

**How-to Guide
SAP NetWeaver 2004**



How To... SQL Editor within Visual Composer

Version 1.00 – March 2006

DRAFT – PRELIMINARY VERSION

**Applicable Releases:
SAP NetWeaver '04**

© Copyright 2005 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data

contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

1 Scenario

This document discusses the nuances of using SQL Editor to pull from JDBC sources within the pilot release of Visual Composer 7.0 on SAP NetWeaver 2004.

2 Introduction

When using the SQL Editor within Visual Composer, there are many nuances you need to be aware of.

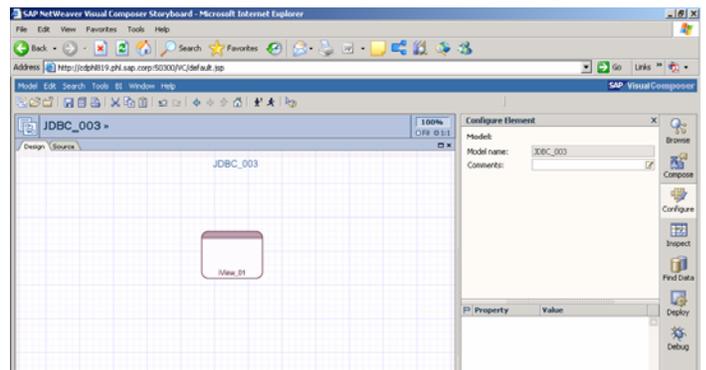
1. To build your SQL Statements to select on values that are specified within an input form in VC, you will have to build your SQL Statement on the input port.
2. It is recommended to never update the SQL directly within the SQL Editor. Instead, you can use the "SQL_Statement" field on the input port. This will overwrite the SQL call to the database.
3. When entering SQL code into the "SQL_Statement" field on the input port, you are using the Visual Composer formula editor. Make sure you enter your SQL Statement as a string. It is recommended to use double quotes around the entire SQL Statement and single quotes around values within the SQL code. Also, you can use Visual Composer concatenation ("&") to concatenate fields within your SQL call.
4. When concatenating fields within your SQL Statement, make sure you cover the case where users don't type in a value to filter on, or you make the user input parameters mandatory fields.
5. When modifying the SQL_Statement and a space exist in a table name (ie. My Budget Table), square brackets must be used around the table when being referenced: [My Budget Table]
"SELECT t1.OrderID, t1.ProductID, t1.UnitPrice, t1.Quantity,
t1.Discount FROM Northwind.dbo.[Order Details] t1"
6. When using any data service from a JDBC connection, the input and output ports are the same. The only fields that are available in the input ports are the fields in the output port.

3 Step by Step

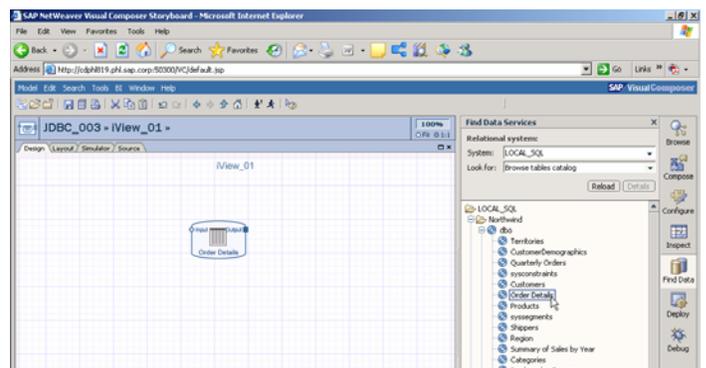
3.1 Build a SQL Statement using the Northwind database and the “Order Details” table

Within this scenario, we will show how to manually write SQL to achieve a filter that would otherwise not be possible. For example, it would be very easy to filter the “Order Details” table from the Northwind database to filter “ProductID = 42”. This is because we can just enter this value in an input form and map this field to the input port for this table, which includes the field “ProductID”. However, if we wanted to select data where “ProductID > 42”, this would involve writing SQL Code. This is what we will do in this example.

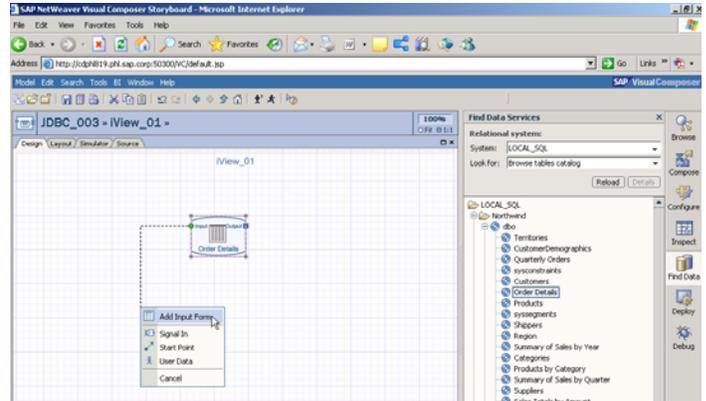
1. Login to Visual Composer and build an iView. Make sure you have already connected a SQL Server database that has the sample database “Northwind” loaded.



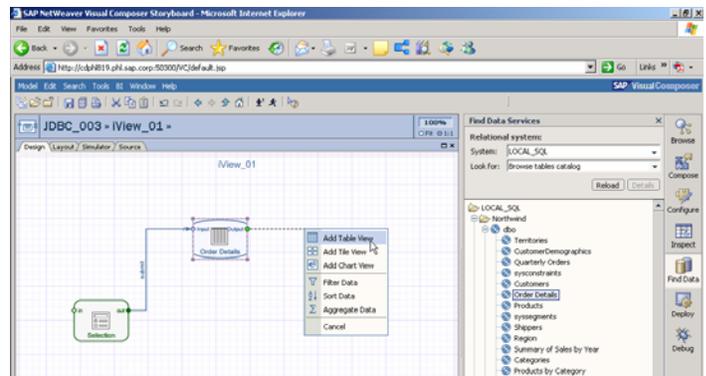
2. Find the “Order Details” table from your database and drag and drop it into your iView.



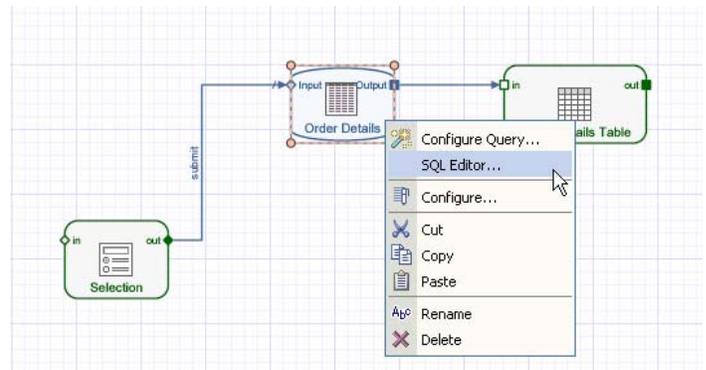
3. Drag out an input form from the input port and name the form "Selection".



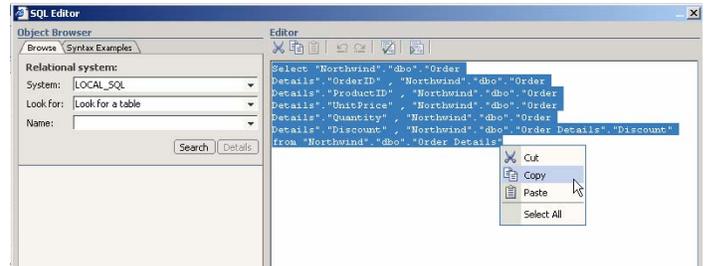
4. Drag and drop a table from the output port and name the table "Order Details Table".



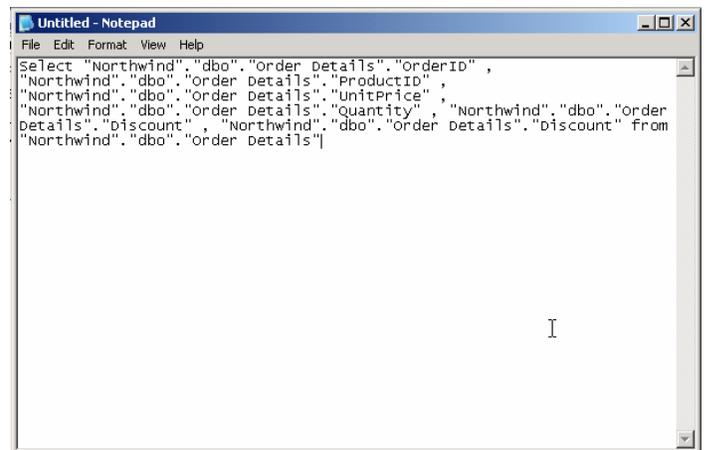
5. Right click on your data service and choose "SQL Editor".



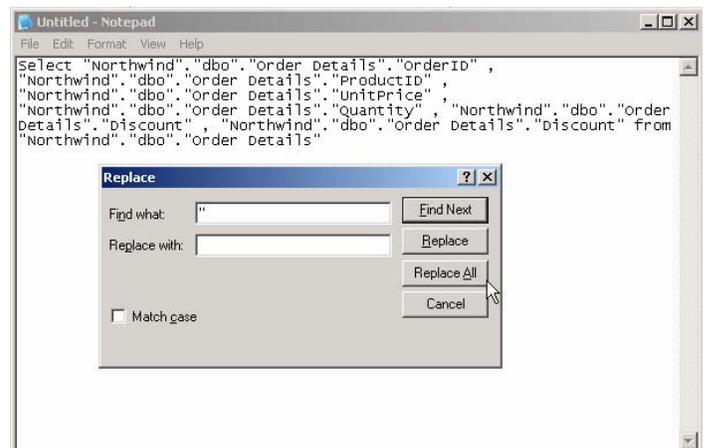
- Highlight all your code and choose "Copy".



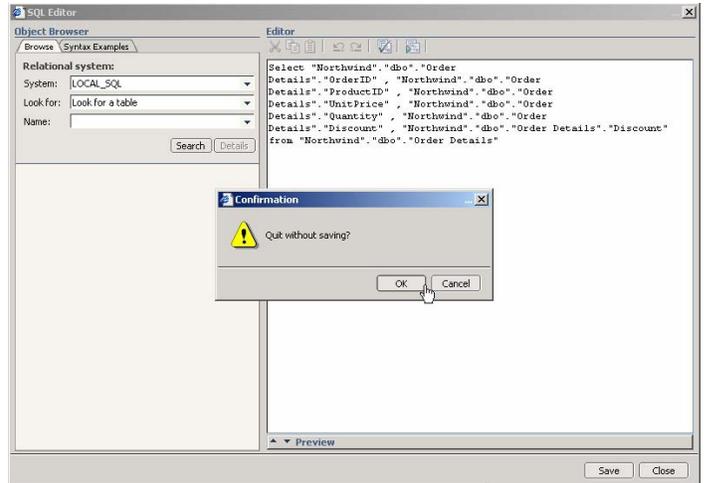
- Paste all the code into Notepad.



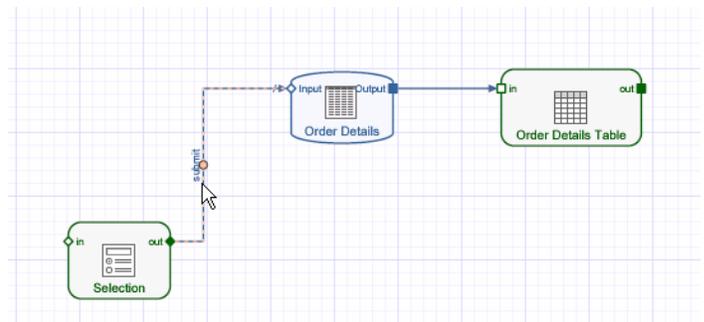
- Do a find and replace and type in a double quote for the find field, and leave the replace field empty. Do not put a space in the replace field. Then hit replace all.



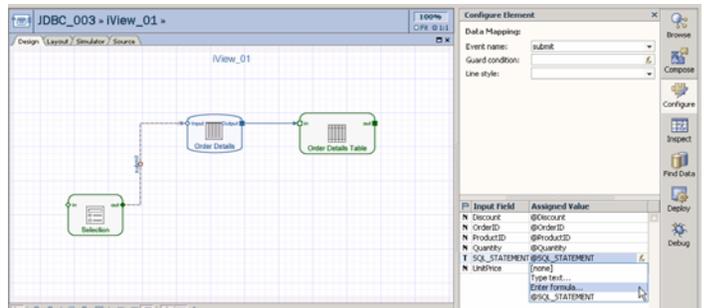
- Go back to Visual Composer and close the SQL Editor. Do not save any changes or make any changes within the SQL Editor directly.



- Click on the line feeding the Input Port of your data service.

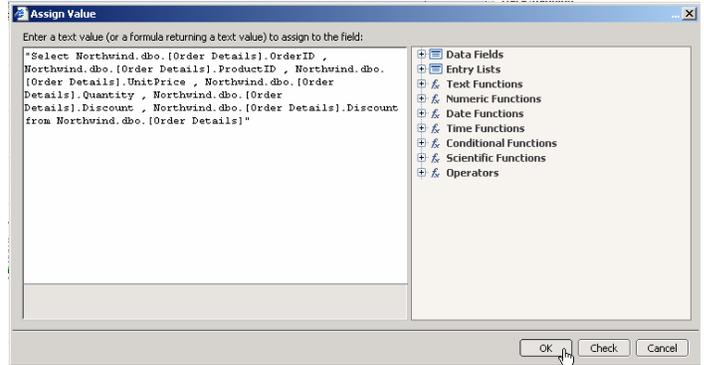


- On the "SQL_STATEMENT" field, choose "Enter formula".



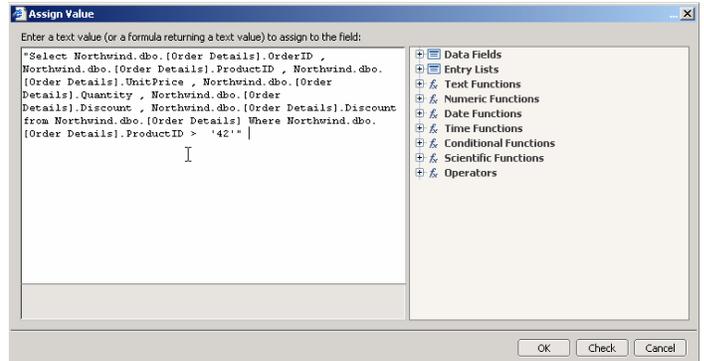
12. Copy and paste all your code from notepad in this formula editor but make sure you put double quotes around the statement and hit check to validate. Also, since the table "Order Details" has a space, we will need to surround this with brackets. This should validate.

NOTE: Within this window, you can enter any SQL Statement you want, but you must enter it as a string which is surrounded with double quotes.



13. In this example, we want to select data where ProductID > 42, so we will update our SQL Statement to do this select.

"Select Northwind.dbo.[Order Details].OrderID , Northwind.dbo.[Order Details].ProductID , Northwind.dbo.[Order Details].UnitPrice , Northwind.dbo.[Order Details].Quantity , Northwind.dbo.[Order Details].Discount , Northwind.dbo.[Order Details].Discount from Northwind.dbo.[Order Details] Where Northwind.dbo.[Order Details].ProductID > '42'"

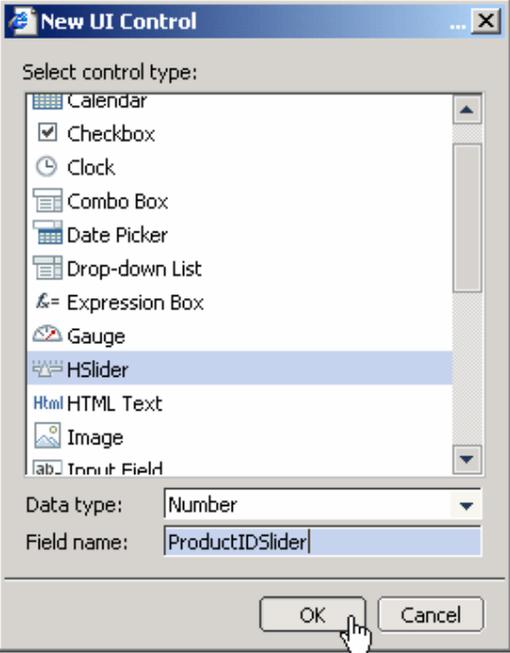


14. Your output table should return only data where ProductID > 42 now. Suppose we wanted to make this

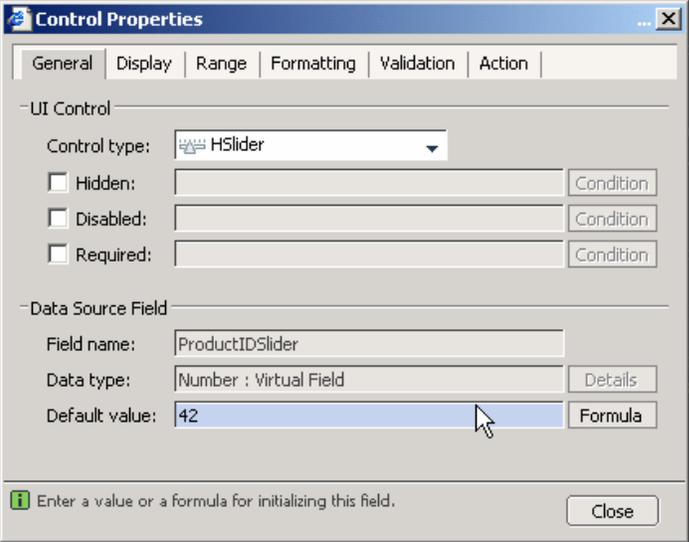
Order Details Table				
Orderid	Productid	Unitprice	Quantity	Discount
10270	43	36.8	25	0
10294	43	36.8	15	0
10302	43	36.8	12	0
10309	43	36.8	20	0
10312	43	36.8	24	0
10340	43	36.8	40	0.05
10458	43	36.8	20	0
10464	43	36.8	3	0
10468	43	36.8	15	0
10507	43	46	15	0.15
10524	43	46	60	0
10530	43	46	25	0

Now we want to enhance this scenario to have a horizontal slider to set our productid value. We will select all data where the ProductID is set by a horizontal slider.

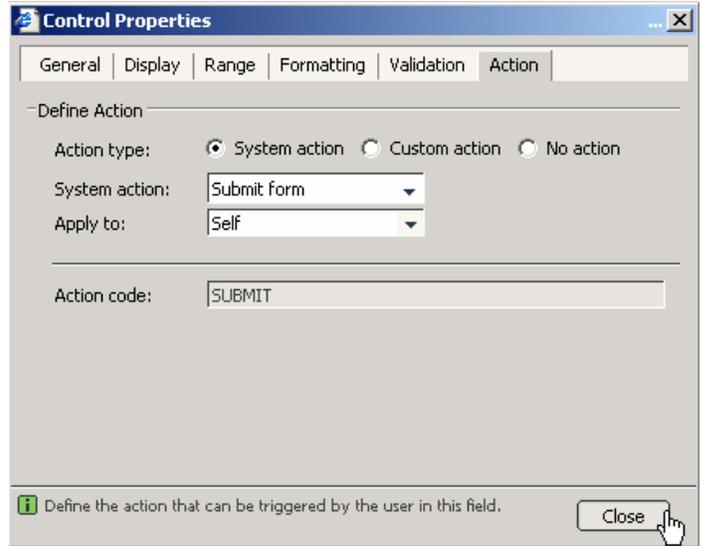
15. Within your model, add a ProductID Slider to your selection form.



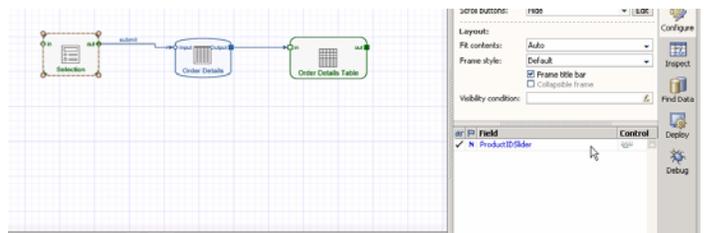
16. Go to the properties of the Slider Object and enter the default value 42.



- Go to the "Action" tab and define "Submit form" as the system action and "Self" for the Apply to.

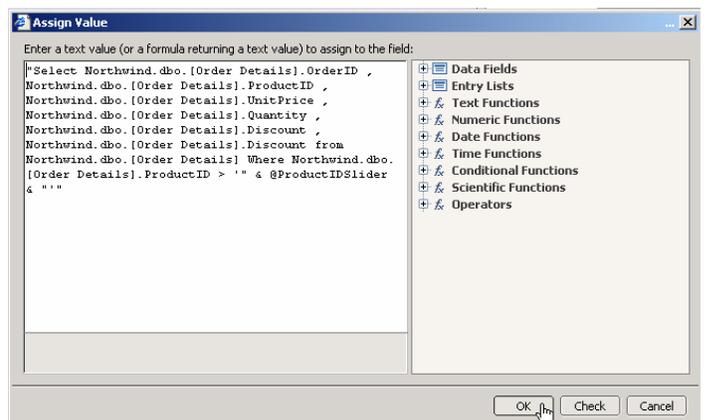


- Delete all other values in your Selection Form except for this slider value.



- Update the SQL_Statement on the line connecting the Selection form and the Data service to the following:

```
"Select Northwind.dbo.[Order Details].OrderID ,
Northwind.dbo.[Order Details].ProductID ,
Northwind.dbo.[Order Details].UnitPrice ,
Northwind.dbo.[Order Details].Quantity ,
Northwind.dbo.[Order Details].Discount ,
Northwind.dbo.[Order Details].Discount from
Northwind.dbo.[Order Details] Where Northwind.dbo.
[Order Details].ProductID > "" & @ProductIDSlider
& """
```



20. Run your model. You should now be able to filter your table based on a dynamic value chosen at runtime!

The screenshot shows a 'Selection' window with a slider control labeled 'Productidslider'. Below it is an 'Order Details Table' with the following data:

Orderid	Productid	Unitprice	Quantity	Discount
10255	36	15.2	25	0
10270	36	15.2	30	0
10298	36	15.2	40	0.25
10313	36	15.2	12	0
10337	36	15.2	20	0
10342	36	15.2	40	0.2
10358	36	15.2	20	0.05
10371	36	15.2	6	0.2
10406	36	15.2	5	0.1
10525	36	19	30	0
10527	36	19	30	0.1
10547	36	19	60	0

4 Syntax Examples

Example of filtering between 2 values:

```
"SELECT t1.ProductID, t1.ProductName, t1.SupplierID, t1.CategoryID, t1.QuantityPerUnit,
t1.UnitPrice, t1.UnitsInStock, t1.UnitsOnOrder, t1.ReorderLevel, t1.Discontinued,
t2.CategoryID, t2.CategoryName, t2.Description, t2.Picture, t3.SupplierID, t3.CompanyName,
t3.ContactName, t3.ContactTitle, t3.Address, t3.City, t3.Region, t3.PostalCode,
t3.Country, t3.Phone, t3.Fax, t3.HomePage FROM Northwind.dbo.Products t1,
Northwind.dbo.Categories t2, Northwind.dbo.Suppliers t3 WHERE (t1.SupplierID =
t3.SupplierID) AND (t1.CategoryID = t2.CategoryID) AND t1.ProductID BETWEEN" & "" & @One
& "" & "AND" & "" & @Two & ""
```

Example of being able to handle "Blank input" as well:

```
AND t2.CategoryName LIKE '%' & IF(ISNULL(@CatName),'%',@CatName) & "%'" & "AND t3.City LIKE
'" & IF(ISNULL(@City),'%',@City) & "%'"
```

We are using the **LIKE** statement in order to filter on partial inputs (ie. searching fields on partials; filter for Philadelphia by only typing in Phil)
SQL works by typing a % after the **Phil** in order to filter just by those first few letters (ie. Phil%).

So to get the SQL_Statement to work with a blank field and without one we have to use the following setup:

```
IF(ISNULL(@City),'%',@City) & "%'"
```

-- We are assigning a % if the input field is null and assigning % to any input so we the SQL is properly configured. Therefore you can have as input:
<blank> or <Phil%> or <Philadelphia%>

In either case, the % is assigned behind the scenes so the user doesn't have to worry about inputting a %