



Adding Stored Functions to an iDashboards Oracle Data Source

Introduction:

iDashboards only supports Oracle Functions when it comes to Oracle data sources.

It is required for a function to result in at least two values in the form of a REF CURSOR. One output will be used for an x-axis and one output for a y-axis. More outputs can be configured for multiple y-axes.

The return types can be:

- Integer
- Number
- String
- Date/Datetime

Input arguments are supported but not necessarily required. Note that the input argument can be either 'Input' mode or 'Input/Output' mode. If an input argument is necessary for the function to work properly, iDashboards can be forced to supply one. The return types can be:

- Integer
- Number
- String

Date and Datetime data types are not supported as procedure arguments. In order to use 'Date' data types, the Date needs to be passed into the function as a string and converted to a Date/Datetime within the function.

If you're using Oracle as a Data Source, you'll need to add the following to the 'optional driver properties' textbox of your respective Oracle data source in the Admin Application interface:

```
oracle.jdbc.defaultNChar=true
```

Function:

Below are the steps to register and use an Oracle PL/SQL Function as an iDashboards data source:

Run the Following example code against your Oracle Database. You can do this either from the SQL command prompt or from an SQL editor.

```
DROP TABLE data_test;  
  
CREATE TABLE data_test (  
  x NUMBER(5) NOT NULL primary KEY,  
  x_squared NUMBER(10) NOT NULL,  
  pivot NUMBER(5) NOT NULL  
);
```



```
DECLARE
  x INTEGER := 1;
  y INTEGER;
  pivot INTEGER;
BEGIN
  FOR x IN 1..1001 LOOP
    y := x * x;
    pivot := ((x+50) / 100);
    INSERT INTO data_test VALUES(x, y, pivot);
  END LOOP;
END;
/
COMMIT;

CREATE OR REPLACE PACKAGE pkg AS
  TYPE ref_cursor IS REF CURSOR;

  FUNCTION y_equals_x_squared(lo_x IN NUMBER, hi_x IN NUMBER) RETURN
  ref_cursor;

  FUNCTION y_equals_x_squared(lo_x IN NUMBER, hi_x IN NUMBER, dummy IN
  NUMBER) RETURN ref_cursor;

END;
/
show errors

CREATE OR REPLACE PACKAGE BODY pkg AS
  FUNCTION y_equals_x_squared(lo_x IN NUMBER, hi_x IN NUMBER) RETURN
  ref_cursor IS c ref_cursor;
  BEGIN
    OPEN c FOR
      SELECT * FROM data_test
      WHERE (lo_x IS NULL OR x >= lo_x)
      AND (hi_x IS NULL OR x <= hi_x);
    RETURN c;
  END;

  FUNCTION y_equals_x_squared(lo_x IN NUMBER, hi_x IN NUMBER, dummy IN
  NUMBER) RETURN ref_cursor IS c ref_cursor;
  BEGIN
    OPEN c FOR
      SELECT * FROM data_test
      WHERE (lo_x IS NULL OR x >= lo_x)
      AND (hi_x IS NULL OR x <= hi_x);
    RETURN c;
  END;
END PACKAGE pkg;
```

To execute in command prompt:

```
@y_equals_x_squared(x,y);
```



To register this function so it is available in the Data Sources window of the user application:

1. Add the Oracle Date Source in the iDashboards Admin application.
2. Click on the 'Procedures' tab against the Oracle data source and select 'Function' from the dropdown list. Click the 'Add Procedure' button. Refer to Figure1 below.

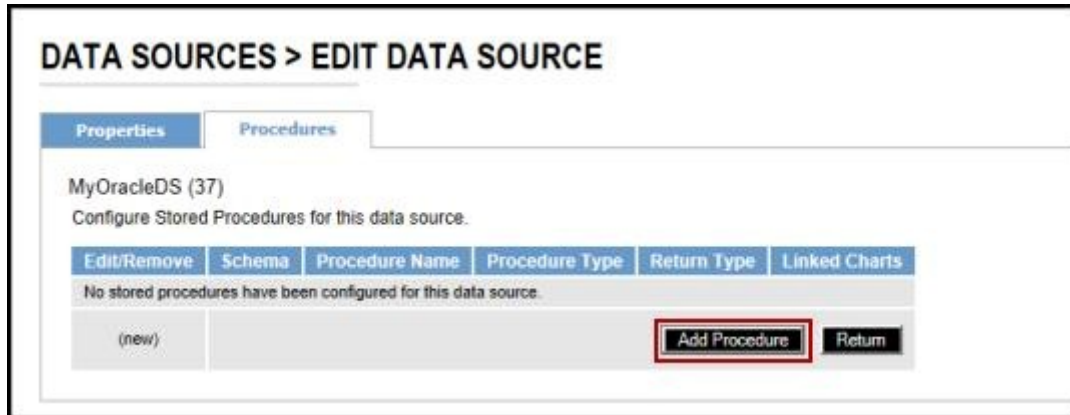


Figure 1

3. Configure the function 'Output Columns' and 'Procedure Arguments' in the iDashboards data source to correspond to the function created in the database. Name the iDashboards 'ProcedureName' 'pkg.y_equals_x_squared' using the proper schema 'Owner'. Refer to Figure below.

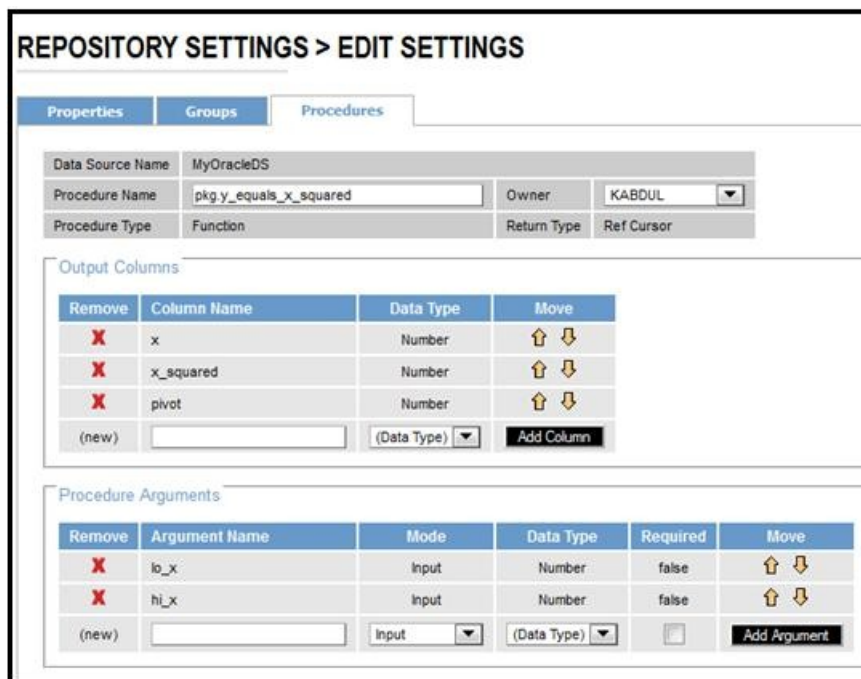


Figure 2



- Now to create a chart, from the data source type panel on the left select the 'Stored Procedures' Option. From the list of stored procedures available under your Oracle data source, pick the 'pkg.y_equals_x_squared'.
- Create a chart with x-axis 'x' and 2 y-axis, y-axis1 as 'x_squared' and y-axis2 as 'pivot'.
- Since the function created in the database requires two input arguments, create two numeric input parameters - 'lo_x' and 'hi_x'. In the input parameters tab, create a chart parameter called 'lo_x' and another called 'hi_x'. Make sure these are static input parameters since in this case we want the user to provide the low and high values to the calculation. A picklist can also be used if necessary.
- The arguments tab, will automatically show the input argument expected by the function. In the 'Value' column, type in the parameter macro as '\${param:lo_x}' for argument 'lo_x' and '\${param:hi_x}' for argument 'hi_x'. Click 'Continue' to select a chart type. Refer to Figure 3 below.

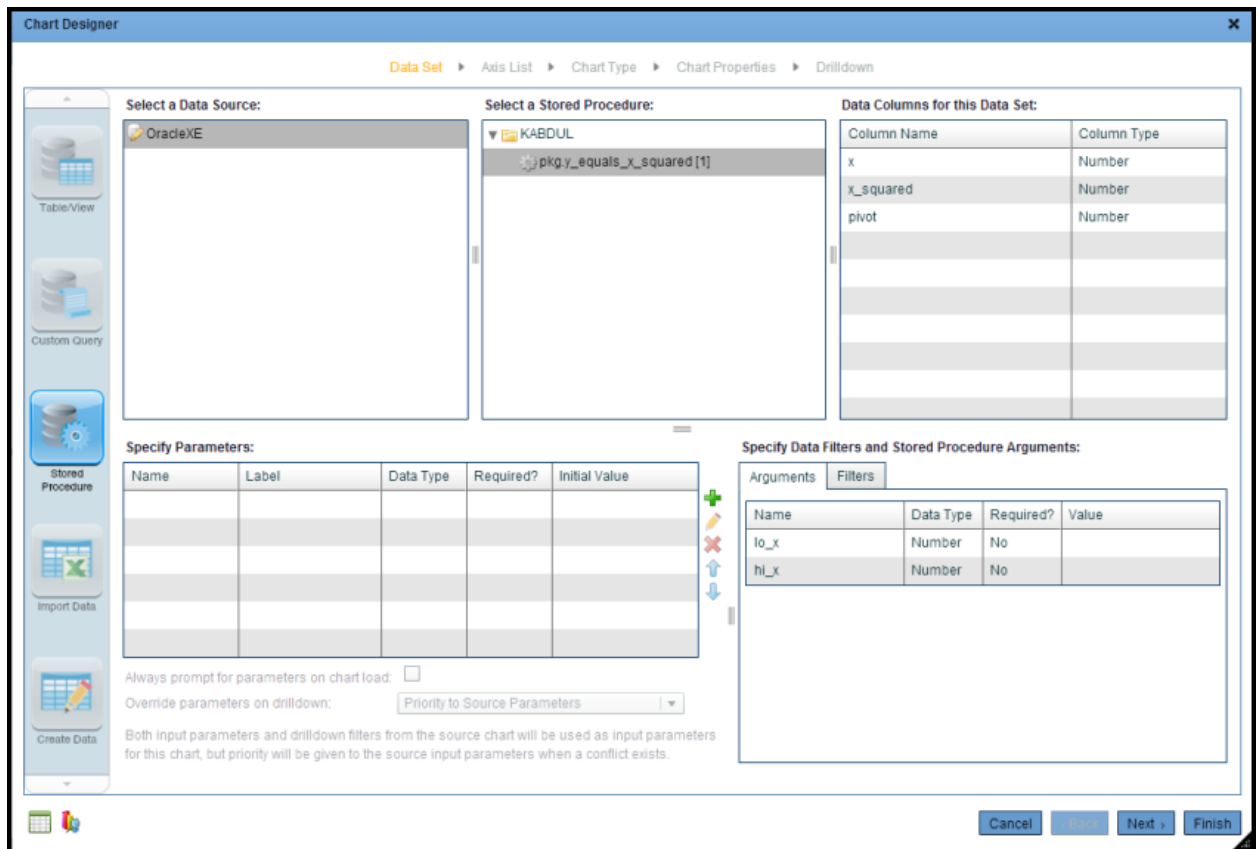


Figure 3

- Click 'Next' to then to the 'Axis List' tab where we can select what resulting data we want to display in the chart. Select all the columns on the left.

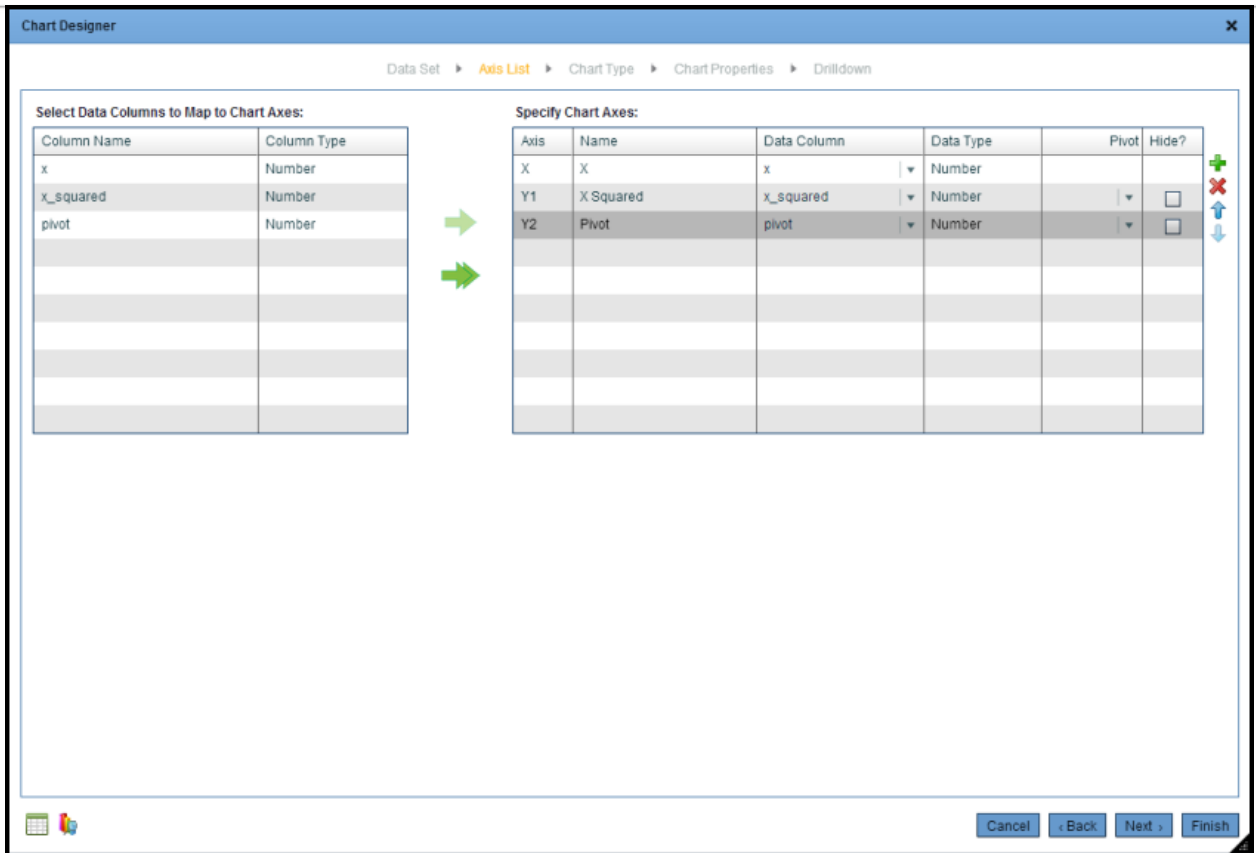


Figure 4

9. Then click 'Next' to select the chart type. Select a 'Tabular Chart'.

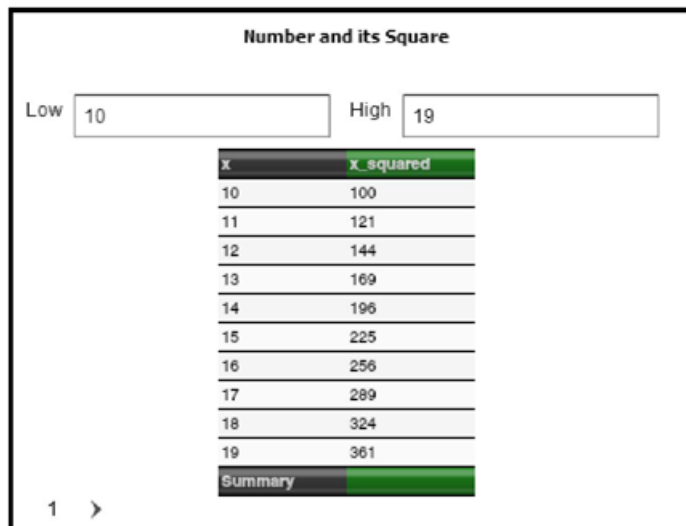


Figure 5