The Syntax of Relational Operators

This document records the syntax – i.e. keywords and parameters - of all the operators at the relational level of abstraction.

Note: these operators are all members of the syntax categories Monadic-Operator and Dyadic-Operator given in the document “BNF Syntax Rules for RAQUEL Relational Algebra”.

First the syntax of the monadic operators is given, listed in order of the semantic categories used in the document “Categorisation of Relational Operators by Semantics”, then the syntax of the dyadic operators listed in the same semantic order.

This is followed by the syntax of the parameters.

Monadic Operators

Operators to extract part of a relvalue.

Monadic-Operator ::= Project[ AttributeName-Set ] |
Restrict[ TruthValued-Expression ]

Operators to execute calculations using a relvalue’s attribute values.

Monadic-Operator ::= Extend[ AttributeValue-Assignment-Set ] |
GroupBy[ AttributeName-Set ] With[ Aggregate-AttributeValue-Assignment-Set ]

Operators to restructure a relvalue.

Monadic-Operator ::= Nest[ AttributeName-MultiAssignment-Set ] |
Unnest[ AttributeName-MultiAssignment-Set ] |
Rename[ AttributeName-SingleAssignment-Set ]

Operators to obtain meta data about a relvalue.

Monadic-Operator ::= Meta[ MetaParameters ]
Operators to merge relvalues by tuple.

Monadic-Operator ::= Dist[ Union ] | Dist[ Intersect ]

Dyadic Operators

Operators to compare 2 relvalues.

Dyadic-Operator ::= Set-Comparison |

\[
\begin{align*}
&\text{Divide} | \text{Divide[ Set-Comparison ]} | \\
&\text{Divide[ Set-Comparison ]} | \text{Divide[ Set-Comparison ]} \\
&\text{Divide[ Set-Comparison ]Relvalue} | \\
&\text{Divide Relvalue[ Set-Comparison ]} \\
&\text{Divide Relvalue[ Set-Comparison ]Relvalue}
\end{align*}
\]

Set-Comparison ::= = | ~= | 

\[
\begin{align*}
&\text{Super} | \text{Super=} | \text{Sub=} | \text{Sub} | \\
&\text{Disjoint} | \sim\text{Disjoint} | \\
&\text{Member} | \sim\text{Member}
\end{align*}
\]

Operators to merge 2 relvalues together by tuple.

Dyadic-Operator ::= Union | Intersect | Diff

Operators to merge 2 relvalues together by tuple.

Dyadic-Operator ::= Join[ AttributeName-MinSet ] | \\
Join[ AttributeName-MinSet ] | Join[ AttributeName-MinSet ]] | \\
Join[ AttributeName-MinSet ]]Relvalue |
Join Relvalue[ AttributeName-MinSet ]] |
Join Relvalue[[ AttributeName-MinSet ]]Relvalue |
Gen[ Dyadic-TruthValued-Expression ] |
Gen[[ Dyadic-TruthValued-Expression ]]Relvalue |
Gen Relvalue[ Dyadic-TruthValued-Expression ]] |
Gen Relvalue[[ Dyadic-TruthValued-Expression ]]Relvalue

Parameters
The parameters are listed in alphabetic order of the names of the non-terminals used to represent parameters. Where further production rules are needed to explain a parameter, they come immediately after the parameter’s production rule.

On the right-hand side of production rules, ‘|’ separates alternatives.
If an item on the right-hand side has the suffix ‘--×’, then that item can appear zero, one or more times in the production rule.
If an item on the right-hand side has the suffix ‘--+’ then that item can appear one or more times in the production rule.
The separator that is necessary between 2 or more items is ignored below. In fact it is a comma if the items are names and a semi-colon if they are operator or assignment expressions.

Notes are added where production rules are inadequate to explain everything about the required syntax.
‘Operator-Expression’ is defined in “BNF Syntax Rules for RAQUEL Relational Algebra”.
‘ScalarOperator-Expression’, ‘ScalarVariable’ and ‘ScalarValue’ are defined in “BNF Syntax Rules for RAQUEL Scalar Algebra”.
-------------------------------
Aggregate-AttributeValue-Assnment-Set
 ::= Aggregate-AttributeValue-Assnment--+

Used by the GroupBy operator.
Aggregate-AttributeValue-Assignment ::= ScalarAttributeName <-- Aggregate-ScalarExpression \| RelationalAttributeName <-- Aggregate-RelationalExpression

Aggregate-ScalarExpression ::= Aggregate-ScalarExpression \| ( Aggregate-ScalarExpression ) \| Aggregate-ScalarExpression Monadich-ScalarOperator \| Aggregate-ScalarExpression Dyadic-ScalarOperator ( ScalarExpression ) \| Aggregate-ScalarExpression Dyadic-ScalarOperator ScalarValue \| Aggregate-ScalarExpression Dyadic-ScalarOperator ScalarVariable \| ScalarExpression Dyadic-ScalarOperator ( Aggregate-ScalarExpression )

Aggregate-ScalarExpression ::= Aggregatable-Expression Aggregate-Operator

Aggregatable-Expression ::= Bag[AttributeName-Set] \| Project[AttributeName-Set]

It is intended in future to include a Sequence operator among Aggregatable-Expressions, and allow Aggregatable-Expressions to be an operand in a (relational) operator expression.

Aggregate-Operator ::= Sum \| Min \| Max \| Meta \| Nest[AttributeName-MultiAssignment-Set] \| Any \| All

It is intended to revise Aggregate-Operators in future, possibly allowing the use of a Distributed operator.

Aggregate-RelationalExpression ::= Operator-Expression Dist[Union] \| Operator-Expression Dist[Intersect]

AttributeMinSet ::= AttributeName--+ \| ~ AttributeName--+
'MinSet' refers to the fact that currently Join does not provide a Cartesian product function, so there must always be at least one attribute, which must be common to both operands.

Attribute-Set ::= Attribute--x | ~ Attribute--x

Used by the Join operator.

-----------------------------

AttributeName-MultiAssignment-Set ::= AttributeName-MultiAssignment--+

Used by the Project and GroupBy operators.

-----------------------------

AttributeName-MultiAssignment ::= AttributeName <-- AttributeName--+

Used by the Nest and Unnest operators.

-----------------------------

AttributeName-SingleAssignment-Set ::= AttributeName-SingleAssignment--+

-----------------------------

AttributeName-SingleAssignment ::= AttributeName <-- AttributeName

Used by the Rename operator.

-----------------------------

AttributeValue-Assignment-Set ::= AttributeValue-Assignment--+

-----------------------------

AttributeValue-Assignment ::= ScalarAttributeName <-- ScalarOperator-Expression | RelationalAttributeName <-- Insert Operator-Expression |

Used by the Extend operator.

Used by the \textit{Gen} operator.

ScalarDyadic-TruthValued-Comparison ::= 
ScalarOperator-Expression \textbf{DyadicScalar-Comparator} ( ScalarOperator-Expression ) | 
ScalarOperator-Expression \textbf{DyadicScalar-Comparator} ScalarVariable

For each comparison in ‘ScalarDyadic-TruthValued-Comparison’, the attributes in the comparator’s left-hand operand must be drawn from the relational operator’s left-hand operand, and the attributes in the comparator’s right-hand operand must be drawn from the relational operator’s right-hand operand.

An actual \textbf{DyadicScalar-Comparator} is one drawn from the set available for the relevant scalar type (which must include = and \textasciitilde=).

RelationalDyadic-TruthValued-Comparison ::= 
Operator-Expression \textbf{Set-Comparison} ( Operator-Expression ) | 
Operator-Expression \textbf{Set-Comparison} Relvar

For each comparison in ‘RelationalDyadic-TruthValued-Comparison’, the attributes in the comparator’s left-hand operand must be drawn from the relational operator’s left-hand operand, and the attributes in the comparator’s right-hand operand must be drawn from the relational operator’s right-hand operand.

-------------------------------

\textbf{MetaParameters} ::= \textit{the full set is not yet defined}

Used by the \textit{Meta} operator.

Each item in the set specifies a particular kind of meta data that is returned within a relvalue format.

-------------------------------
TruthValued-Expression ::= TruthValue | NOT TruthValue |
                        (TruthValued-Expression) | NOT (TruthValued-Expression) |
                        TruthValued-Expression AND (TruthValued-Expression) |
                        TruthValued-Expression OR (TruthValued-Expression) |
                        TruthValued-Expression XOR (TruthValued-Expression) |
                        Scalar-TruthValued-Comparison | Relational-TruthValued-Comparison

Used by the Restrict operator.

TruthValue ::= Scalar-TruthValue | Relational-TruthValue

Scalar-TruthValued-Comparison ::= ScalarOperator-Expression MonadicScalar-Comparator |
                                ScalarOperator-Expression DyadicScalar-Comparator (ScalarOperator-Expression) |
                                ScalarOperator-Expression DyadicScalar-Comparator ScalarVariable |
                                ScalarOperator-Expression DyadicScalar-Comparator ScalarValue

Relational-TruthValued-Comparison ::= Operator-Expression Set-Comparison (Operator-Expression) |
                                  Operator-Expression Set-Comparison Relvar |
                                  Operator-Expression Set-Comparison RelValue