIN
                        HSP_OBJECT uo ON uo.[OBJECT_ID] = mu.UDA_ID
                  WHERE
                        mu.MEMBER_ID = m.MEMBER_ID FOR XML Raw)
                        , '"/><row UDA="', ', '), '<row UDA="', ''), '"/>', '' )
             AS MemberUDA
      ,CASE a.USE_445
            WHEN 0 THEN 'None'
            WHEN 1 THEN '445'
            WHEN 2 THEN '454'
            WHEN 3 THEN '544'
            ELSE ''
       END AS WeeklySpread
      ,CASE a.TIME_BALANCE
            WHEN 0 THEN 'None'
            WHEN 1 THEN 'First'
            WHEN 2 THEN 'Last'
            WHEN 3 THEN 'Average'
            ELSE ''
       END AS TimeBalance
      ,CASE a.SKIP_VALUE
            WHEN 0 THEN 'None'
            WHEN 1 THEN 'Skip Missing'
            WHEN 2 THEN 'Skip Zeros'
            WHEN 3 THEN 'Skip Missing and Zeros'
            ELSE ''
       END AS SkipValue
      ,CASE a.ACCOUNT_TYPE
            WHEN 1 THEN 'Expense'
            WHEN 2 THEN 'Revenue'
            WHEN 3 THEN 'Asset'
            WHEN 4 THEN 'Liability'
            WHEN 5 THEN 'Equity'
            WHEN 6 THEN 'Statistical'
            WHEN 7 THEN 'Saved Assumption'
            ELSE ''
       END AS AccountType
      ,CASE a.VARIANCE_REP
            WHEN 1 THEN 'Expense'
            WHEN 2 THEN 'Non-expense'
            ELSE ''
       END AS VarianceReporting
      ,CASE a.CURRENCY_RATE
            WHEN 0 THEN 'None'
            WHEN 1 THEN 'Average'
            WHEN 2 THEN 'Ending'
            WHEN 3 THEN 'Historical'
            ELSE ''
       END AS CurrencyRate
      ,CASE m.DATA_TYPE
            WHEN 0 THEN 'Unspecified'
            WHEN 1 THEN 'Currency'
            WHEN 2 THEN 'Non-currency'
```

```
            WHEN 3 THEN 'Percentage'
            WHEN 4 THEN 'Enum'
            WHEN 5 THEN 'Date'
            WHEN 6 THEN 'Text'
            ELSE ''
       END AS DataType
      ,mf.FORMULA
FROM
      HSP_ACCOUNT a
INNER JOIN
      HSP_MEMBER m ON a.ACCOUNT_ID = m.MEMBER_ID
LEFT JOIN
      HSP_MEMBER_FORMULA mf ON m.MEMBER_ID = mf.MEMBER_ID
INNER JOIN
      HSP_OBJECT o ON m.MEMBER_ID = o.[OBJECT_ID]
INNER JOIN
      HSP_OBJECT po ON o.PARENT_ID = po.[OBJECT_ID]
INNER JOIN
      HSP_PLAN_TYPE pt ON a.SRC_PLAN_TYPE = pt.PLAN_TYPE
ORDER BY
      o.POSITION
```

2.3 The Scenario Dimension

Like the account dimension, the scenario dimension contains a variety member properties specific only to scenarios. In addition to the HSP_MEMBER and HSP_OBJECT tables, the HSP_SCENARIO table is used in the following query to provide listing of all important information regarding scenarios such as beginning balance, start year, start period, end year, and end period.

```
SELECT
       so.[OBJECT_NAME] AS Scenario
      ,CASE m.DATA_STORAGE
            WHEN 0 THEN 'Store Data'
            WHEN 1 THEN 'Never Share'
            WHEN 2 THEN 'Label Only'
            WHEN 3 THEN 'Shared Member'
            WHEN 4 THEN 'Dynamic Calc and Store'
            WHEN 5 THEN 'Dynamic'
       END AS DataStorage
      ,CASE m.CONSOL_OP
            WHEN 0 THEN '+'
            WHEN 1 THEN '-'
            WHEN 2 THEN '*'
            WHEN 3 THEN '/'
            WHEN 4 THEN '%'
            WHEN 5 THEN '~'
            WHEN 6 THEN '^'
       END AS Consolidation
      ,CASE USEBEGBAL
            WHEN 0 THEN 'No'
            WHEN 1 THEN 'Yes'
       END AS UseBegBalance
      ,mf.FORMULA AS MemberFormula
      ,fo.[OBJECT_NAME] AS FXTable
      ,syo.[OBJECT_NAME] AS StartYear
      ,spo.[OBJECT_NAME] AS StartPeriod
      ,eyo.[OBJECT_NAME] AS EndYear
      ,epo.[OBJECT_NAME] AS EndPeriod
FROM
      HSP_SCENARIO s
INNER JOIN
      HSP_MEMBER m ON s.SCENARIO_ID = m.[MEMBER_ID]
LEFT JOIN
      HSP_MEMBER_FORMULA mf ON m.MEMBER_ID = mf.MEMBER_ID
LEFT JOIN
      HSP_OBJECT fo ON s.FX_TBL = fo.[OBJECT_ID]
INNER JOIN
      HSP_OBJECT so ON s.SCENARIO_ID = so.[OBJECT_ID]
INNER JOIN
      HSP_OBJECT syo ON s.START_YR_ID = syo.[OBJECT_ID]
INNER JOIN
      HSP_OBJECT eyo ON s.END_YR_ID = eyo.[OBJECT_ID]
INNER JOIN
      HSP_OBJECT spo ON s.START_TP_ID = spo.[OBJECT_ID]
```

```
INNER JOIN
	HSP_OBJECT epo ON s.END_TP_ID = epo.[OBJECT_ID]
```

2.4 *The Entity Dimension*

Like the scenario and account dimensions, the entity dimension contains several unique fields. In this specific query, the default currency of the entity is one of those unique fields. Other fields included in this table (but not the following query) are related to Workforce Planning.

```
SELECT
       po.[OBJECT_NAME] AS ParentName
      ,o.[OBJECT_NAME] AS MemberName
      ,REPLACE(REPLACE(REPLACE((SELECT pt.[TYPE_NAME]
        FROM HSP_PLAN_TYPE pt
        WHERE e.USED_IN & pt.PLAN_TYPE <> 0 FOR XML Raw)
        , '"/><row TYPE_NAME="', ', '), '<row TYPE_NAME="', ''), '"/>', '' )
        AS PlanTypes
      ,CASE m.DATA_STORAGE
            WHEN 0 THEN 'Store Data'
            WHEN 1 THEN 'Never Share'
            WHEN 2 THEN 'Label Only'
            WHEN 3 THEN 'Shared Member'
            WHEN 4 THEN 'Dynamic Calc and Store'
            WHEN 5 THEN 'Dynamic'
       END AS DataStorage
      ,CASE m.CONSOL_OP
            WHEN 0 THEN '+'
            WHEN 1 THEN '-'
            WHEN 2 THEN '*'
            WHEN 3 THEN '/'
            WHEN 4 THEN '%'
            WHEN 5 THEN '~'
            WHEN 6 THEN '^'
       END AS Consolidation
      ,REPLACE(REPLACE(REPLACE((
            SELECT
                  ato.[OBJECT_NAME] + ': ' + ao.[OBJECT_NAME] AS Alias
            FROM
                  HSP_ALIAS a
            INNER JOIN
                  HSP_OBJECT ao ON a.MEMBER_ID = ao.[OBJECT_ID]
            INNER JOIN
                  HSP_OBJECT ato ON a.ALIASTBL_ID = ato.[OBJECT_ID]
            WHERE
                  a.MEMBER_ID = m.MEMBER_ID FOR XML Raw)
                  , '"/><row Alias="', ', '), '<row Alias="', ''), '"/>', '' )
       AS MemberAlias
      ,REPLACE(REPLACE(REPLACE((
            SELECT
                  uo.[OBJECT_NAME] AS UDA
            FROM
                  HSP_MEMBER_TO_UDA mu
            INNER JOIN
                  HSP_OBJECT uo ON uo.[OBJECT_ID] = mu.UDA_ID
            WHERE
```

```
                  mu.MEMBER_ID = m.MEMBER_ID FOR XML Raw)
                  , '"/><row UDA="', ', '), '<row UDA="', ''), '"/>', '' )
       AS MemberUDA
      ,CASE m.DATA_TYPE
            WHEN 0 THEN 'Unspecified'
            WHEN 1 THEN 'Currency'
            WHEN 2 THEN 'Non-currency'
            WHEN 3 THEN 'Percentage'
            WHEN 4 THEN 'Enum'
            WHEN 5 THEN 'Date'
            WHEN 6 THEN 'Text'
            ELSE ''
       END AS DataType
      ,mf.FORMULA AS MemberFormula
      ,c.SYMBOL AS DefaultCurrency
FROM
      HSP_ENTITY e
INNER JOIN
      HSP_MEMBER m ON e.ENTITY_ID = m.[MEMBER_ID]
INNER JOIN
      HSP_OBJECT o ON e.ENTITY_ID = o.[OBJECT_ID]
INNER JOIN
      HSP_OBJECT po ON o.PARENT_ID = po.[OBJECT_ID]
INNER JOIN
      HSP_CURRENCY c ON e.DEFAULT_CURRENCY = c.CURRENCY_ID
LEFT JOIN
      HSP_MEMBER_FORMULA mf ON m.MEMBER_ID = mf.MEMBER_ID
```

2.5 Other Dimensions

The following query presents a more generic approach to determining information regarding dimensions. Simply provide a dimension ID and the query will return all of the members of that dimension and other relevant information.

```
SELECT
       po.[OBJECT_NAME] AS Parent_Name
      ,o.[OBJECT_NAME] AS Member_Name
      ,CASE m.DATA_STORAGE
            WHEN 0 THEN 'Store Data'
            WHEN 1 THEN 'Never Share'
            WHEN 2 THEN 'Label Only'
            WHEN 3 THEN 'Shared Member'
            WHEN 4 THEN 'Dynamic Calc and Store'
            WHEN 5 THEN 'Dynamic'
       END AS Data_Storage
      ,CASE m.CONSOL_OP
            WHEN 0 THEN '+'
            WHEN 1 THEN '-'
            WHEN 2 THEN '*'
            WHEN 3 THEN '/'
            WHEN 4 THEN '%'
            WHEN 5 THEN '~'
            WHEN 6 THEN '^'
       END AS Consolidation
      ,REPLACE(REPLACE(REPLACE((
            SELECT
                  ato.[OBJECT_NAME] + ': ' + ao.[OBJECT_NAME] AS Alias
            FROM
                  HSP_ALIAS a
            INNER JOIN
                  HSP_OBJECT ao ON a.MEMBER_ID = ao.[OBJECT_ID]
            INNER JOIN
                  HSP_OBJECT ato ON a.ALIASTBL_ID = ato.[OBJECT_ID]
            WHERE
                  a.MEMBER_ID = m.MEMBER_ID FOR XML Raw)
                  , '"/><row Alias="', ', '), '<row Alias="', ''), '"/>', '' )
       AS Member_Alias
      ,REPLACE(REPLACE(REPLACE((
            SELECT
                  ado.[OBJECT_NAME] + ': ' + amo.[OBJECT_NAME] AS Attribute
            FROM
                  HSP_MEMBER_TO_ATTRIBUTE ma
            INNER JOIN
                  HSP_OBJECT ado ON ado.[OBJECT_ID] = ma.ATTR_ID
            INNER JOIN
                  HSP_OBJECT amo ON amo.[OBJECT_ID] = ma.ATTR_MEM_ID
            WHERE
                  ma.MEMBER_ID = m.MEMBER_ID FOR XML Raw)
                  , '"/><row Attribute="', ', '), '<row Attribute="', ''), '"/>',
'' )
```

```
       AS Member_Attributes
      ,REPLACE(REPLACE(REPLACE((
            SELECT
                  uo.[OBJECT_NAME] AS UDA
            FROM
                  HSP_MEMBER_TO_UDA mu
            INNER JOIN
                  HSP_OBJECT uo ON uo.[OBJECT_ID] = mu.UDA_ID
            WHERE
                  mu.MEMBER_ID = m.MEMBER_ID FOR XML Raw)
                  , '"/><row UDA="', ', '), '<row UDA="', ''), '"/>', '' )
       AS Member_UDA
      ,*
FROM
      HSP_MEMBER m
INNER JOIN
      HSP_OBJECT o ON m.MEMBER_ID = o.[OBJECT_ID]
INNER JOIN
      HSP_OBJECT po ON o.PARENT_ID = po.[OBJECT_ID]
WHERE
      DIM_ID = 52823
```

3 User Access

Most objects in Planning can be secured. The purpose of this query is to provide a high-level view of the secured objects in your Planning application. It is important to note that the HSP_OBJECT_TYPE table is joined using a left join. This was done because this table is incomplete and therefore must be handled with additional code for missing items such as Task Lists.

```
SELECT
       o.[OBJECT_NAME] AS [Secured_Object]
      ,uo.[OBJECT_NAME] AS [User_Name]
      ,CASE a.ACCESS_MODE
            WHEN 3 THEN 'Read'
            WHEN 1 THEN 'Write'
            WHEN -1 THEN 'Deny'
       END AS Access_Level
      ,CASE o.OBJECT_TYPE
            WHEN 24 THEN 'Task List'
            WHEN 107 THEN 'Composite Form'
            ELSE ot.[TYPE_NAME]
       END AS Object_Type
      ,CASE a.FLAGS
            WHEN 0 THEN 'Member'
            WHEN 5 THEN 'Children'
            WHEN 6 THEN 'Children (Inclusive)'
            WHEN 8 THEN 'Descendants'
            WHEN 9 THEN 'Descendants (Inclusive)'
       END AS Access_Level
FROM
      HSP_ACCESS_CONTROL a
INNER JOIN
      HSP_OBJECT uo ON a.[USER_ID] = uo.[OBJECT_ID]
INNER JOIN
      HSP_OBJECT o ON a.[OBJECT_ID] = o.[OBJECT_ID]
LEFT JOIN
      HSP_OBJECT_TYPE ot ON o.OBJECT_TYPE = ot.OBJECT_TYPE
ORDER BY
      o.OBJECT_TYPE
      ,o.[OBJECT_NAME]
```

4 Forms

Forms are one of the many complex objects in a Planning application. Often times, we wonder what exactly is out there. If we make a change to a dimension, what will be impacted? Who has access to the forms we might be changing? The purpose of the following query is to quickly provide a look at all forms contained in the Planning application. The query contains all of the dimensional layout information and security access among other fields.

```
SELECT
      o.[OBJECT_NAME] AS Form_Name
      ,REPLACE(REPLACE(REPLACE((SELECT flo.[OBJECT_NAME]
        FROM HSP_FORM_LAYOUT fl
        INNER JOIN HSP_OBJECT flo on fl.DIM_ID = flo.[OBJECT_ID]
        WHERE fl.FORM_ID = f.FORM_ID AND fl.LAYOUT_TYPE = 0 FOR XML Raw)
        , '"/><row OBJECT_NAME="', ', '), '<row OBJECT_NAME="', ''), '"/>', '' )
        AS POV_Dimensions
      ,REPLACE(REPLACE(REPLACE((SELECT flo.[OBJECT_NAME]
        FROM HSP_FORM_LAYOUT fl
        INNER JOIN HSP_OBJECT flo on fl.DIM_ID = flo.[OBJECT_ID]
        WHERE fl.FORM_ID = f.FORM_ID AND fl.LAYOUT_TYPE = 1 FOR XML Raw)
        , '"/><row OBJECT_NAME="', ', '), '<row OBJECT_NAME="', ''), '"/>', '' )
        AS Page_Dimensions
    ,REPLACE(REPLACE(REPLACE((SELECT flo.[OBJECT_NAME]
        FROM HSP_FORM_LAYOUT fl
        INNER JOIN HSP_OBJECT flo on fl.DIM_ID = flo.[OBJECT_ID]
        WHERE fl.FORM_ID = f.FORM_ID AND fl.LAYOUT_TYPE = 2 FOR XML Raw)
        , '"/><row OBJECT_NAME="', ', '), '<row OBJECT_NAME="', ''), '"/>', '' )
        AS Row_Dimensions
    ,REPLACE(REPLACE(REPLACE((SELECT flo.[OBJECT_NAME]
        FROM HSP_FORM_LAYOUT fl
        INNER JOIN HSP_OBJECT flo on fl.DIM_ID = flo.[OBJECT_ID]
        WHERE fl.FORM_ID = f.FORM_ID AND fl.LAYOUT_TYPE = 3 FOR XML Raw)
        , '"/><row OBJECT_NAME="', ', '), '<row OBJECT_NAME="', ''), '"/>', '' )
        AS Column_Dimensions
    ,REPLACE(REPLACE(REPLACE((SELECT fla.[OBJECT_NAME]
        FROM HSP_FORM_ATTRIBUTES fa
        INNER JOIN HSP_OBJECT fla on fa.DIM_ID = fla.[OBJECT_ID]
        WHERE fa.FORM_ID = f.FORM_ID FOR XML Raw)
        , '"/><row OBJECT_NAME="', ', '), '<row OBJECT_NAME="', ''), '"/>', '' )
        AS Attribute_Dimensions
    ,REPLACE(REPLACE(REPLACE((
            SELECT fc.CALC_NAME
                  + CASE fc.RUN_ON_LOAD WHEN 1 THEN ' (Run on load)' ELSE '' END
                  + CASE fc.RUN_ON_SAVE WHEN 1 THEN ' (Run on save)' ELSE '' END
                  as CALCs
        FROM HSP_FORM_CALCS fc
        WHERE fc.FORM_ID = f.FORM_ID FOR XML Raw)
        , '"/><row CALCs="', ', '), '<row CALCs="', ''), '"/>', '' )
        AS Calcs
    ,REPLACE(REPLACE(REPLACE((
            SELECT aco.[OBJECT_NAME]
        FROM HSP_ACCESS_CONTROL ac
        INNER JOIN HSP_OBJECT aco ON aco.[OBJECT_ID] = ac.[USER_ID]
        WHERE ac.[OBJECT_ID] = f.FORM_ID AND ac.ACCESS_MODE = 3 FOR XML Raw)
        , '"/><row OBJECT_NAME="', ', '), '<row OBJECT_NAME="', ''), '"/>', '' )
        AS Users_Write
    ,REPLACE(REPLACE(REPLACE((
            SELECT aco.[OBJECT_NAME]
        FROM HSP_ACCESS_CONTROL ac
        INNER JOIN HSP_OBJECT aco ON aco.[OBJECT_ID] = ac.[USER_ID]
        WHERE ac.[OBJECT_ID] = f.FORM_ID AND ac.ACCESS_MODE = 1 FOR XML Raw)
        , '"/><row OBJECT_NAME="', ', '), '<row OBJECT_NAME="', ''), '"/>', '' )
        AS Users_Read
    ,REPLACE(REPLACE(REPLACE((
            SELECT aco.[OBJECT_NAME]
        FROM HSP_ACCESS_CONTROL ac
        INNER JOIN HSP_OBJECT aco ON aco.[OBJECT_ID] = ac.[USER_ID]
        WHERE ac.[OBJECT_ID] = f.FORM_ID AND ac.ACCESS_MODE = -1 FOR XML Raw)
        , '"/><row OBJECT_NAME="', ', '), '<row OBJECT_NAME="', ''), '"/>', '' )
        AS Users_Denied
      ,o.MODIFIED AS Last_Modified
      ,o.MODIFIED_BY AS Modified_By
FROM
      HSP_FORM f
INNER JOIN
      HSP_OBJECT o ON f.FORM_ID = o.[OBJECT_ID]
```

5 Task Lists

As tools like Smart View improve and add access to Planning objects like Task Lists, Planning administrators begin to need more visibility to completion of these task lists. The following query is a high-level sample of a listing of task list completion. The query will return all users who have access to each task list and also the completion status of that task list. This query could be modified to include a completion percentage and perhaps a number of tasks remaining.

```
SELECT
       uo.[OBJECT_NAME] AS Username
      ,o.[OBJECT_NAME] AS TaskList
      ,CASE
           WHEN ut.COMPLETED_DATE IS NULL THEN 'Not Complete'
           ELSE 'Complete'
      END AS TaskListStatus
      ,*
FROM
      HSP_ACCESS_CONTROL ac
INNER JOIN
      HSP_OBJECT o ON ac.[OBJECT_ID] = o.[OBJECT_ID]
INNER JOIN
      HSP_OBJECT uo ON ac.[USER_ID] = uo.[OBJECT_ID]
INNER JOIN
      HSP_TASK t ON ac.[OBJECT_ID] = t.TASK_ID
LEFT JOIN
      HSP_USER_TASK ut ON ac.[USER_ID] = ut.[USER_ID] AND ac.[OBJECT_ID] =
ut.TASK_ID
WHERE
      o.OBJECT_TYPE = 24
```

6 Approvals (Process Management)

In version 11.1.2.1 of Planning, Process Management has been renamed to Approvals. The tables in the repository are still prefixed with HSP_PM. The following query outlines the paths available for modifying an approval. Essentially, this query shows what impact an action will have on the state of an approval process.

```
SELECT
       REPLACE(REPLACE(fs.NAME, 'LABEL_', ''), '_', ' ') AS [FromState]
      ,REPLACE(REPLACE(a.NAME, 'LABEL_', ''), '_', ' ') AS [Action]
      ,REPLACE(REPLACE(ts.NAME, 'LABEL_', ''), '_', ' ') AS [ToState]
      ,rl.PM_RULE_LIST_NAME AS ApprovalTemplate
      ,CASE r.CHANGABLE_BY
            WHEN 0 THEN 'Owner'
            WHEN 1 THEN 'Anyone with access'
            WHEN 2 THEN 'Admin'
            WHEN 3 THEN 'No one'
            WHEN 8 THEN 'Unknown 8'
            WHEN 9 THEN 'Unknown 9'
            ELSE 'Unknown'
       END AS ChangableBy
      ,CASE r.NEW_OWNER
            WHEN 0 THEN 'Owner'
            WHEN 1 THEN 'Anyone with access'
            WHEN 2 THEN 'Admin'
            WHEN 3 THEN 'No one'
            WHEN 4 THEN 'Self'
            WHEN 8 THEN 'Unknown 8'
            WHEN 9 THEN 'Unknown 9'
            WHEN 10 THEN 'Unknown 10'
            WHEN 11 THEN 'Unknown 11'
            WHEN 12 THEN 'Unknown 12'
            WHEN 13 THEN 'Unknown 13'
            WHEN 14 THEN 'Unknown 14'
            ELSE 'Unknown'
       END AS NewOwner
FROM
      HSP_PM_RULES r
INNER JOIN
      HSP_PM_ACTIONS a ON r.ACTION_ID = a.ACTION_ID
INNER JOIN
      HSP_PM_STATES fs ON r.FROM_STATE_ID = fs.STATE_ID
INNER JOIN
      HSP_PM_STATES ts ON r.TO_STATE_ID = ts.STATE_ID
INNER JOIN
      HSP_PM_RULE_LIST rl ON r.PM_RULE_LIST_ID = rl.PM_RULE_LIST_ID
```

7 Supporting Detail

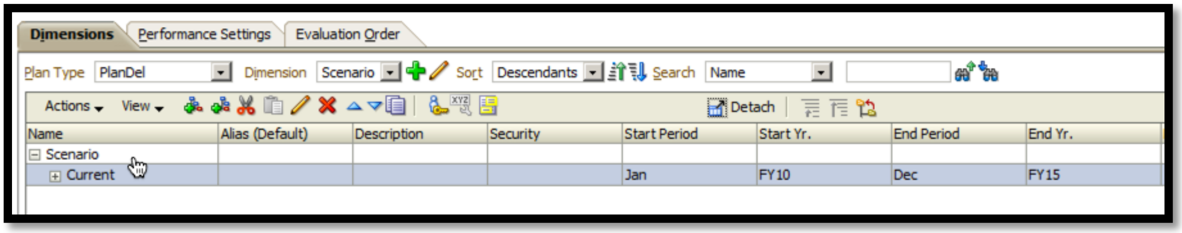
Have you ever wondered where all of the supporting detail exists in your Planning application? Well…we did, so this query was born. This is a fairly simple query that simply shows where supporting detail exists by plan type. In this example, 8 dimensions are specified because the demo we are using has only 8 dimensions. The tables contain up to 20 dimension intersections.

```
SELECT
       pt.[TYPE_NAME] AS PlanType
      ,o1.[OBJECT_NAME] AS DIM1
      ,o2.[OBJECT_NAME] AS DIM2
      ,o3.[OBJECT_NAME] AS DIM3
      ,o4.[OBJECT_NAME] AS DIM4
      ,o5.[OBJECT_NAME] AS DIM5
      ,o6.[OBJECT_NAME] AS DIM6
      ,o7.[OBJECT_NAME] AS DIM7
      ,o8.[OBJECT_NAME] AS DIM8
FROM
      HSP_COLUMN_DETAIL sd
INNER JOIN
      HSP_PLAN_TYPE pt ON pt.PLAN_TYPE = sd.PLAN_TYPE
INNER JOIN
      HSP_OBJECT o1 ON o1.[OBJECT_ID] = sd.DIM1
INNER JOIN
      HSP_OBJECT o2 ON o2.[OBJECT_ID] = sd.DIM2
INNER JOIN
      HSP_OBJECT o3 ON o3.[OBJECT_ID] = sd.DIM3
INNER JOIN
      HSP_OBJECT o4 ON o4.[OBJECT_ID] = sd.DIM4
INNER JOIN
      HSP_OBJECT o5 ON o5.[OBJECT_ID] = sd.DIM5
INNER JOIN
      HSP_OBJECT o6 ON o6.[OBJECT_ID] = sd.DIM6
INNER JOIN
      HSP_OBJECT o7 ON o7.[OBJECT_ID] = sd.DIM7
INNER JOIN
      HSP_OBJECT o8 ON o8.[OBJECT_ID] = sd.DIM8
```

8 Deleting Years

As applications age, often it is necessary to remove historical information. The issue is that Planning does not offer an interface to allow you to remove a year. A year can be deleted by a few queries against the Planning repository. In this example, **FY10** data is being deleted and **FY11** will become the earliest year in the application. Follow these steps to delete a year from Hyperion Planning:

1. Ensure that no scenarios reference the year you are deleting. In this instance, the scenario **Current** must be modified to be **FY11** before **FY10** can be deleted.



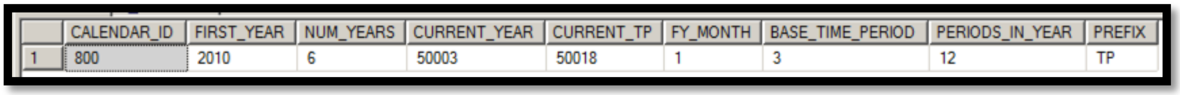
Plan Type: PlanDel Dimension: Scenario Sort: Descendants Search: Name

Name	Alias (Default)	Description	Security	Start Period	Start Yr.	End Period	End Yr.
⊟ Scenario							
⊞ Current				Jan	FY10	Dec	FY15

2. Connect to the Planning Repository and open a new query.
3. Select all columns from the **HSP_CALENDAR** table to determine if any changes are necessary:

```
SELECT * FROM HSP_CALENDAR
```

4. Note that **2010** is the first year and **50003** is the ID of the current year:



	CALENDAR_ID	FIRST_YEAR	NUM_YEARS	CURRENT_YEAR	CURRENT_TP	FY_MONTH	BASE_TIME_PERIOD	PERIODS_IN_YEAR	PREFIX
1	800	2010	6	50003	50018	1	3	12	TP

5. It can be assumed that **50003** is likely the ID of **FY10**, but to verify, run this query:

```
SELECT * FROM HSP_OBJECT WHERE OBJECT_ID = 50003
```

6. Next, use this query to determine the ID of the new current year (**FY11**):

```
SELECT * FROM HSP_OBJECT WHERE OBJECT_NAME = 'FY11'
```

7. Once the new ID has been determined, use this query to update the **HSP_CALENDAR** table with the new first year and current year information:

```
UPDATE HSP_CALENDAR SET FIRST_YEAR = '2011', CURRENT_YEAR = 50004
```

8. Next, the record for the year being deleted must be deleted from the **HSP_MEMBER** table using this query:

```
DELETE FROM HSP_MEMBER WHERE MEMBER_ID = 50003
```

9. Once the member record has been deleted, the final SQL Query is to remove the actual object using this query:

```
DELETE FROM HSP_OBJECT WHERE OBJECT_ID = 50003
```

10. Once the changes have been to the repository, Planning must be restarted for any changes to take affect.

9 Deleting a Dimension

Sometimes dimensions become no longer necessary as business processes change. Other times, an admin accidentally added a dimension thinking that surely they could delete it if they needed to. Planning however does not actually support deleting a dimension. Prior to deleting a dimension, make sure that you have a completed the following tasks:

- Remove any dimension members.
- Remove any attribute dimensions associated with the dimension.
- Remove any additional aliases associated with the dimension.
- Remove any references to the dimension from all forms and reports.
- Remove any security rights from the dimension.
- Back up your planning repository.

Once you have completed these tasks, you are ready to delete the dimension from the repository. Follow these steps to delete a dimension from Planning:

1. Determine the ID of the dimension you are deleting:
   ```
   SELECT * FROM HSP_OBJECT WHERE OBJECT_NAME = 'ToBeDeleted'
   ```
2. Delete any references to the dimension in the POV table:
   ```
   DELETE FROM HSP_MRU_MEMBERS WHERE MEMBER_ID = 50524
   ```
3. Delete the member from the HSP_MEMBER table:
   ```
   DELETE FROM HSP_MEMBER WHERE MEMBER_ID = 50524
   ```
4. Delete the dimension from the HSP_DIMENSION table:
   ```
   DELETE FROM HSP_DIMENSION WHERE DIM_ID = 50524
   ```
5. Delete the unique name reference from HSP_UNIQUE_NAMES table:
   ```
   DELETE FROM HSP_UNIQUE_NAMES WHERE OBJECT_ID = 50524
   ```
6. Delete the actual object from the HSP_OBJECT table:
   ```
   DELETE FROM HSP_OBJECT WHERE OBJECT_ID = 50524
   ```
7. Restart planning and verify that the dimension has been deleted.
8. Test everything thoroughly.