# Filter using ${vars} in the select

This template is described as used by the code samples FilterVariablesXML.cs and FilterVariablesXML.java.

These first 4 tables each use a ${var} in the condition of the select. If you were not using filters you would call DataSourceProvider.Map/setMap() passing a hashtable where the key of each entry is the below 3 var names and the value would be an int, string, & int.

There is no date example because XPath views a date as a string. (You can only do inequalities in XPath with numbers.)

However, for filters you can instead pass a FilterCondition or FilterList(C#) / SelectFilter or SelectList(Java). The variables do **not** need to be defined in AutoQuery as select variables - ${id} is not. The select requirement is only for AutoQuery, not the engine.

The comparison in the forEach tags below is irrelevant when passing in a filter. The ones below each use a different comparison and you will see that makes no difference. That is because the engine replaces the condition with the condition created by the filter. All that matters is that there is a condition of some kind using the variable so it can be replaced.

## Table filtered by Employee.EmployeeID = ${employeeId}

|  |  |  |  |
| --- | --- | --- | --- |
| EmployeeID | LastName | FirstName | BirthDate |
| [Employee][EmployeeID] | [LastName] | [FirstName] | [BirthDate] |

:forEach]

## Table filtered by Employee.FirstName != ${employeeName}

|  |  |  |  |
| --- | --- | --- | --- |
| EmployeeID | LastName | FirstName | BirthDate |
| [Employee][EmployeeID] | [LastName] | [FirstName] | [BirthDate] |

:forEach]

## Table filtered by Employee.EmployeeID = ${id} (not a select variable)

|  |  |  |  |
| --- | --- | --- | --- |
| EmployeeID | LastName | FirstName | BirthDate |
| [Employee][EmployeeID] | [LastName] | [FirstName] | [BirthDate] |

:forEach]

# Filtering with no ${var} in the select

In the following table there is no use of a ${var} in the select. We can still apply an ad-hoc filter. The one requirement is the table of the column we filter on must be returned in the select. (In other words, the select must return a column from the table that the ad-hoc column is in.)

We are using a node other than Employee here because, if we used a global ad-hoc variable in Employee it would be applied to all the above tables also.

|  |  |  |
| --- | --- | --- |
| TerritoryID | TerritoryDescription | RegionID |
| [Territory][TerritoryID] | [TerritoryDescription] | [RegionID] |

:forEach]