ADF Code Corner

109. How-to further filter detail queries based on a condition in the parent view using ADF BC

twitter.com/adfcodecorner

Abstract:

In Oracle ADF BC, parent – child behavior between view objects is configured through view links you declaratively create between dependent view objects. But what if you need to further filter a detail view object, for example to only show employees of a department with a specific salary. Again, the solution can be conigured on a view link and this article will show you how.

Author: Frank Nimphius, Oracle Corporation
twitter.com/fnimphiu
24-May-2014

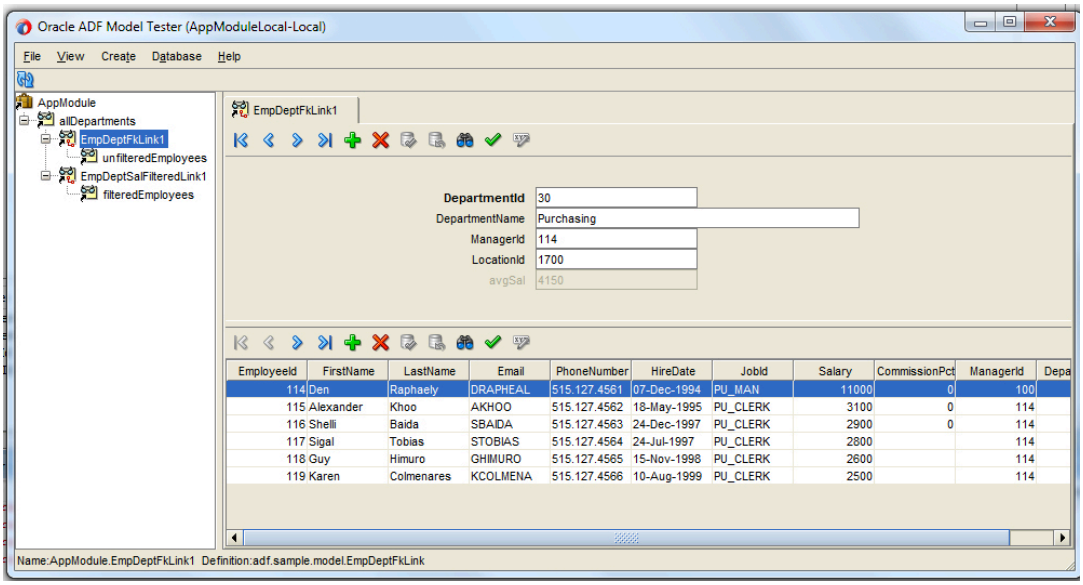
Oracle ADF Code Corner is a loose blog-style series of how-to documents that provide solutions to real world coding problems.

Disclaimer: All samples are provided as is with no guarantee for future upgrades or error correction. No support can be given through Oracle customer support.

Please post questions or report problems related to the samples in this series on the OTN forum for Oracle JDeveloper: http://forums.oracle.com/forums/forum.jspa?forumID=83

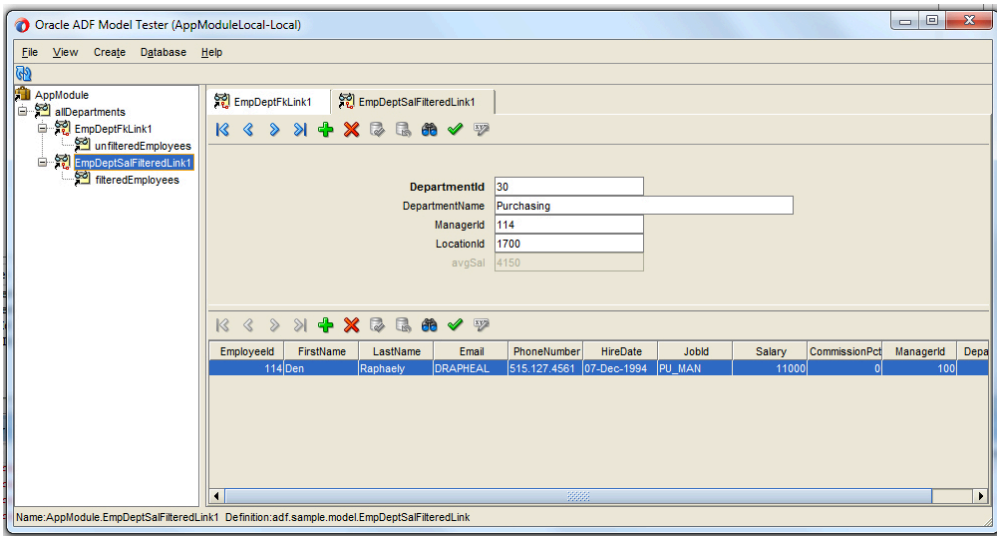
Introduction

The images below show the running sample that shows two instances of the EmployeesView connected with different view links to the DepartmentsView.



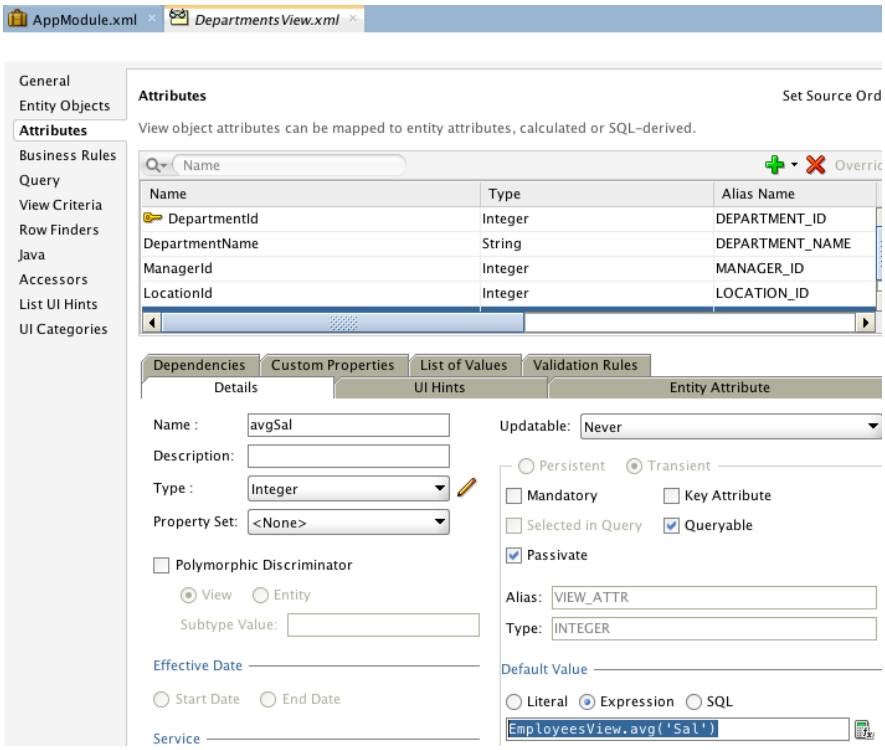
The image above uses the by-default generated view link that filters employee records based on a selected parent department. For the “Purchasing” department the employee count for this link is six (using the Oracle HR sample schema)

The image belo shows an instance of EmployeesView that is based on a view link that filters the returned employee rows by the salary in addition to the parent/detail link. A the delimter for the filtering, the DepartentsView contains an average salary transient attribute. Employees are shown in the sample if the salary is above or equal the average. For the “Purchasing” department this leaves it with a single employee record being returned.

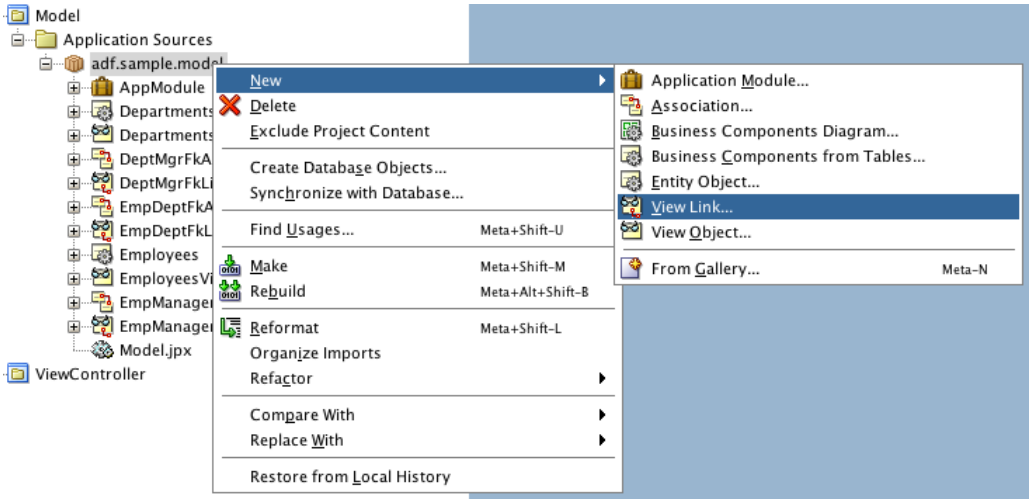


How the sample is built

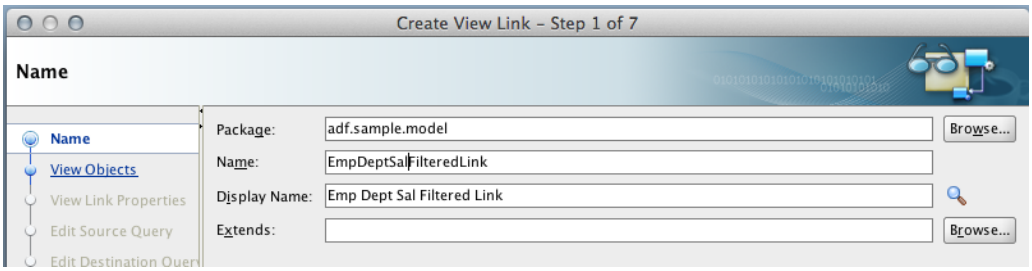
The image below shows the definition of the **avgSal** transient attribute added to the DepartmentsView. As you can see, the value is derived from a Groovy expression accessing the default link to employees.



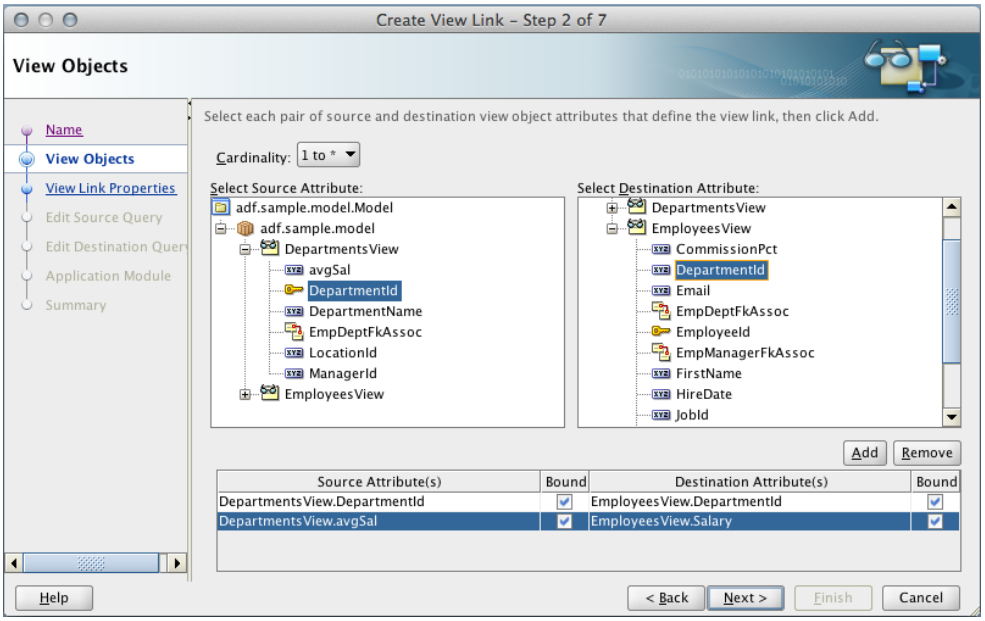
Instead of modifying the existing view link that is created by default, which would have been an option, I created an extra view link. For this, a right mouse click onto the Model project brings up a context menu to access the **New** gallery to create a new ADF Business Components View Link. The image below shows JDeveloper 12c, which is context sensitive in its popup menus.



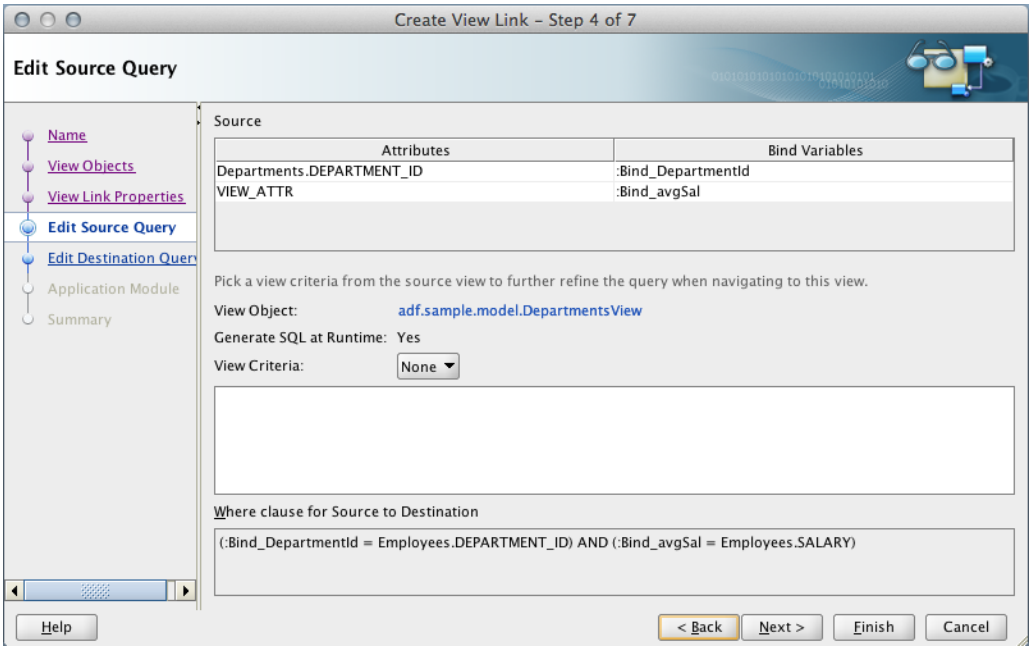
The link is named with the relationship it provides and the additional salary filter.



In the second dialog of the **Create View Link** dialog, there are two configuration required: a definition that queries the EmployeesView based on a selected row in the DepartmentsView. The definition uses the DepartmentId FK constraint. The second configuration is to create a relationship between the **avgSal** in the DepartmentsView and the **Salary** attribute in the EmployeesView.



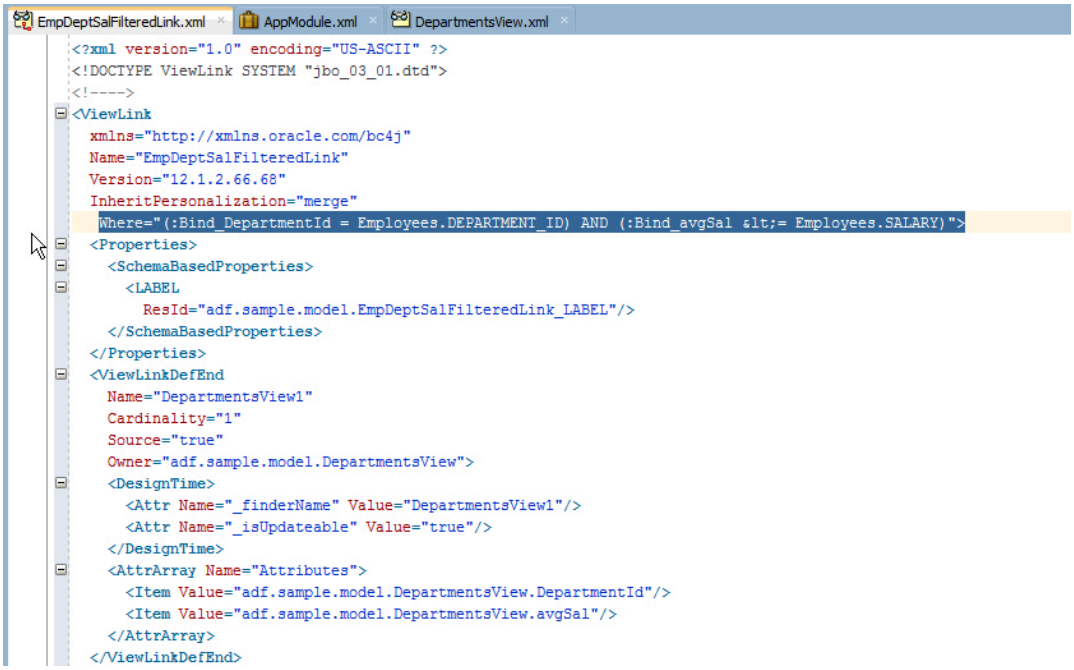
The query that is created based on the setting in the image above is shown below.



Note that, by the time of writing, Oracle JDeveloper 12c doesn't allow you to customize the where clause directly in the **Create View Link** dialog window. Therefore manual correction is required after view link creation as described below.

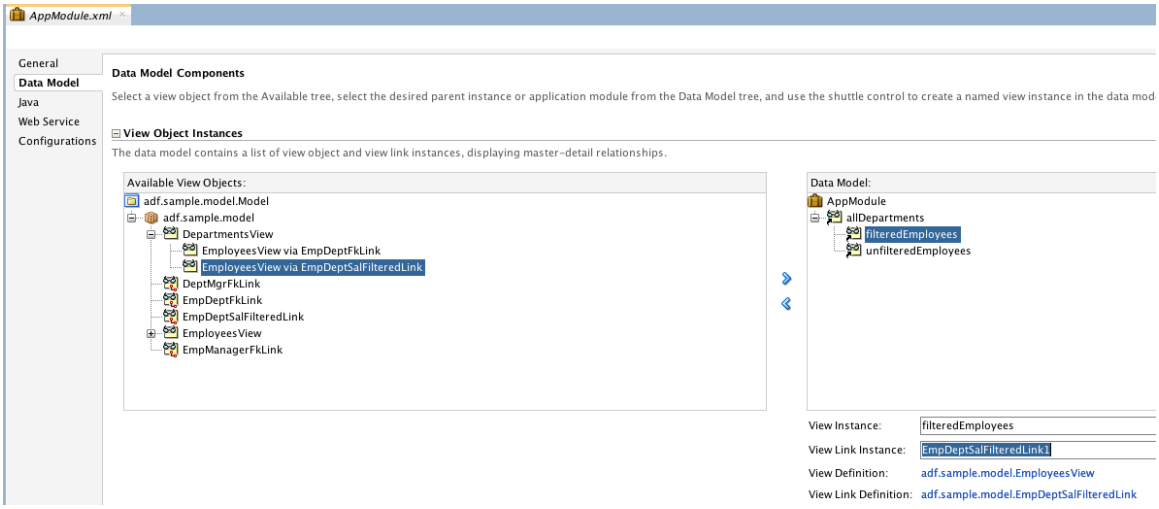
If you are on **JDeveloper 11g**, you can directly change the **:Bind_avgSal = Employees.Salary** part of the where clause to **:Bind_avgSal <= Employees.Salary**

To change the where clause manually in Oracle JDeveloper 12c, double click on the view link file entry in the JDeveloper Application Navigator to open the view link editor. Switch to the Source view (see tab at the bottom of the ViewLink editor) to edit the XML. Edit the XML as shown below and save your work.

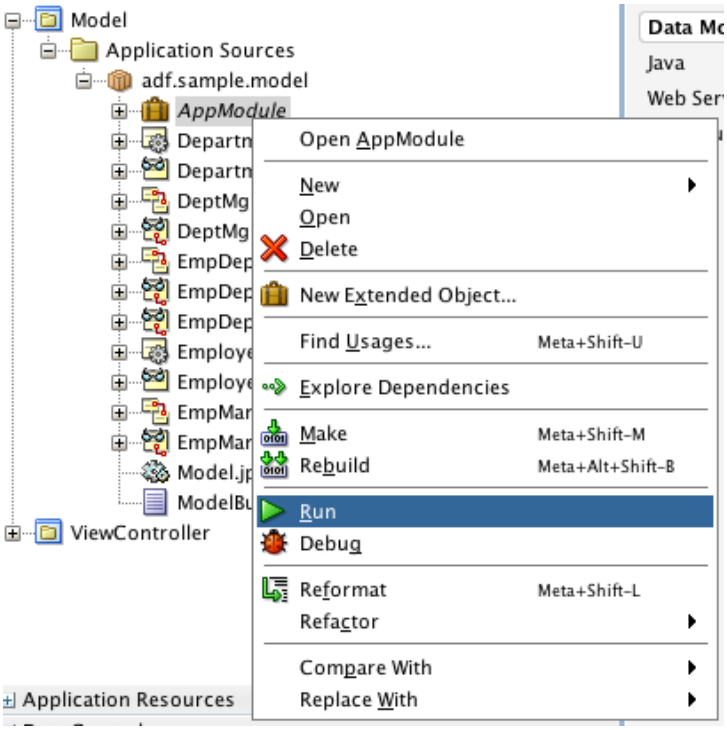


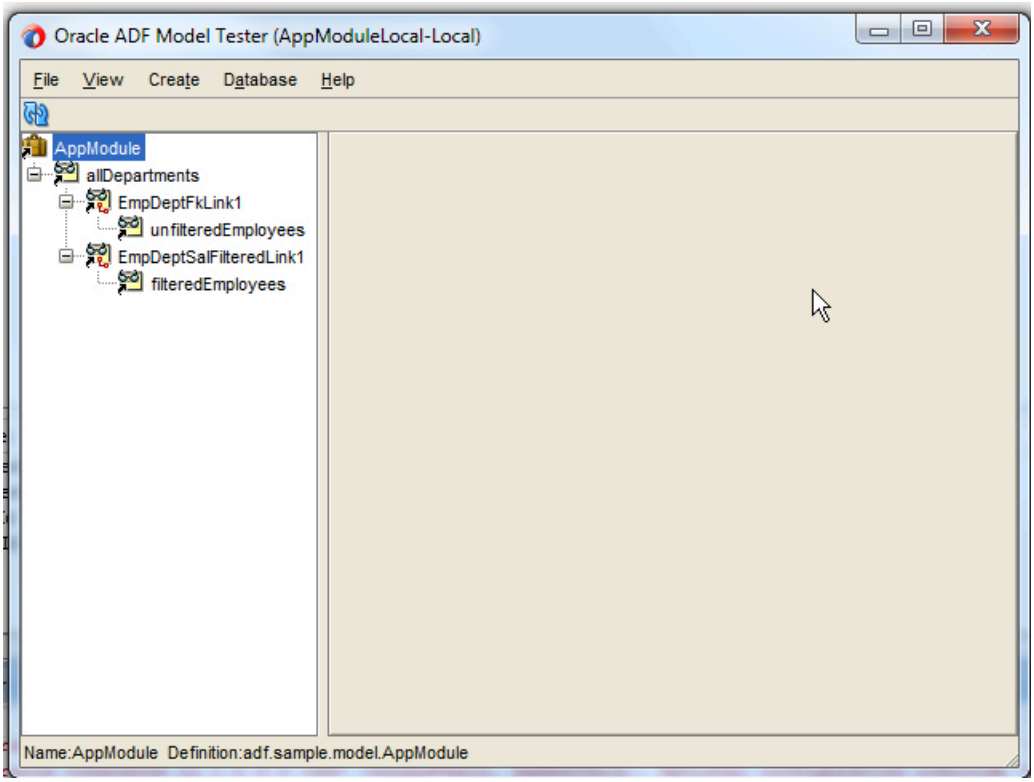
To define an EmployeeView instance based on the newly created view link, **double click** on the Application Module and select the **Data Model** menu option.

In the **Data Model** list, select the department view instance and expand the DepartmentsView object in the **View Object Instances** list. Select the EmployeesView object referenced by the view link that you just created (the one that filters by Salary) and use the arrow right icon to shuttle the employee view object to the data model.



Run the application module tester by selecting the application module and choosing **Run** from the context menu.





Summary

In this article I demonstrated how you can modify a view link to filter detail records based on a condition. The sample application is built with JDeveloper 12.1.2. The solution however should work for all version of JDeveloper 11g and 12c.

You can download the sample application in a ZIP file from the ADF Code Corner website. Just look out for sample #109. Before you can run the sample, ensure you configured he database connect to point to a HR schema in a local database.

RELATED DOCOMENTATION
