

FEATURES.DBMS File

Features of the RAQUEL DBMS

This document lists, and briefly describes, the subset of the RAQUEL notation that the prototype DBMS implements.

Relational Subset

Relational Value Assignments

Retrieve : retrieves the value of the relational expression on its right and sends it to the output sink named on its left.

When the Raquel GUI is used with the Raquel DBMS as its 'Driver Application', the GUI's Result frame is a sink named 's' (lower case 's'). So relational values should be retrieved to 's' in order to appear in the GUI.

Insert : inserts the relational value on its right into the relational variable named on its left.

Delete : deletes the relational value on its right from the relational variable named on its left.

Real : assigns the relational value on its right to the real relational variable named on its left. The term 'Real' contrasts with 'View' and means that the LHS variable is a permanent, real variable. (The corresponding SQL term is 'base relation'). If a relational variable of that name does not already exist, it is created with the assigned value; if it does already exist, then that variable's value is replaced by that on the right.

A default storage mechanism is used to hold the data of a new variable.

Relational Algebra Operators

Project : returns a relational value consisting of the specified attributes of the operand relational value. The result has no duplicate tuples. The attributes must be specified by listing their names.

Restrict : returns a relational value consisting of those tuples of the operand which, when the 'restriction condition' is applied to them, return the truth value **true**.

Family of **Natural Join** operators. In addition to the standard **Natural Join** operator, there are also :

- left and right semi join versions of it;
- left, right and full outer join versions of it; these also take a 'default tuple(s)' which specifies what attribute values are to be used in 'padding out' a retained tuple from an operand so that all the requisite attributes appear in the result. Note that RAQUEL does *not* use "nulls".

If there is not at least one attribute that is common to both operands, an error is returned. It has been observed that this typically arises by mistake, and therefore an error is appropriate (as opposed to, say, returning a value that is a Cartesian Product of the operands).

Family of **Generalised Join** operators. In addition to the standard **Generalised Join** operator, there are also :

left and right semi join versions of it;

left, right and full outer join versions of it; these also take a 'default tuple(s)' which specifies what attribute values are to be used in 'padding out' a retained tuple from an operand so that all the requisite attributes appear in the result – note that RAQUEL does *not* use "nulls".

There must be no attributes common to both operands, or an error is returned. If the expression used to match tuples from the operands is just the truth value **true**, a Cartesian Product of the operands is returned. (Note that it is not intended to provide a separate **Cartesian Product** operator, as it would be very little used in practice).

Scalar Subset

Scalar Data Types

The **Truth** type : its 2 possible values are expressed by the keywords **true** and **false**. Because keywords in RAQUEL are case insensitive, the expression of values **true** and **false** are also case insensitive; e.g. they could be expressed as "True", "trUE", "tRUe", "False", faLSE", FALSE" (but abbreviations, e.g. "t", "T", "f", or "F", are *not* accepted).

The **Number** type : its possible values are expressed as a sequence of decimal numeric digits, optionally containing a decimal point.

The **Text** type : its possible values are sequences of characters enclosed in single quote marks. The quote marks are not part of the text value. The text can contain any characters except the single quote character.

Scalar Assignments

Value assignment, written as "<--".

Scalar Operators

For every scalar type, the '=' and '~=' comparison operators.

For the **Truth** type : **And**, **Or**, **Not**, **Xor**.

For the **Number** type :

- the additional comparison operators <, <=, >, >=, which return a **Truth** value;
- +, -, *, /, ^, which return a **Number** value.

For the **Text** type :

- **Cat**, **SubText**, **Position**, which return a **Text** value (in the case of **Position**, the **Text** value consists of the single character at that position);
- **Length**, which returns a **Number** value;
- **Find**, which returns a **Truth** value.