Lesson 7: Variables and Dashboard Prompts

This class has been designed to provide the knowledge and tools necessary to use OBIEE ad hoc Answers to its fullest potential. This particular lesson on Dashboard Prompts is not intended as a preparation to start developing dashboards, but rather an overview of basic OBIEE Dashboard functionality and its design potential.

Each functional business area will typically authorize one or two people to develop ‘official’ reports and dashboards for distribution to a set of end users. Functional offices have control over “publishing” in OBIEE, with a functional QA Administrator who will oversee development and govern migration of new or revised reports and dashboards from OBIEEDEV to OBIEETEST, and then on to OBIEEPROD.

Exercise 7a: Variables

Although knowledge of the three types of variables in OBIEE isn't strictly required in order to build analyses and/or dashboards, such knowledge is useful. In this exercise, we'll define and discuss the three types of variables in OBIEE: Repository variables, Session variables, and Presentation variables.

Variables: Definition

A **variable** is nothing more than a single piece of information stored for later use. It may be a character string, a number, or a date. It must be a simple, one-cell object: one string, one number, or one date.

When imagining variables, it is helpful to think of three shelves on the wall. Each shelf is a different color (red, silver, or purple), and each shelf holds some number of paper cups of the same color as the shelf. Each paper cup has a label on the outside, and a small slip of paper inside, on which is written a character string, number or date.

The top shelf is red, and holds red cups. Each of those red cups has a label on it, and inside each red paper cup is a slip of paper with a character string, a number, or a date on it. Those red paper cups represent **Repository** variables. The label on the outside of the cup is the **name** of the Repository variable, and the single piece of information on the slip of paper inside of the cup is the **value** of the Repository variable.

Similarly, there is a silver shelf, on which sit silver paper cups with labels, each paper cup containing a slip of paper with a single piece of information. The silver cups on the silver shelf represent **Session** variables.

Finally, there is a purple shelf with purple cups, each with a slip of paper with a single piece of information, each representing a **Presentation variable.**

Repository Variables

Repository variables (the red cups) are created and populated by the OBIEE Administrator as part of the metadata repository. Values of repository variables are reset on a regularly scheduled basis, and cannot be changed by any user. The value of any given repository variable (i.e. the string, number, or date on the piece of paper in the labeled red paper cup) is the same for all users. The date of the most recent data load would be an example of a repository variable.

Session Variables

Session variables (the silver cups) are also created by the OBIEE Administrator as part of the metadata repository. The values of session variables are established when a user logs in to OBIEE, and the same session variable may have a different value for each user.

There are two “flavors” of system variables:

System session variables: User ID, the user’s data security groups, and the user’s web catalog group(s) are all examples of system session variables. These variables are not eligible to be changed by any user.

Non-system session variables: These are variables which are defined by the OBIEE administrator for whatever purpose may be required. The administrator may allow users to change the values of any or all session variables. The user’s office location might be an example of a session variable that may not be changed by the user. System variables defined for population by dashboard prompts and subsequently used as filter criteria would be examples of session variables that **may** be changed by the user. After their initialization during the user’s session login, populating such system variables can **only** be accomplished with a dashboard prompt.

Presentation Variables

Presentation Variables are created by, and exist only in the context of, a Dashboard Prompt. The values of Presentation variables may be used as filtering conditions for any analyses on the dashboard(s) on which the dashboard prompt is present. The use of a dashboard prompt is the **only** way to create a presentation variable.

There is no way to just “create and populate a variable” for use in OBIEE. It must be defined in the repository or on a dashboard prompt.

Dashboard Prompts vs. Column Filter Prompts

In an earlier lesson, we learned how to create Column Filter Prompts, which can be used to provide users with a filtering and selection mechanism in the absence of a dashboard. If an analysis isn’t intended for deployment to a dashboard, the Column Filter Prompt method of filtering an analysis is the proper mechanism to use. Dashboard prompts are used when deployment of an analysis on a dashboard is desired.

Dashboard Prompts

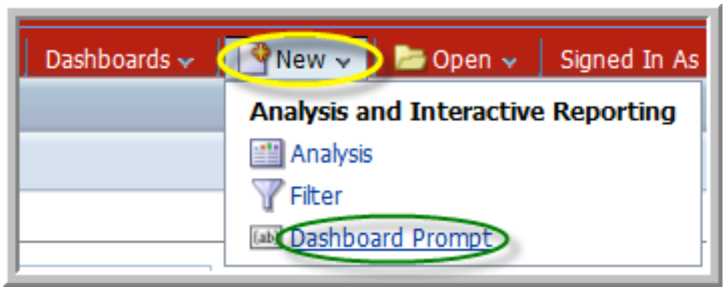
Dashboard Prompts are used in conjunction with Answers analyses in which one or more Answers Filters are set to “is prompted”, or which feature a filter condition based on the values of one or more Presentation variables. For example, an Answers analysis might require the user to select a Fiscal Month from a list of all values, whereupon the Answers analysis would display data for that Fiscal Month. That list of all values can be included in an object known as a **Dashboard Prompt,** which requires the Answers analysis to have an “is prompted” filter on the Fiscal Month column.

The prompt we are about to create can be included on one or more Dashboards, and can be used to control the scope of analyses on an individual dashboard page, or an entire dashboard.

Exercise 7b: Filtering using Presentation Variables

In this exercise, we’ll create a dashboard prompt in which our users will enter two numbers to be stored into two Presentation Variables. We’ll then create an Answers analysis with a filter that returns only those rows in which Corrected Hours falls between our two input values. Presentation Variables are created using the optional **Set Variable** field in a dashboard prompt.

1. Create a new Dashboard Prompt using the New icon on the toolbar at the top of the OBIEE Answers screen. Select **Dashboard Prompt** from the dropdown list.



2. As was the case when creating an analysis, a dashboard prompt is sourced from a single Subject Area. Select the **Training** Subject Area as the source for this dashboard prompt.

   When you get additional Answers access to other Subject Areas, those will appear in this drop down list, just as they will in the Answers Subject Area list on the default Answers page.

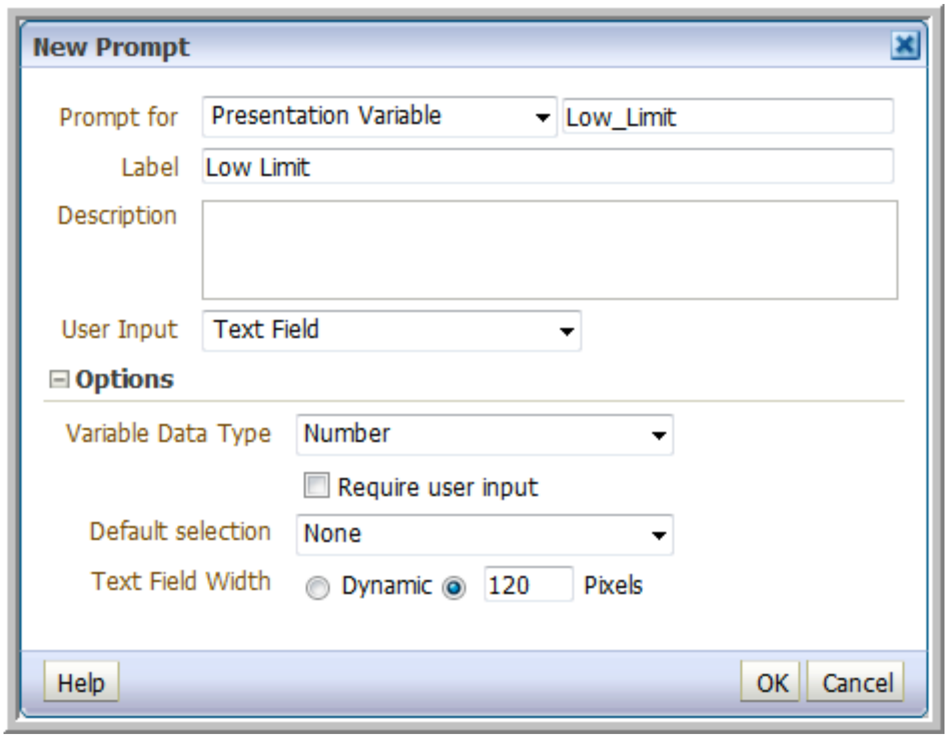
3. This dashboard prompt will populate two **Presentation Variables**, based on the user's input into two dashboard prompt boxes. Click the green ✚ sign at the top right to begin the creation of a new prompt.

This particular prompt will be a **Variable Prompt**. Variable Prompts permit us to present the user with data entry boxes, as opposed to Column Prompts which display a list of values from a particular column (such as Fiscal Month) for the users to choose from.

4. Select **Variable Prompt** from the dropdown list.

5. Create the Prompt as shown in the screenshot below, then click **OK**. The key features of this Dashboard Prompt are:

   a. The user's entry will be stored in a Presentation Variable called **Low_Limit**.
   b. The words **Low Limit** will be displayed above the data entry box.
   c. The User Input is a **Text Field**, a simple field that accepts typed values.
   d. The Variable Data Type is a number. Only digits 0-9 are allowed in this prompt.



New Prompt

Prompt for: Presentation Variable ▾ Low_Limit

Label: Low Limit

Description:

User Input: Text Field ▾

⊟ Options

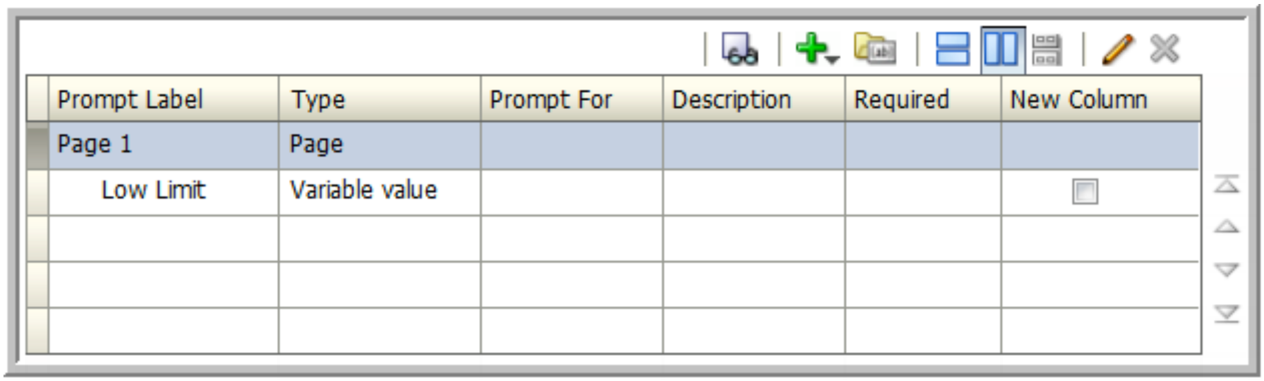
Variable Data Type: Number ▾

☐ Require user input

Default selection: None ▾

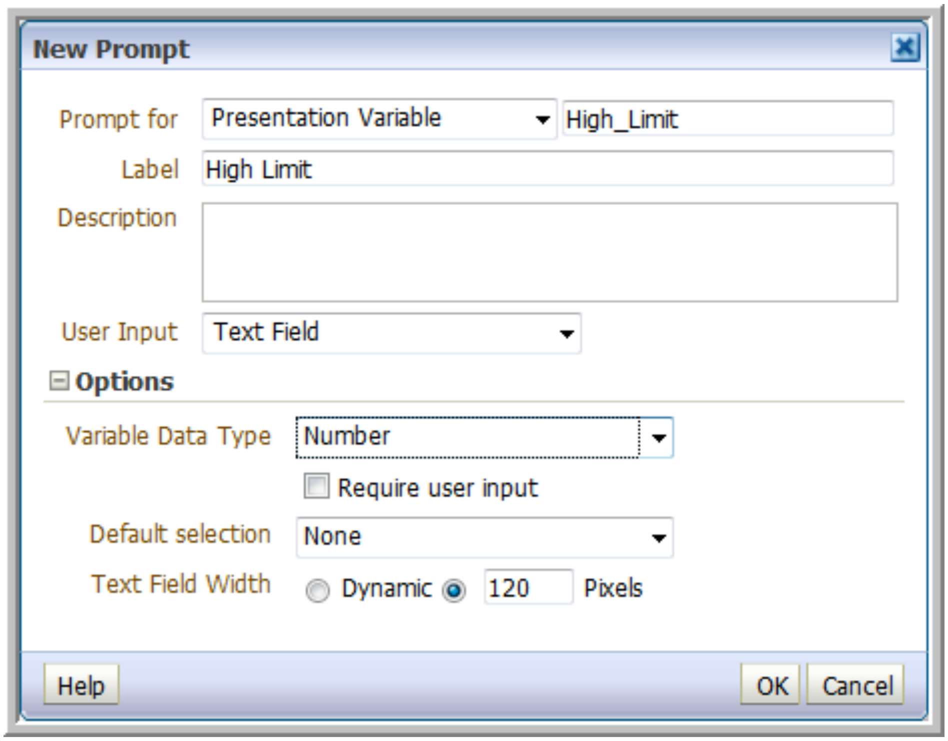
Text Field Width: ○ Dynamic ● 120 Pixels

6. Compare your results to this screenshot:



Prompt Label	Type	Prompt For	Description	Required	New Column
Page 1	Page				
Low Limit	Variable value				□

7. Starting with the green ✚ again, add another Variable Prompt, with these characteristics:



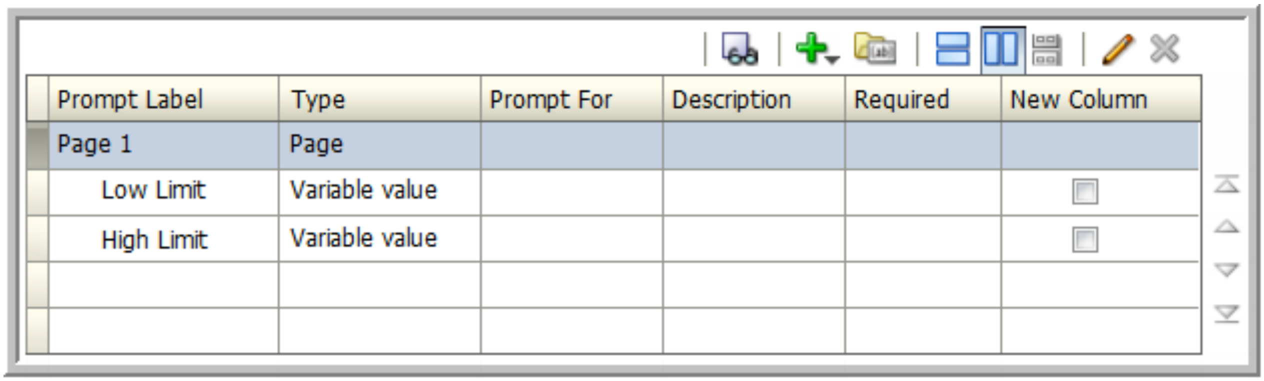
New Prompt

Prompt for	Presentation Variable ▾ High_Limit
Label	High Limit
Description	
User Input	Text Field ▾

⊟ **Options**

Variable Data Type	Number ▾
	□ Require user input
Default selection	None ▾
Text Field Width	○ Dynamic ● 120 Pixels

8. Compare your results to this screenshot:

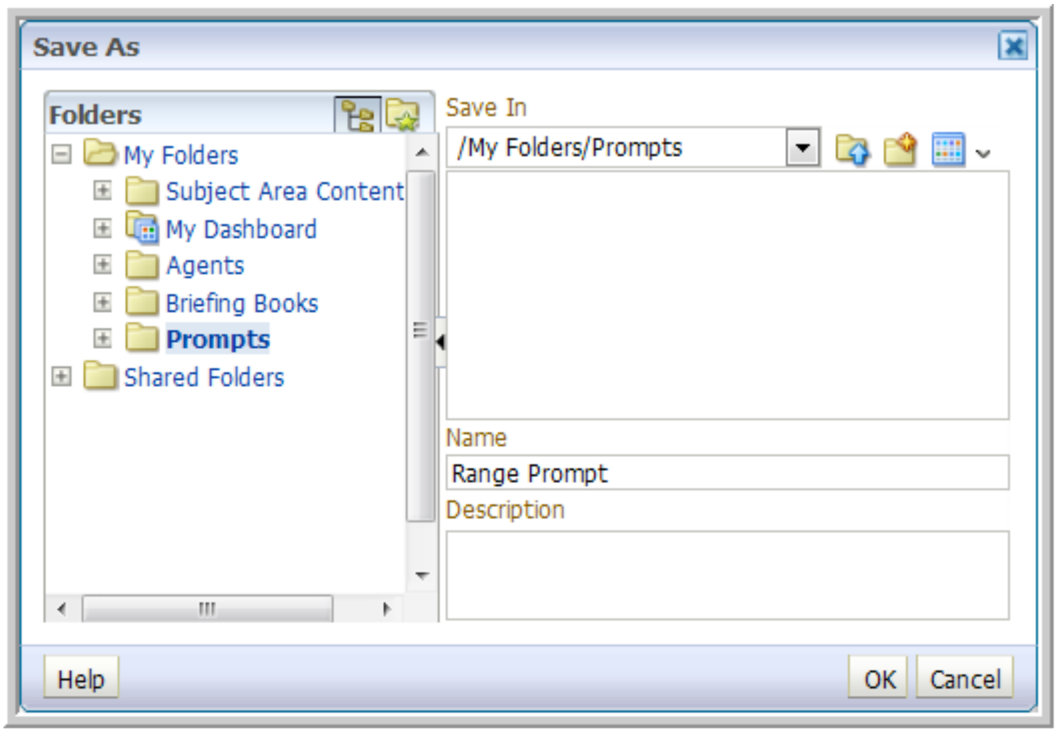


Prompt Label	Type	Prompt For	Description	Required	New Column
Page 1	Page				
Low Limit	Variable value				□
High Limit	Variable value				□

9. Use the Save icon to save this re-usable prompt in **My Folders**, in a **NEW subfolder** called **Prompts**.

★	This is a Cornell **Best Practice** for organizing all of the objects in **My Folders**

a. In the Save As dialog box, use the **New Folder** icon at the top to create Prompts.
b. Double-click the new Prompts folder in the list.
c. Save your new Dashboard Prompt in that subfolder as **Range Prompt.**



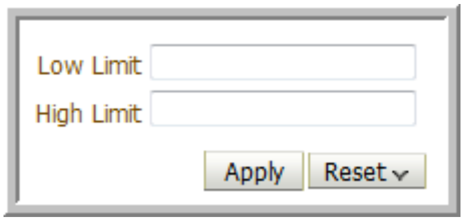
Save As

Save In: /My Folders/Prompts

Name: Range Prompt

Description:

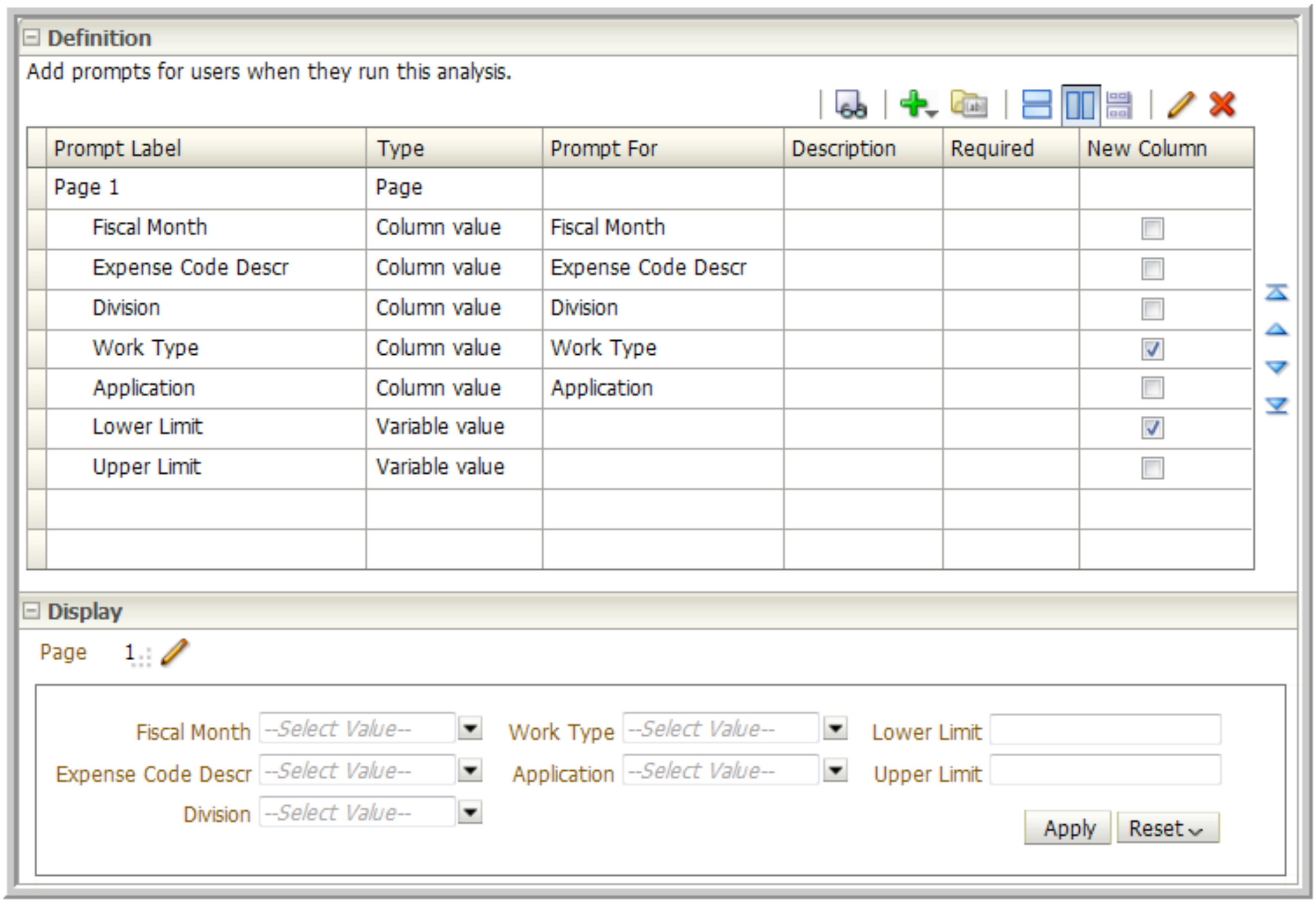
10. View the Display section at the bottom of the screen. Notice that the two prompt entry boxes are arranged one above the other. This is the default behavior in OBIEE 11g.



11. Also notice the toolbar at the top right of the Definition section at the top of the screen. Notice that the icon for the Column-based Layout is currently highlighted. This is causing the prompts to be arranged in one vertical column.
12. Let's switch to a Row-based Layout, in which the prompts are laid out side-by-side. Click the Row-based Layout icon. Now the prompts should be side-by-side on one row.

New Row or **New Column**: By default, prompts for all of the prompts within a single dashboard prompt will appear on one row, or one column, depending on which icon is clicked, as shown above. If a checkbox under New Row or New Column is checked, it means that a new row or column should be started with that prompt.

This example dashboard prompt contains five Column Prompts and two Variable Prompts.



⊟ Definition

Add prompts for users when they run this analysis.

Prompt Label	Type	Prompt For	Description	Required	New Column
Page 1	Page				
Fiscal Month	Column value	Fiscal Month			☐
Expense Code Descr	Column value	Expense Code Descr			☐
Division	Column value	Division			☐
Work Type	Column value	Work Type			☑
Application	Column value	Application			☐
Lower Limit	Variable value				☑
Upper Limit	Variable value				☐

⊟ Display

Page 1

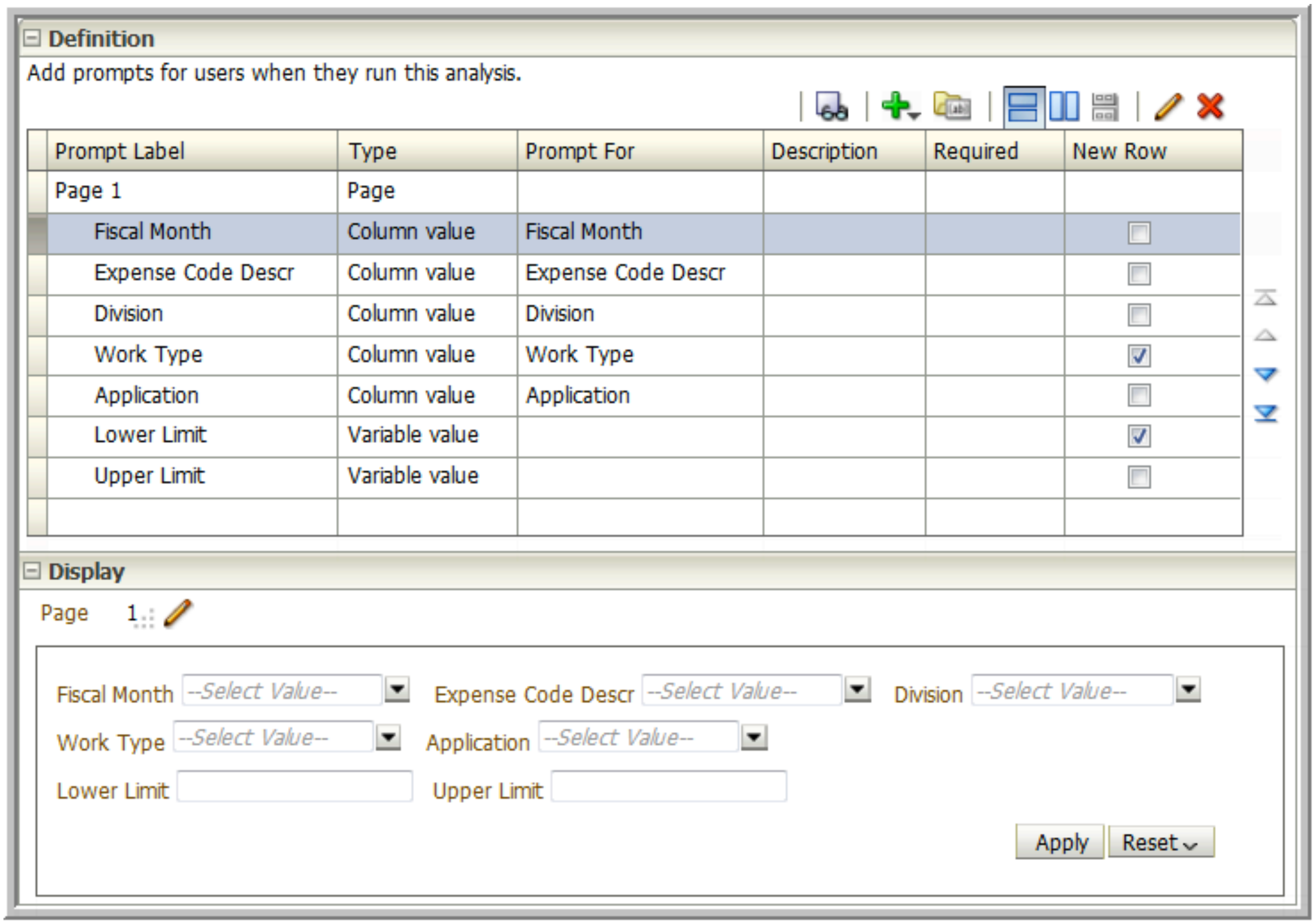
Fiscal Month --Select Value--
Expense Code Descr --Select Value--
Division --Select Value--

Work Type --Select Value--
Application --Select Value--

Lower Limit
Upper Limit

In this example, the Column-based Layout icon is selected. By default, all 7 of these prompts will be arranged vertically, in one column. But notice that under the **New Column** heading there are two checkboxes ticked, one for Work Type, another for Lower Limit. Notice that in the Display section, the Work Type prompt starts a new column, as does the Lower Limit prompt.

The same example dashboard with a default Row-based Layout is shown here.



Definition

Add prompts for users when they run this analysis.

Prompt Label	Type	Prompt For	Description	Required	New Row
Page 1	Page				
Fiscal Month	Column value	Fiscal Month			□
Expense Code Descr	Column value	Expense Code Descr			□
Division	Column value	Division			□
Work Type	Column value	Work Type			☑
Application	Column value	Application			□
Lower Limit	Variable value				☑
Upper Limit	Variable value				□

Display

Page 1

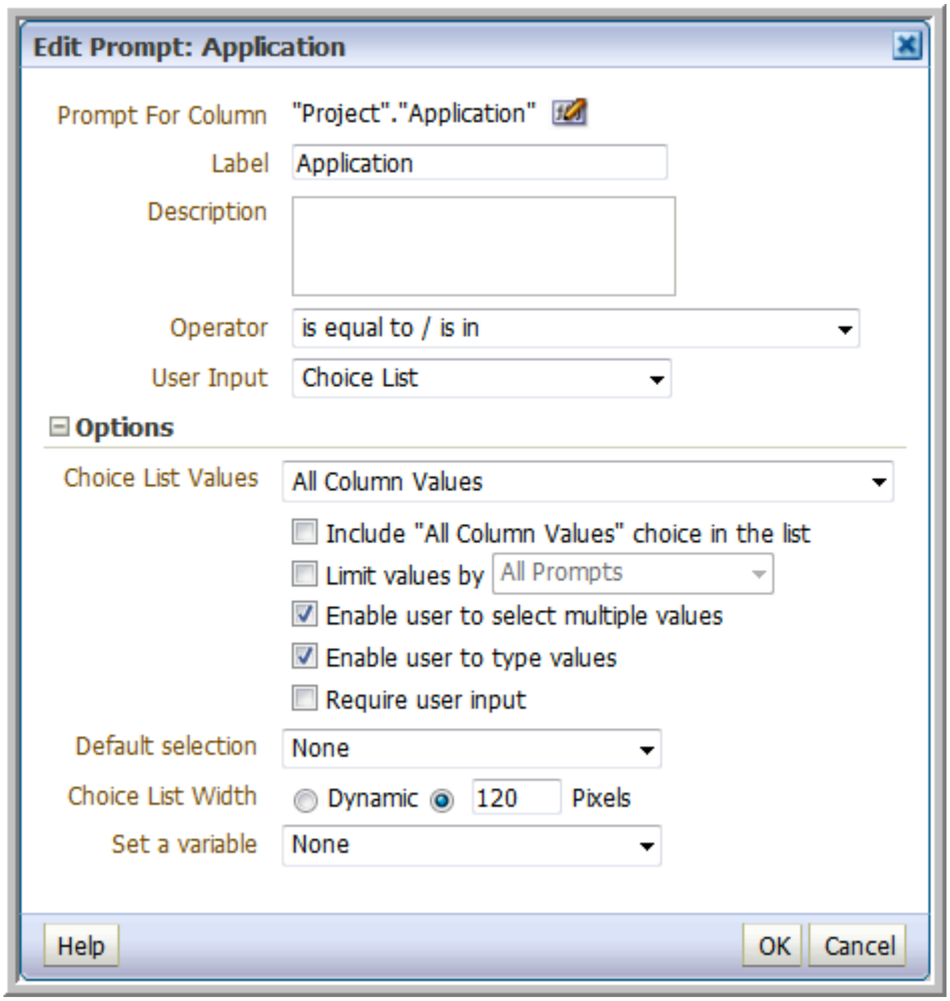
Fiscal Month --Select Value-- Expense Code Descr --Select Value-- Division --Select Value--

Work Type --Select Value-- Application --Select Value--

Lower Limit Upper Limit

By default, all prompts would be arranged on a single row. Notice that a new row is started with the Work Type prompt and with the Lower Limit prompt, because of the ticked checkboxes.

Let’s examine the options available to us on the Edit Prompt dialog.



Edit Prompt: Application

Prompt For Column "Project"."Application"

Label Application

Description

Operator is equal to / is in

User Input Choice List

Options

Choice List Values All Column Values

□ Include "All Column Values" choice in the list

□ Limit values by All Prompts

☑ Enable user to select multiple values

☑ Enable user to type values

□ Require user input

Default selection None

Choice List Width ○ Dynamic ● 120 Pixels

Set a variable None

Label: The text shown above the prompt box.

Operator: There are many different operators to choose from. While **is equal to / is in** is the most frequently used, some of the other commonly used values include:

- is not equal to / is not in;
- is greater than
- is less than
- is between
- contains

User Input: The type of selection mechanism. There are five options available:

- Choice List: A simple dropdown list of values to choose from.
- Text Field: The user types a selection into the field
- Check Boxes: The user can select one or more values from a list of check boxes.
- Radio Buttons: The user can select one value from a radio button list.
- List Box: Similar to the Choice List, except that the available values are shown on the left side of a selection dialog, and the selected values are shown on the right.

Values: Which values are displayed to the user? Some of the options here include:

- All Column Values – display all values of the specified column
- Specific Column Values – display only specifically named values of the column
- SQL Results – display only those values returned as the result of a logical SQL statement. For example, the SQL statement
  select "Org"."Work Type" from Training where "Effort"."Corrected Hours" > 500
  would return a limited set of Work Types for display in the prompt.

Include "All Column Values" choice in the list
Not only would the list include the values specified in the Values area, it would also have an additional **All Choices** value at the top of the list, allowing the user to quickly select all values.

Limit values by
This option allows us to display only relevant values of some prompts based on other prompts. For example, we might want to select values of Work Type, then only see values in the Application prompt that are related to those Work Types. If so, we would tick the **Limit values by** checkbox for the Application prompt, and specify that its values should be limited by the user's selections from the Work Type column prompt.

Enable user to select multiple values
If only one value may be selected from a prompt, this checkbox would not be ticked.

Enable user to type values
This option allows users to type values for selection. This is most effectively implemented with columns containing short values, such as Fiscal Months. Since searches and matches are case sensitive, allowing this option for longer values might not be very productive.

Require User Input
When this option is selected, users cannot click the Apply button to execute the prompt query until they have selected a prompt value. An asterisk displays to the left of the prompt label to indicate that the prompt is required.

Default selection: There are five possible options for the **Default to** condition:

- None: No default is specified.
- Specific Value: A specified, hard-coded value.
- All Column Values: This option is only available when the **Include "All Choices" choice in the list** option is selected. Select this option to specify the "All Choices" default value for the prompt.
- Server Variable: The value of a Repository Variable or Session Variable will be the default. For a session variable, prefix the name with NQ_SESSION, such as NQ_SESSION.USER.
- SQL Results: The results of a SQL statement.
- Variable Expression: references to reserved (system) session variables, including:
    - @{system.currentTime}
    - @{system.productVersion}
    - @{session.locale}
    - @{session.language}
    - @{session.loginTime}
    - @{session.logoutTime}
    - @{session.lastAccessTime}
    - @{session.currentUser.id}
    - @{user.homeDirectory}
    - @{user.id}
    - @{user.displayName}

Selection Value Width:

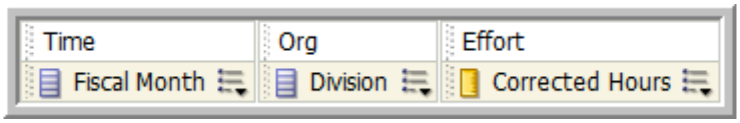
- Dynamic: Let OBIEE determine the width of the check list, radio buttons, etc… based on the widths of their contents.
- # Pixels: Display the list of values at a fixed number of pixels, concatenated on the left.

Set a Variable: The two possible values are **Presentation Variable** and **Request Variable**

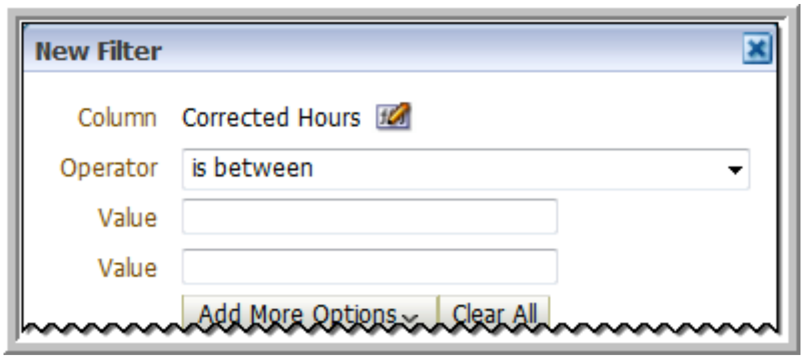
- **Presentation Variable:** A variable that is created by the dashboard prompt
- **Request Variable:** The name of a Session Variable.

Turn to the next page to continue with the dashboard prompt/analysis exercise.

13. Create a new analysis with the Fiscal Month, Division, and Corrected Hours columns:



14. Begin a filter for the Corrected Hours column, using the **Is Between** operator


New Filter

Column: Corrected Hours

Operator: is between

Value:

Value:

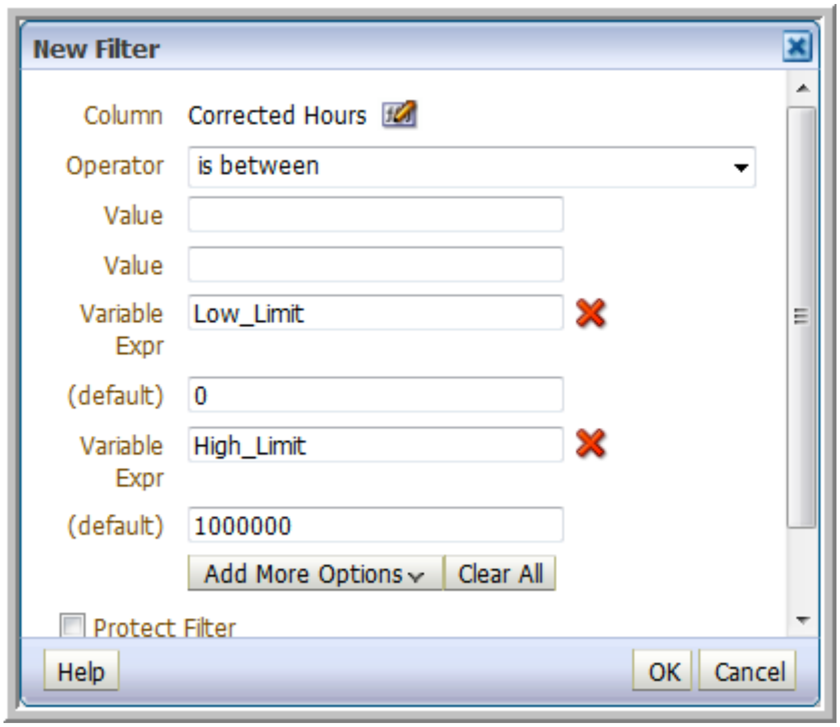
15. As you learned in an earlier lesson in which you used Repository Variables, entry boxes on the filter dialog that are labeled **Value** will use the literal character strings entered there. Since we want to use our two Presentation Variables as the lower and upper bounds of our filter, click in the first **Value** box and click the **Add More Options** button.

16. Select **Presentation Varible.**

17. In the field labeled **Variable Expr**, type type **Low_Limit.**

18. In the field labeled **(default)**, type **0** (zero).

19. Repeat the previous 4 steps for the second Value field, using the High_Limit presentation variable with a default of 1,000,000.



New Filter

Column: Corrected Hours

Operator: is between

Value:

Value:

Variable Expr: Low_Limit

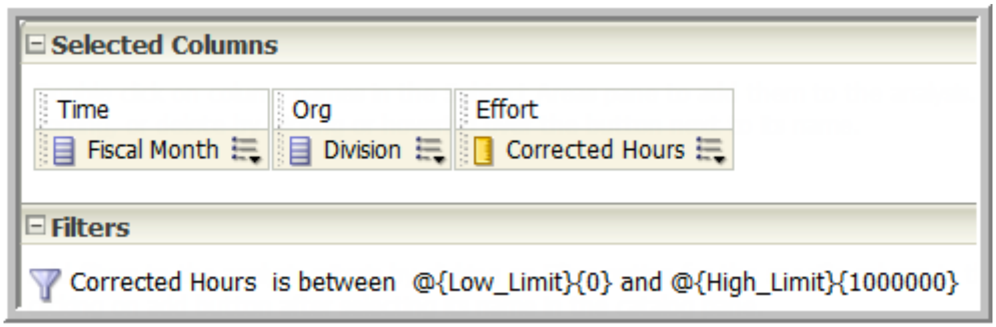
(default): 0

Variable Expr: High_Limit

(default): 1000000

20. Click OK to complete the definition of the new filter.

21. The finished Critieria tab will look like this:



Selected Columns

Time	Org	Effort
Fiscal Month	Division	Corrected Hours

Filters

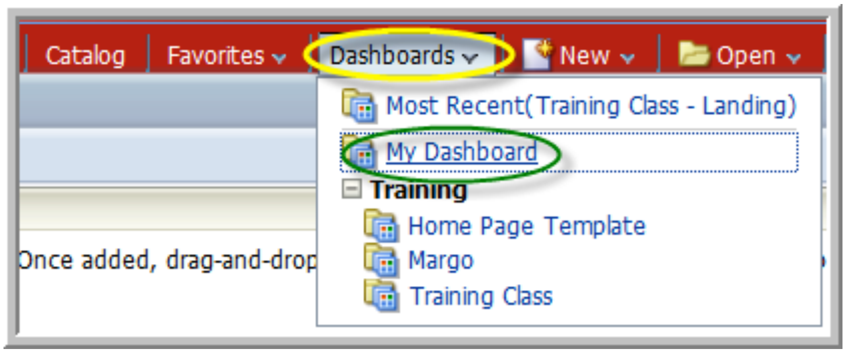
Corrected Hours is between @{Low_Limit}{0} and @{High_Limit}{1000000}

22. Save the analysis in **My Folders** as **Ranged Results.**

Now let’s put the Dashboard Prompt and the Answers analysis together on a dashboard page and see what happens.

★	***CAUTION: In the real world, dashboard development never begins in My Folders.***

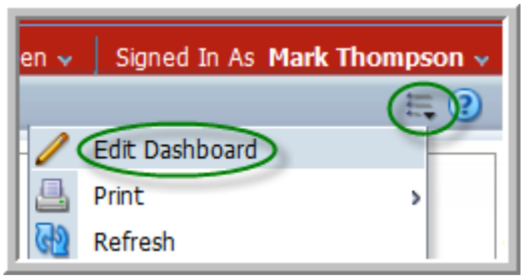
23. Click on the **Dashboards** link in the toolbar, and select My Dashboard.



Note: **My Dashboard** is your own personal dashboard space, to be used for whatever purpose you desire. Nobody else can see or alter your **My Dashboard**. You cannot explicitly share access to your **My Dashboard**. It is exclusively assigned to you, in your personal workspace.

The list of shared dashboards that you see below the My Dashboard link may vary, depending on the access rights granted to you by the administrators of the system based on your needs.

24. At the top right of the page, underneath your sign-in name, click the **Page Options** button, then **Edit Dashboard.**



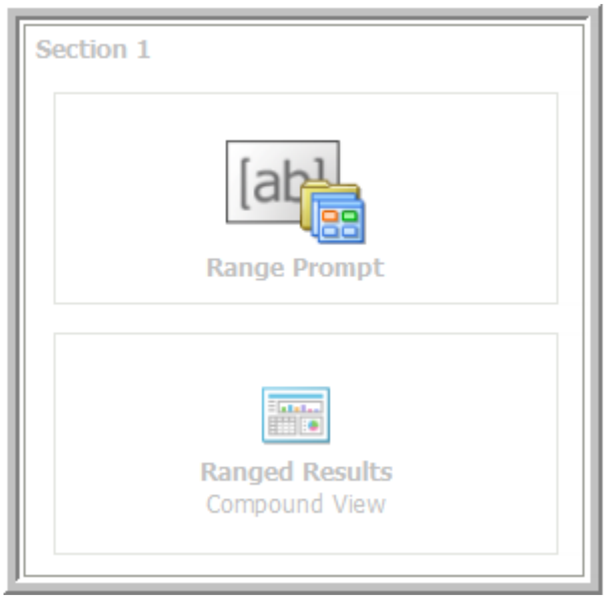
25. From the toolbar above the empty design canvas, click the **Add Dashboard Page** icon, and enter **Range** as the page name. Click **OK**.

26. On the left side of the screen, drill down into **My Folders**.

27. Drill further into the **Prompts** folder.

28. Grab and drag the Range Prompt object to the right, dropping anywhere on the empty design canvas.

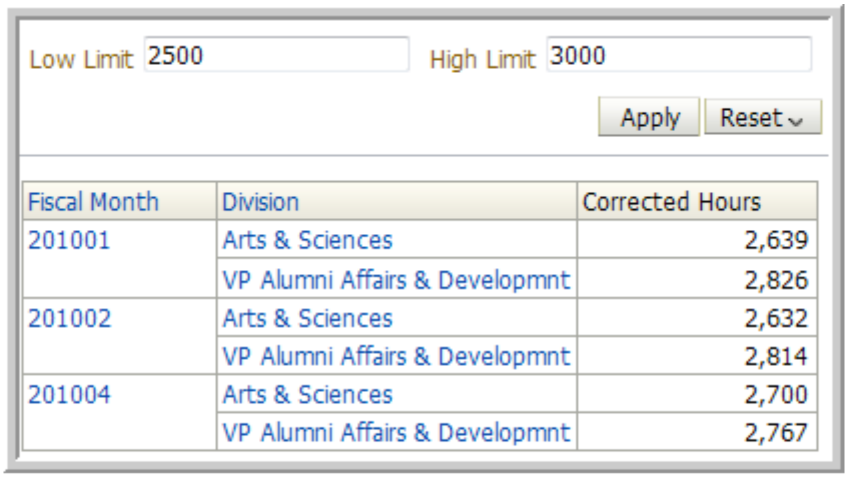
29. Further down in the My Folders list, grab and drag the **Ranged Results** object, dropping it directly below the Range Prompt placeholder. Make sure to drop Ranged Results **inside** Section 1 by placing the mouse directly onto the bottom border line of Section 1, and release the mouse button when you see the border around Section 1 turn yellow.



30. Click the **Save** button to save the dashboard page.

31. Click the ▷ Run button to view the results.

32. Test the interaction between the analysis and the dashboard prompt by entering values into the Low Limit and High Limit fields, and clicking the Apply button.



Low Limit 2500 High Limit 3000

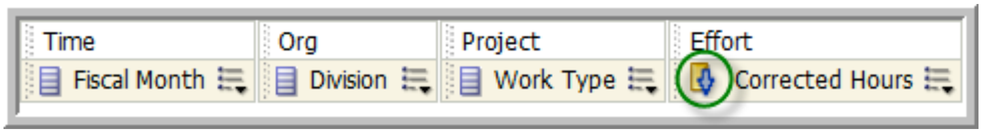
Fiscal Month	Division	Corrected Hours
201001	Arts & Sciences	2,639
	VP Alumni Affairs & Developmnt	2,826
201002	Arts & Sciences	2,632
	VP Alumni Affairs & Developmnt	2,814
201004	Arts & Sciences	2,700
	VP Alumni Affairs & Developmnt	2,767

Exercise 7c: Filtering for a combined X% of a group

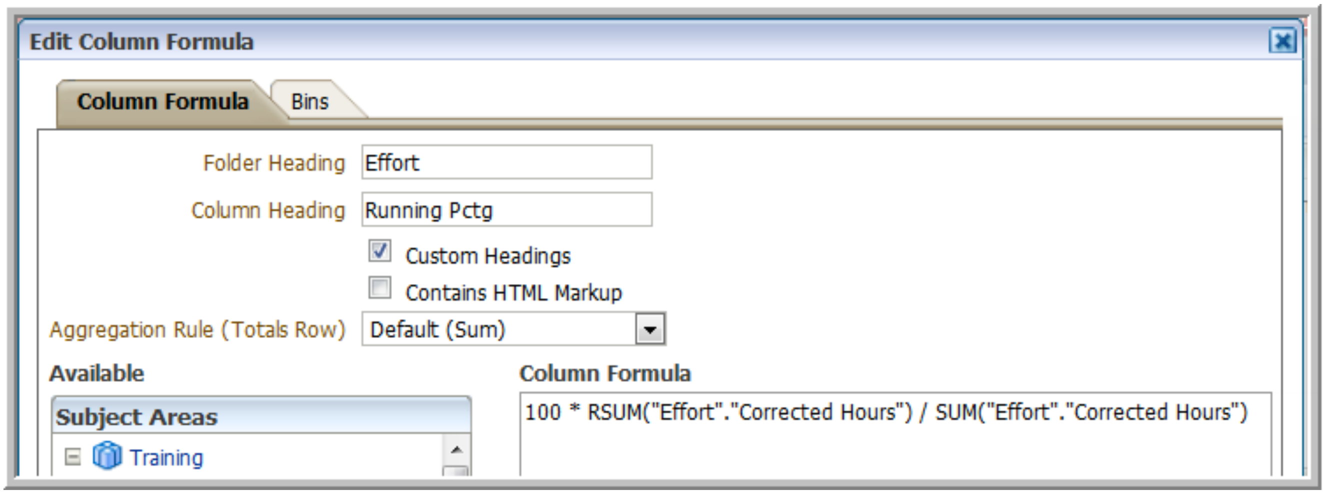
In this exercise, we use a filter to help us display only those rows which comprise the top X% of Corrected Hours, where X is input into a dashboard prompt by the user.

Let's start with a new Answers analysis.

1. Create this new Answers analysis with the Fiscal Month, Division, Work Type, and Corrected Hours columns, sorting by Corrected Hours in descending order as shown.



2. Add any column to the analysis, such as Corrected Hours again, then change its formula and Customize its Column Heading as shown below. Refer to page **Error! Bookmark not defined.** for a refresher on modifying and creating column formulas.



Folder Heading: Effort

Column Heading: Running Pctg

☑ Custom Headings

□ Contains HTML Markup

Aggregation Rule (Totals Row): Default (Sum)

Column Formula

```
100 * RSUM("Effort"."Corrected Hours") / SUM("Effort"."Corrected Hours")
```

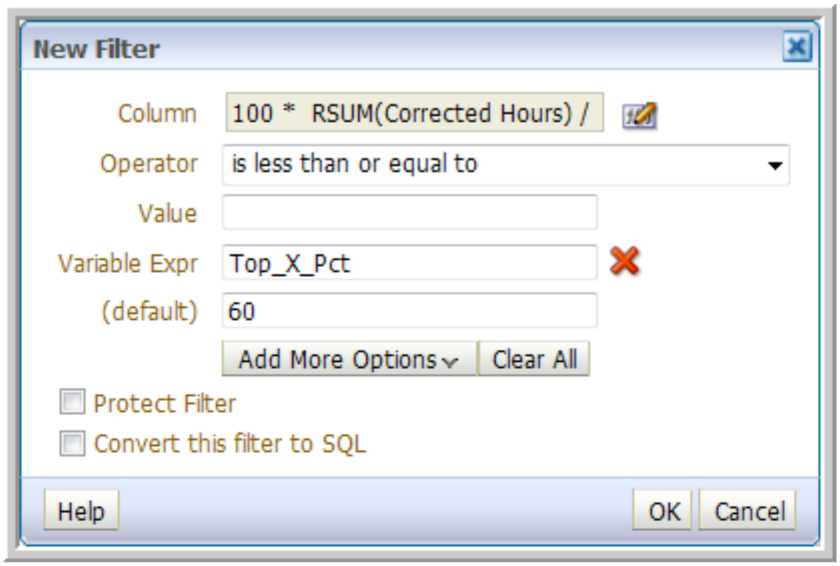
The RSUM (running sum) function is found under **Running Aggregates**, and the SUM function is found under **Aggregates** (or you may just type them if you wish). The formula for **Running Pctg** is:

100 * RSUM(Effort.Corrected Hours) / SUM(Effort.Corrected Hours)

Running Pctg is a running total of the percentage of total Corrected Hours encountered row-by-row in the Answers analysis.

3. Modify the **Running Pctg** column properties to display with **1 decimal** and a **percent sign**.

4. Create a filter for the **Running Pctg** column as shown here.



New Filter

Column: 100 * RSUM(Corrected Hours) /

Operator: is less than or equal to

Value:

Variable Expr: Top_X_Pct

(default): 60

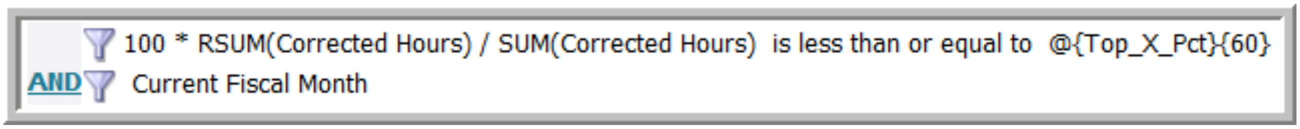
□ Protect Filter

□ Convert this filter to SQL

Top_X_Pct is the name of a Presentation Variable that we'll create shortly. Set a default, such as 60, which will mean to display the top 60% of the results. Note that this setting only applies for testing in Answers. We will also specify a default value on the dashboard prompt.

Helpful Hint: Notice that since Running Pctg is a custom formula, the ***formula*** for the column, and not column name, is what will be saved as the filter condition. Should the formula for the **Running Pctg** column change in the future, ***this filter will not reflect that change.***

5. Apply the previously saved **Current Fiscal Month** filter to the analysis. The filters should look like this:



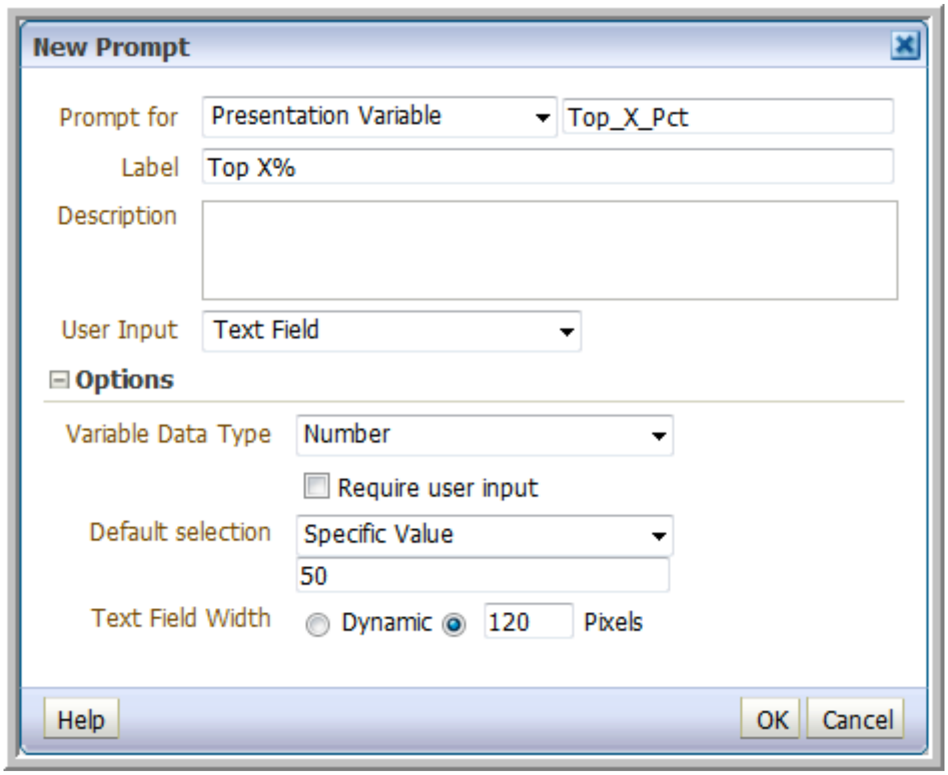
100 * RSUM(Corrected Hours) / SUM(Corrected Hours) is less than or equal to @{Top_X_Pct}{60}

AND Current Fiscal Month

6. Save the analysis as **Top X% Cumulative.**

So far, we've created an analysis which looks for the presence of a Presentation Variable called Top_X_Pct. Remembering that Presentation Variables are created in Dashboard Prompts, our next step will be to create that Dashboard Prompt and its associated Presentation Variable, so that our users can enter any percentage of results that they would like to see on the dashboard.

7. Create a new dashboard prompt as shown here.



New Prompt

Prompt for: Presentation Variable ▾ Top_X_Pct

Label: Top X%

Description:

User Input: Text Field ▾

⊟ Options

Variable Data Type: Number ▾

☐ Require user input

Default selection: Specific Value ▾

50

Text Field Width: ○ Dynamic ● 120 Pixels

Note that we're specifying a default of 50 as the value of the Top_X_Pct presentation variable. The default specified in the analysis filter was 60? Which of the two values do you expect will be used as the default on the dashboard?

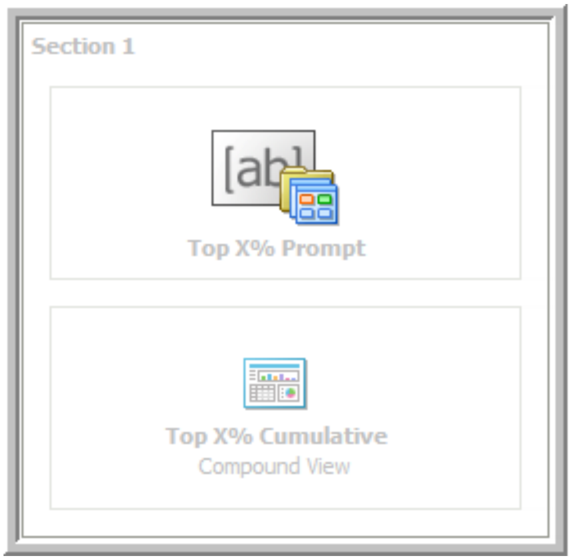
Also note that presentation variable names are case sensitive.

8. Save the dashboard prompt under **My Folders \ Prompts** as **Top X% Prompt**.

Now let's put the Dashboard Prompt and the report together on a dashboard page.

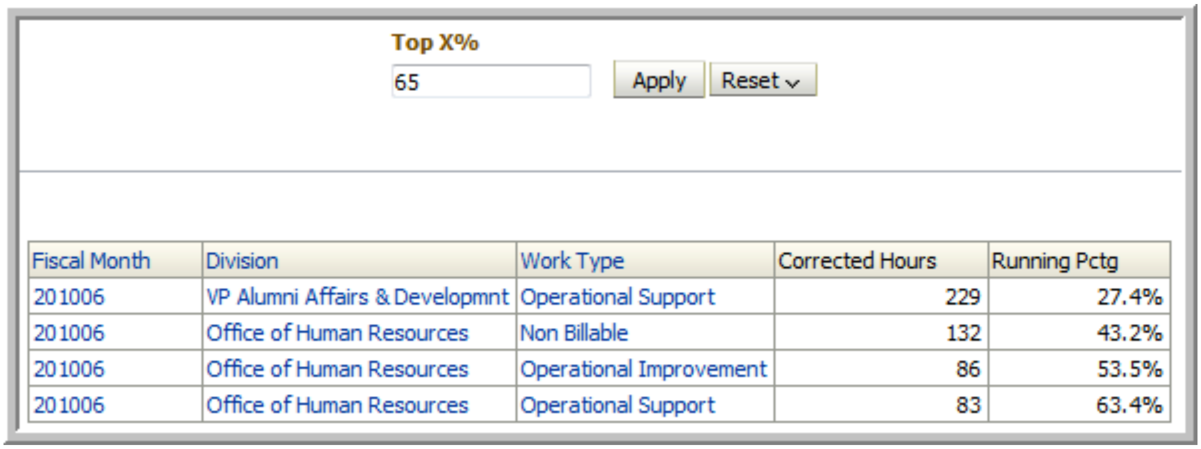
9. Click on the **Dashboards** link, choose My Dashboard, choose **Page Options … Edit Dashboard**, and add a new page called **Top X%.**

10. Drag **Top X% Prompt** and **Top X% Cumulative** from the selection panel, and drop them onto the new dashboard page, as shown here:



11. Save and run the dashboard. Notice that the dashboard prompt default of 50 is initially shown.

12. Enter a value such as **65** into the **Top X% field**, and click **Apply.** Your report should only show those combinations of Division and Work Type which make up less than the top 65% of the total of all Corrected Hours for the current Fiscal Month.



Top X%

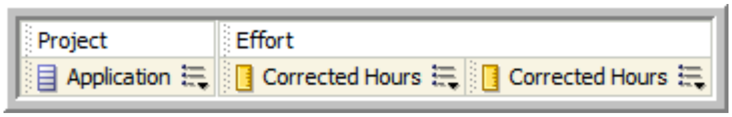
65

Fiscal Month	Division	Work Type	Corrected Hours	Running Pctg
201006	VP Alumni Affairs & Developmnt	Operational Support	229	27.4%
201006	Office of Human Resources	Non Billable	132	43.2%
201006	Office of Human Resources	Operational Improvement	86	53.5%
201006	Office of Human Resources	Operational Support	83	63.4%

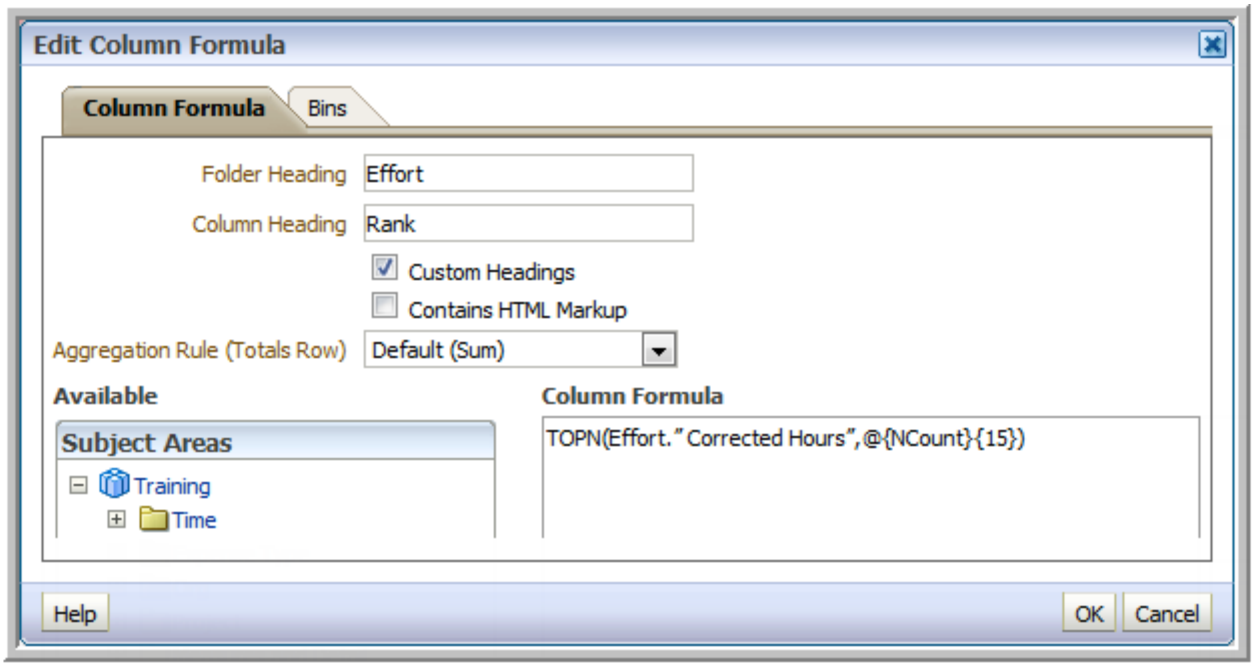
Exercise 7d: Filtering with TopN / BottomN functions

The TopN and BottomN functions are actually filters, not functions. When you include a TopN or BottomN function in a column formula, the analysis is automatically filtered to only return the number of rows specified in that function.

1. Create this Answers analysis with the Application column once, and the Corrected Hours column twice.



2. Modify the formula for the 2nd Corrected Hours column so that its column heading is **Rank**, and its formula is TOPN(“Effort”.”Corrected Hours”,@{NCount}{15}) as shown below.



Folder Heading: Effort

Column Heading: Rank

☑ Custom Headings

□ Contains HTML Markup

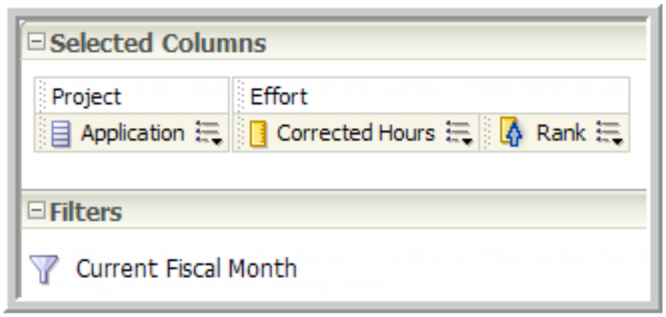
Aggregation Rule (Totals Row): Default (Sum)

Column Formula: TOPN(Effort.”Corrected Hours”,@{NCount}{15})

The TOPN function is found under **Aggregates.** The @{NCount}{15} shows the use of a Presentation Variable called NCount (with a default value of 15) that will be created in a dashboard prompt. You may type it in as shown, or you may use the **Variable** button to insert it in the formula. **Reminder: Presentation Variable names are case sensitive**.

3. Sort the results in ascending order based on the Rank column.

4. Apply the Current Fiscal Month filter to the analysis. The criteria should look like this:



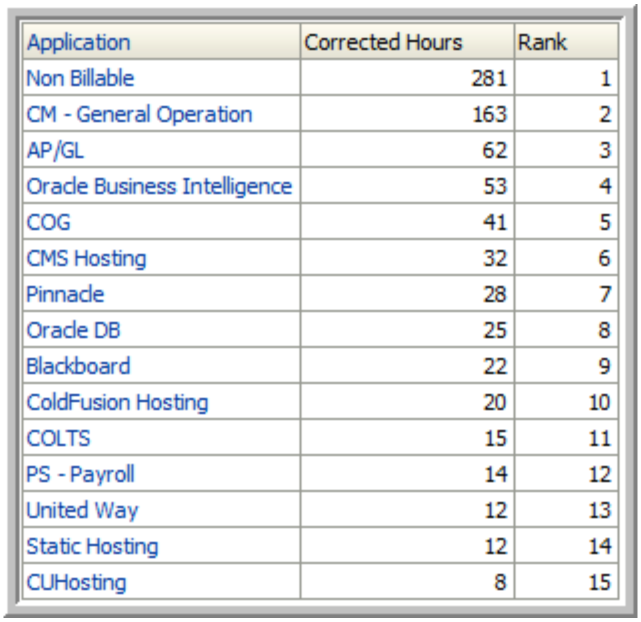
Selected Columns

Project	Effort	
Application	Corrected Hours	Rank

Filters

Current Fiscal Month

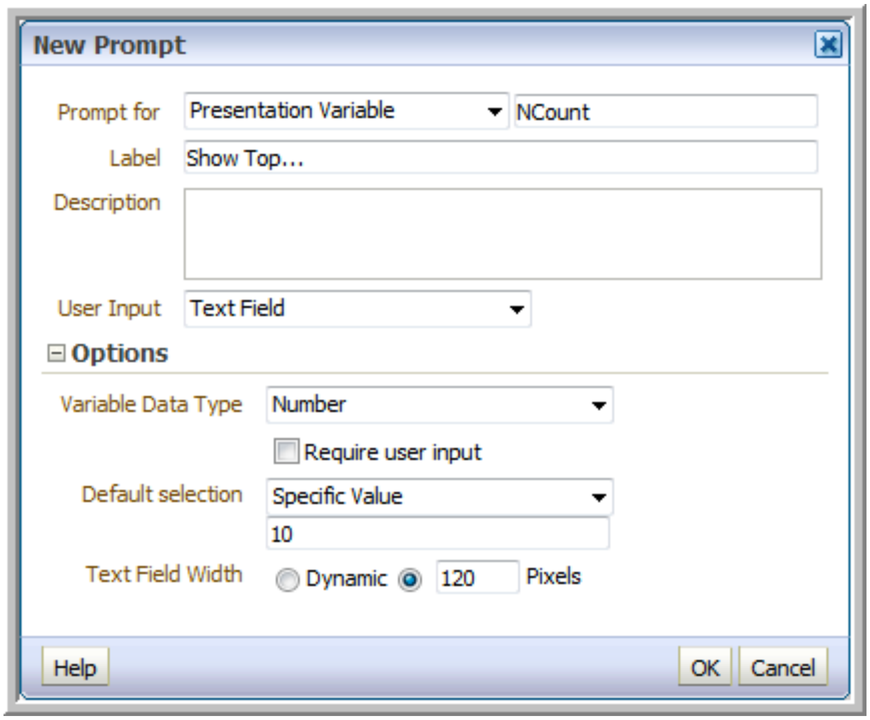
5. View the results.



Application	Corrected Hours	Rank
Non Billable	281	1
CM - General Operation	163	2
AP/GL	62	3
Oracle Business Intelligence	53	4
COG	41	5
CMS Hosting	32	6
Pinnacle	28	7
Oracle DB	25	8
Blackboard	22	9
ColdFusion Hosting	20	10
COLTS	15	11
PS - Payroll	14	12
United Way	12	13
Static Hosting	12	14
CUHosting	8	15

6. Save the analysis as **Top N.**

7. Create a new dashboard prompt as shown here, and save it as **Top N Prompt**. Make sure to set a **Default selection** value such as 10.



New Prompt

Prompt for: Presentation Variable — NCount

Label: Show Top...

Description:

User Input: Text Field

⊟ Options

Variable Data Type: Number

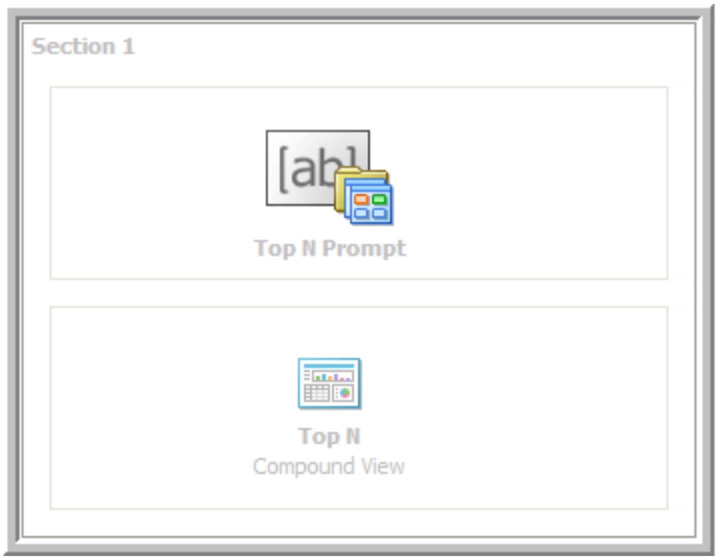
□ Require user input

Default selection: Specific Value

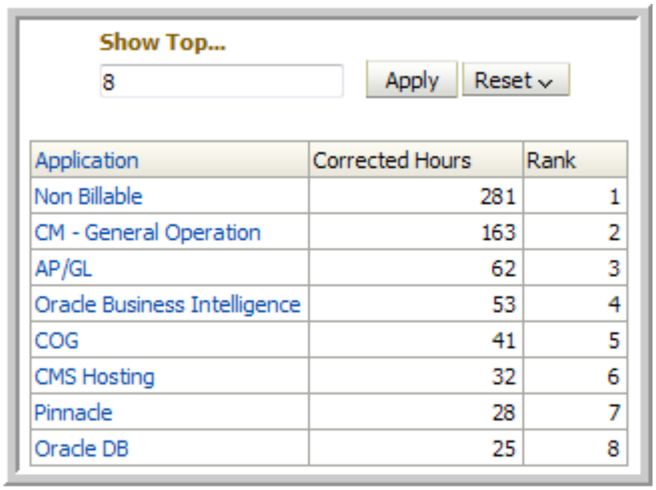
10

Text Field Width: ○ Dynamic ● 120 Pixels

8. Edit My Dashboard, add a new page called **Top N**, and add the **Top N Prompt** and **Top N** analysis to the page.



9. Save the dashboard and view the results. Enter a value into the dashboard prompt and click Apply to show the effect of the TopN function.



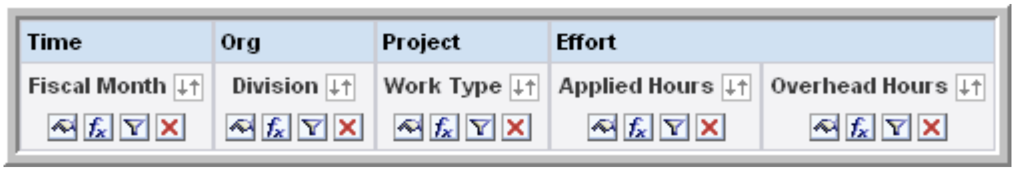
Application	Corrected Hours	Rank
Non Billable	281	1
CM - General Operation	163	2
AP/GL	62	3
Oracle Business Intelligence	53	4
COG	41	5
CMS Hosting	32	6
Pinnacle	28	7
Oracle DB	25	8

Note: In any given analysis, there can only be one TopN or BottomN function. That's one **total**, not one of each.

Exercise 7e: Configuring for Required Selections on Dashboard Prompts

In the absence of any filtering criteria or default dashboard prompt values, it is possible that a significant number of rows could be returned unnecessarily to a dashboard screen prior to the use of the dashboard prompt. In this exercise, we'll learn how avoid that by restricting the execution of analyses on a dashboard page until after the dashboard prompt(s) have been used.

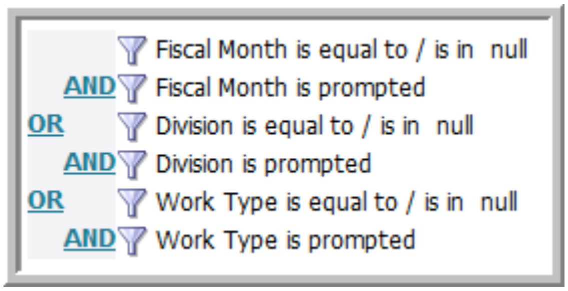
1. Create this simple unfiltered analysis in Answers, and save it as **Required Prompt Analysis**.



Time	Org	Project	Effort	
Fiscal Month	Division	Work Type	Applied Hours	Overhead Hours

2. Remembering the techniques for grouping filters as learned in a previous lesson (refer to page **Error! Bookmark not defined.**), create filters for the three dimension columns as shown below.

   Note: the first filter in each pair uses the **is equal to / is in** operator, and the value to compare is the four characters **n-u-l-l.** This is not an IS NULL operator. The comparison string can be any value that you know does NOT exist in the column, and **null** is usually a pretty good one. This will force the filter to return no results.



```
    Fiscal Month is equal to / is in  null
    AND Fiscal Month is prompted
OR      Division is equal to / is in  null
    AND Division is prompted
OR      Work Type is equal to / is in  null
    AND Work Type is prompted
```

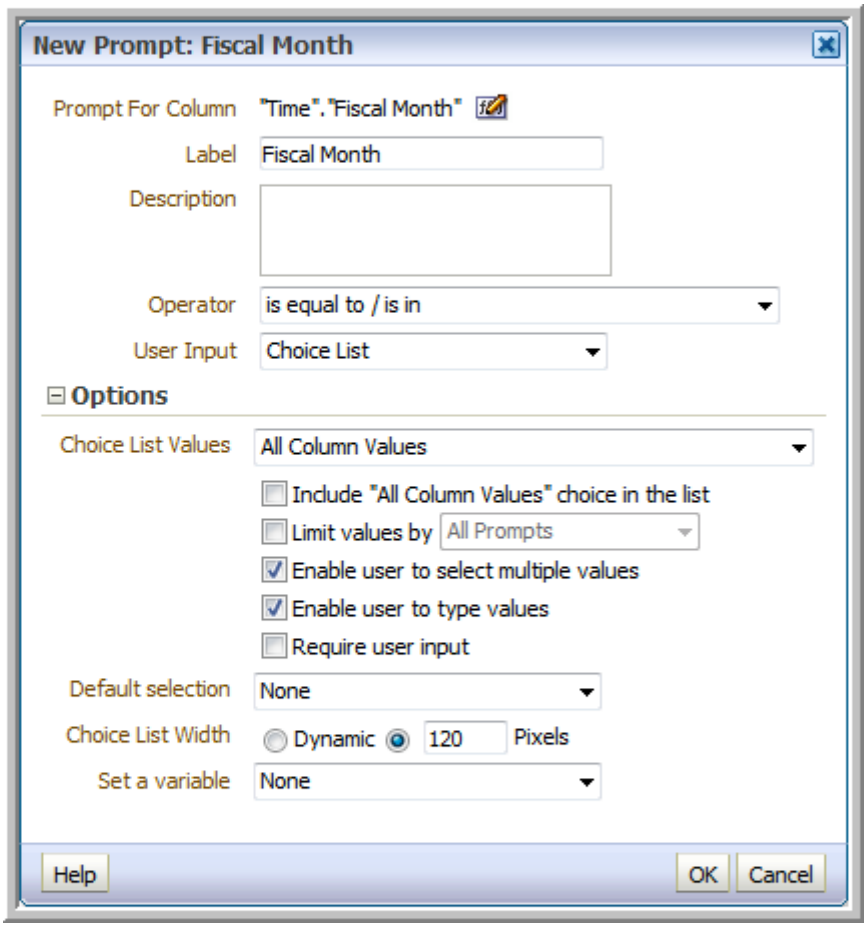
We read the first filter group like this: "I want to see Fiscal Month values that match the character string 'n-u-l-l', plus all values that match what the user has selected from a dashboard prompt." If the user hasn't selected anything in the prompt, and since we know that there isn't a value of Fiscal Month spelled **n-u-l-l,** then that first filter won't return any hits, and the analysis will return zero rows.

And of course, the other two prompt sets work exactly the same way. So, until the user has made a selection from all three prompts, the analysis will not return any rows.

One caveat for YOU as you're developing in Answers: You'll want to click that first AND to temporarily change it to an OR as you're working with it, (or apply some other filter), otherwise you won't see any results in Answers.

3. Resave the analysis.

4. Create and a new dashboard prompt, starting with a Column Prompt for the Fiscal Month column as shown here:



New Prompt: Fiscal Month

Prompt For Column: "Time"."Fiscal Month"

Label: Fiscal Month

Description:

Operator: is equal to / is in

User Input: Choice List

⊟ Options

Choice List Values: All Column Values

□ Include "All Column Values" choice in the list

□ Limit values by All Prompts

☑ Enable user to select multiple values

☑ Enable user to type values

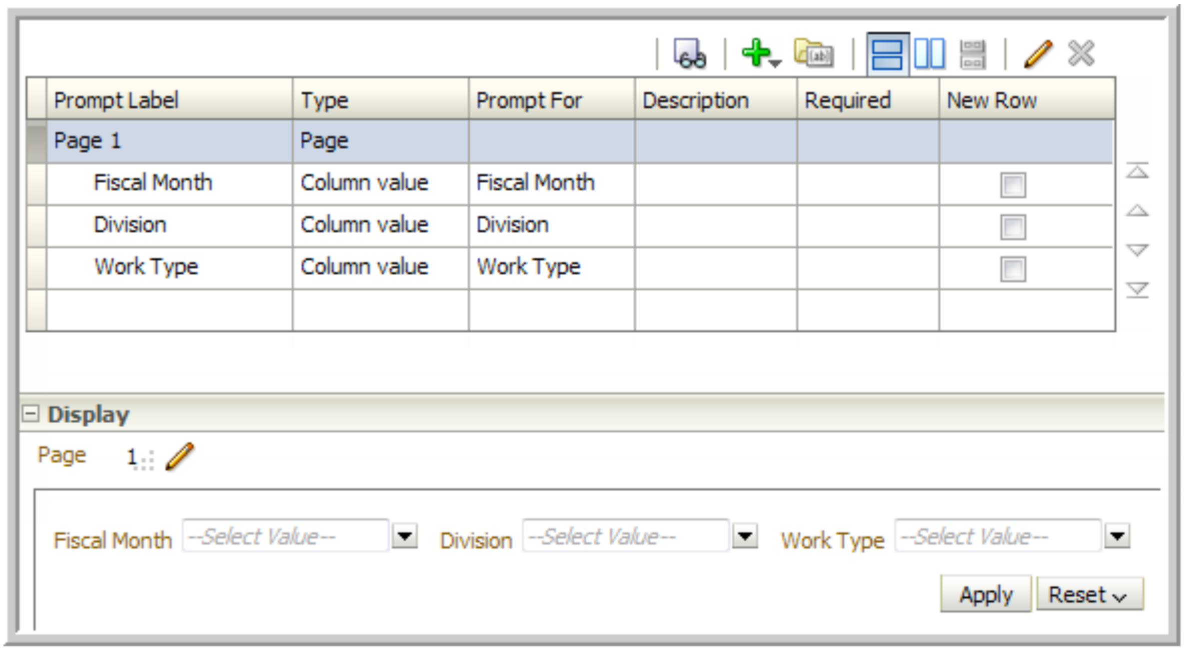
□ Require user input

Default selection: None

Choice List Width: ○ Dynamic ● 120 Pixels

Set a variable: None

5. Create two other prompt values in this dashboard prompt, for the Division and Work Type columns, but otherwise identical to the Fiscal Month prompt shown above.



Prompt Label	Type	Prompt For	Description	Required	New Row
Page 1	Page				
Fiscal Month	Column value	Fiscal Month			□
Division	Column value	Division			□
Work Type	Column value	Work Type			□

Display

Page 1

Fiscal Month --Select Value-- Division --Select Value-- Work Type --Select Value--

6. Save the dashboard prompt as **Required Prompts**.

7. Edit My Dashboard, create a new page named **Required**, and mate the new dashboard prompt with the new analysis.

8. Notice that the analysis initially returns no information (which is what we intended), and that the default No Results message is being displayed. If you wish to do so, you may return to Answers, and add a totally blank No Results message to the analysis, using the techniques learned in an earlier lesson.

9. Make selections from each of the three prompts and click Apply to display the results.

Exercise 7f: Drilling and Navigation

The column values and column headings in an Answers analysis may, at the option of whomever is creating and/or editing the Answers analysis, be defined as **drill links** or **navigation links**.

Definitions:

Drill Link – A drill link displays the next lowest hierarchical level of information. When a user clicks on a column value and/or a column heading, Answers will "drill" down to the next lower level in the hierarchy associated with the source column. Values at the lowest level of the dimension hierarchy cannot be displayed as drill links.

Navigation Link – A navigation link redirects the user to a specified dashboard page or to a specified Answers analysis. When a user clicks on a column value and/or a column heading, Answers will open a specified target Answers analysis or navigate to a specified dashboard page.

Heading Link - The column header is presented as a link.

- When used as a drill link, the "child" column is added to the analysis, and all existing values of the source column are retained (i.e. no filter is applied).
- When used as a navigation link, clicking the heading link will open a specified Answers analysis or navigate to a specified dashboard page.

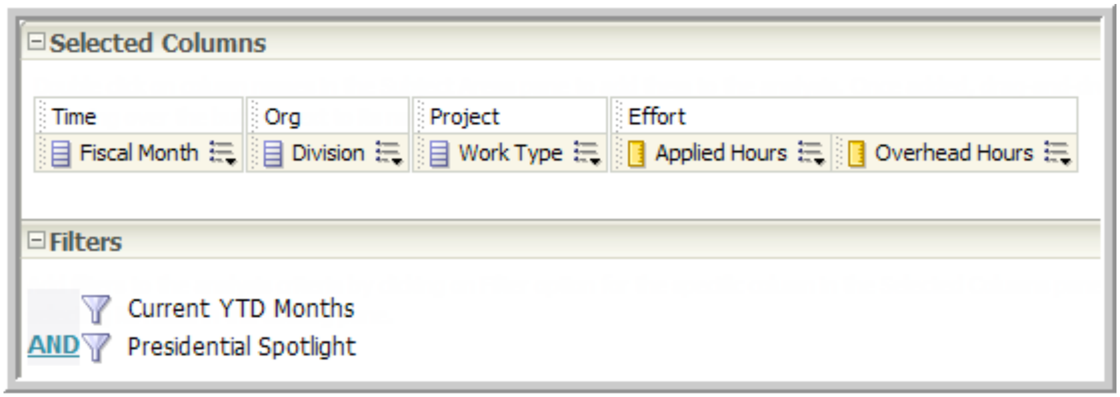
Value Link – The data values in a column are presented as links.

- When used as a Drill Link, the "child" column is added to the analysis, and a filter condition is added to the analysis, limiting the "parent" column to only the value clicked. For example, clicking on a value like Q1-2002 in the Quarter column will add the Month column to the analysis, will add a filter like "Quarter = Q1-2002", and will display only that quarter and the months related to it.
- When used as a navigation link, Answers will open a specified Answers analysis or navigate to a specified dashboard page.

> **Helpful Hint**: When drilling and/or navigating while working **in Answers**, it is usually desirable to click the dashboard preview icon before clicking a drill link or navigation link. Choosing to **drill** without going to the preview screen will **add** new columns and filters to the Answers analysis, which might not be the desired action. Choosing to **navigate** without going to the preview screen shouldn't cause any harm, as it should open the target analysis in a new browser window or tab, but it's a good habit to save often.

Let's take a look at the default drilling / navigation behaviors of an Answers analysis.

1. Create this simple analysis in Answers, and save it as **Drill and Navigate**.



Selected Columns

Time	Org	Project	Effort	
Fiscal Month	Division	Work Type	Applied Hours	Overhead Hours

Filters

Current YTD Months
AND Presidential Spotlight

2. View the table on the Results tab.

3. Click on the Dashboard Preview icon.

4. Click on the Division column heading link. Notice that the three existing Divisions are retained, and their associated "child" Departments are now displayed in a column immediately to right of the Division column. The data values represent data at the Department level.

5. Click the browser's Back button to "undrill".

6. Next, click on the **Operational Improvement** link next to the **Graduate School** Division. Notice that the resulting analysis is now filtered to include only those column values associated with the clicked value, **including the specific values in columns to the left** of the clicked value.

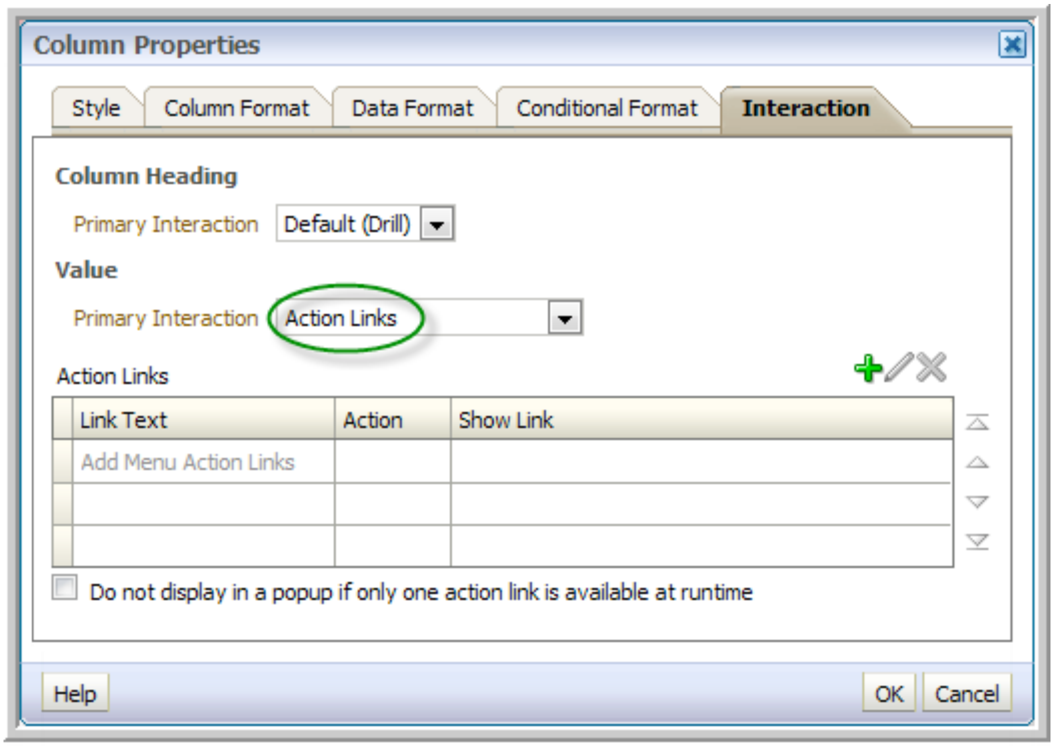
7. Close the dashboard preview browser window.

So we see that clicking to drill on a **column heading** will retain all values in all columns, and just add a new "child" column to the analysis (i.e. will not add filters), while clicking to drill on a **value** within a column will add a new "child"column **and** create filters for all columns to the left of the clicked cell.

Now let's take control over what happens when the user clicks on a column heading or a value.

8. Return to the Criteria tab, and open the Column Properties for the Division column.

9. On the **Interaction** tab, click the **Value Interaction** dropdown, and select **Action Links**. (We'll leave the Column Heading Interaction at its Default setting, which is Drill.)



Column Heading

Primary Interaction: Default (Drill)

Value

Primary Interaction: Action Links

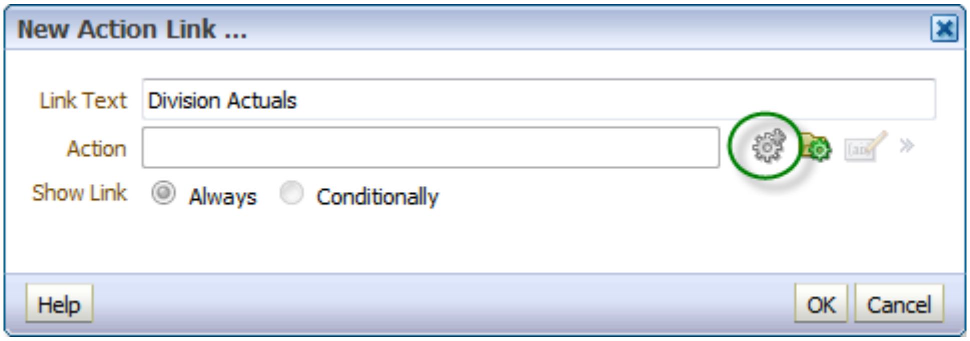
Action Links

Link Text	Action	Show Link
Add Menu Action Links		

☐ Do not display in a popup if only one action link is available at runtime

10. Click the Add Action Link ✚ icon.

11. Begin creating a new Action Link by clicking the icon shown below:



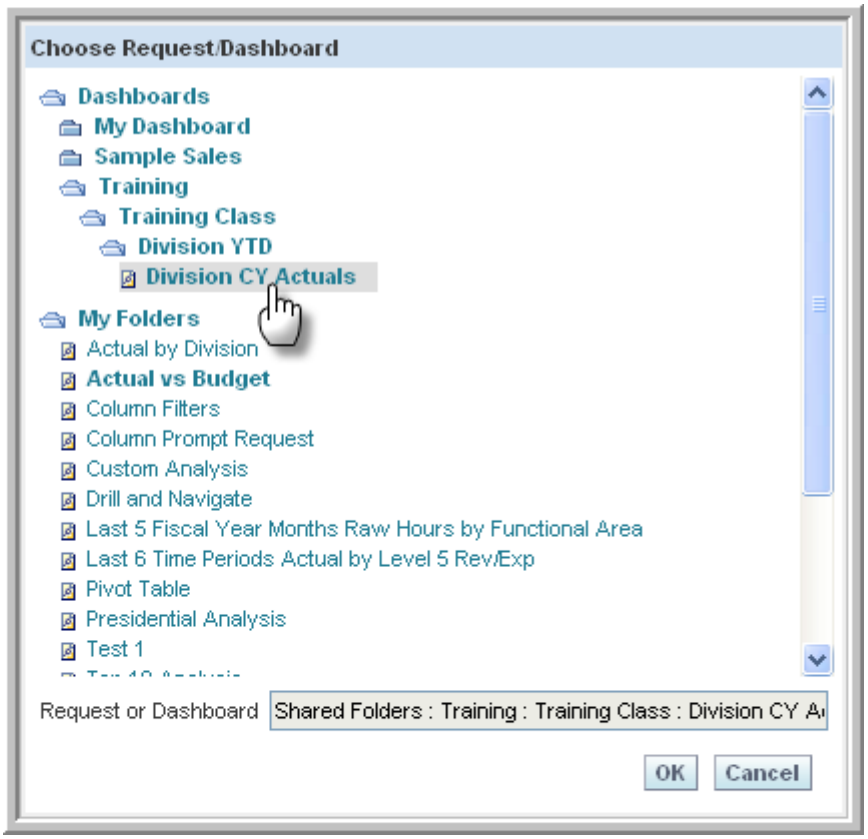
New Action Link ...

Link Text: Division Actuals

Action:

Show Link: ● Always ○ Conditionally

12. Browse to the location shown in the screenshot below, and click OK.



Choose Request/Dashboard

Request or Dashboard: Shared Folders : Training : Training Class : Division CY A

Notes:

c. If an **Answers Analysis** is specified as the destination (as shown here) that Analysis will be opened when the link is clicked. Note that even though we located an analysis by browsing into a dashboard, our means of locating it is irrelevant: we selected an analysis, and only that Analysis will be opened, not the dashboard that contains it.

d. If a **Dashboard Page** is specified as the destination, the Dashboard will be displayed, and the specified Dashboard Page will be selected and visible, when the link is clicked.

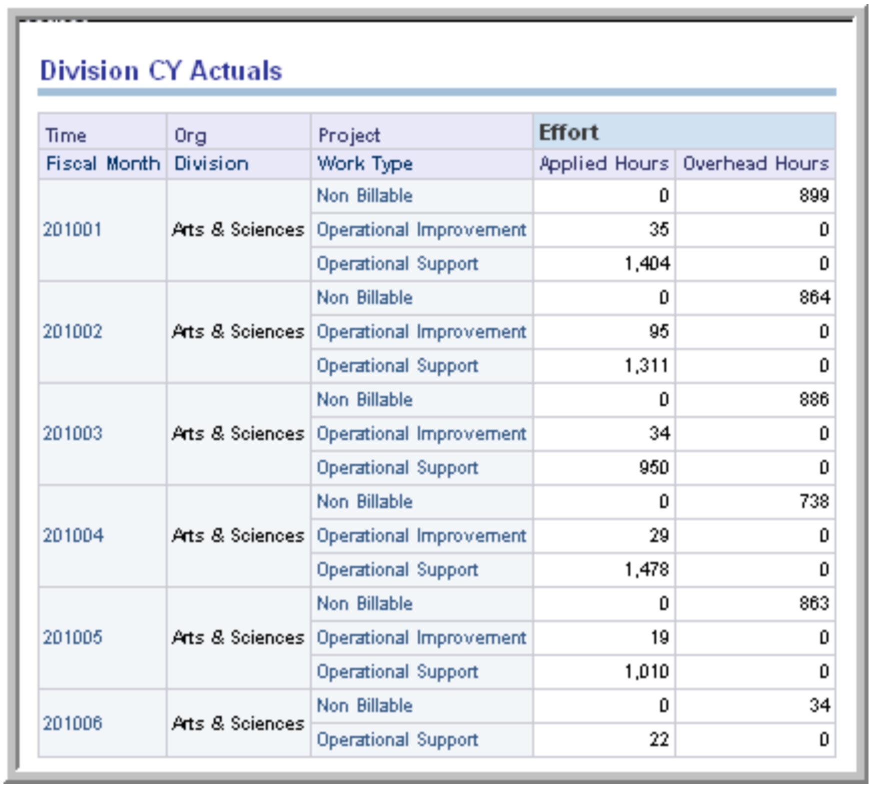
e. If a **Dashboard Name** is specified as the destination, the first page of the Dashboard will be displayed.

13. If you were to choose to add more than one Navigation Target, you would also want to add a value into the Caption box to allow the user to select from multiple destinations.

14. Click OK to close the dialog box

15. Resave the analysis. (My Folders … Drill and Navigate)

16. Let's test our results. Click the Dashboard Preview icon, and click on one of the values in the Division column to see results similar to this:



Division CY Actuals

Time	Org	Project	Effort	
Fiscal Month	Division	Work Type	Applied Hours	Overhead Hours
201001	Arts & Sciences	Non Billable	0	899
		Operational Improvement	35	0
		Operational Support	1,404	0
201002	Arts & Sciences	Non Billable	0	864
		Operational Improvement	95	0
		Operational Support	1,311	0
201003	Arts & Sciences	Non Billable	0	886
		Operational Improvement	34	0
		Operational Support	950	0
201004	Arts & Sciences	Non Billable	0	738
		Operational Improvement	29	0
		Operational Support	1,478	0
201005	Arts & Sciences	Non Billable	0	863
		Operational Improvement	19	0
		Operational Support	1,010	0
201006	Arts & Sciences	Non Billable	0	34
		Operational Support	22	0

17. Notice that the Division column is only displaying the value that we clicked on the source analysis. Reason: this analysis has an **is prompted** filter for the Division column. A navigation link can also serve as a filtering device when the target contains an **is prompted** filter for the column in question.

18. Notice also that the Fiscal Month column is displaying all values. But wait a minute: when we were testing the default drilling behavior, didn't all columns to the left of the clicked value become filtered on just those values? Why is this column behaving differently?

    Remember way back in the Filters lesson where we **protected** the filter containing our YTD Fiscal Months? Now you can see the effect of protecting a filter. If we had not selected the **protected** option for the filter, it would have been ignored in favor of the implied filter resulting from the drilling or navigation action. Without that protection, we would only see one Fiscal Month, whichever one was associated with the row on which we clicked the value of Division.

19. Close the Dashboard Preview.