This table represents an detail overview of supported OCL operators in our implementation `OCL2PSQL`. It consists of several categories, in which contains 2 columns: the operator (on the left) and the syntax example (on the right).

Class Operations	
`allInstances`	*class*`.allInstances()`
`attributes`	*var*`.att`
`association ends`	*var*`.assoc`
Constants Operations	
`Boolean Literal`	{`TRUE`, `FALSE`}
`Integer Literal`	{...,`-2`, `-1`, `0`, `1`, `2`,...}
`String Literal`	`'string'`
Boolean Operations	
`not`	`NOT` *bool-exp*
`and`, `or`, `xor`	*bool-exp* `logic-operator` *bool-exp'*
`=`, `<>`, `>`, `<`, $\leq$, $\geq$	*bool-exp* `compare-operator` *bool-exp'*
Iterative Operations	
`collect`	*source-exp*$\rightarrow$`collect(`*var* $\mid$ *body-exp*`)`
`forAll`	*source-exp*$\rightarrow$`forAll(`*var* $\mid$ *bool-exp*`)`
`exists`	*source-exp*$\rightarrow$`exists(`*var* $\mid$ *bool-exp*`)`
`select`	*source-exp*$\rightarrow$`select(`*var* $\mid$ *bool-exp*`)`
`reject`	*source-exp*$\rightarrow$`reject(`*var* $\mid$ *bool-exp*`)`
`size`	*source-exp*$\rightarrow$`size()`
`isEmpty`	*source-exp*$\rightarrow$`isEmpty()`
`notEmpty`	*source-exp*$\rightarrow$`notEmpty()`
`flatten`[1]	*source-exp*$\rightarrow$`flatten()`
Bag Operations	
`asSet`	*bag-exp*$\rightarrow$`asSet()`

[1] For the time being, this can operate after the `collect` operation