SELECT

Runs a query.

Syntax

```
--query_block
SELECT [DISTINCT] select_item [,select_item]...
FROM table_element [join element]...
[WHERE condition]
[GROUP BY expression [,expression]... [HAVING condition] ] [ORDER BY orderby_item [,orderby_item]... ]
[LIMIT row_count OFFSET row_offset]

--select_item
schema_name.]table_name.]  |  expression [ [as] alias]*

--expression (returns a value)
simple_expression    |  (expression)     |
function(expression) |  case_expression  |
expression [+ | - | * | /] expression

--simple_expression (returns a value)
column_identifier  |  literal  |  NULL

--column_identifier
[[schema_name.]table_name.]column_name

--table_element
[schema_name.]table_name [ [as] alias]  |
(query_block) [as] alias

--join_element
[INNER | LEFT OUTER | RIGHT OUTER | FULL OUTER | CROSS] JOIN table_element ON join_condition

--join_condition (returns TRUE/FALSE/NULL)
expression <=> expression |
condition

--Condition (returns TRUE/FALSE/NULL)
simple_condition              |
(condition)                   |
condition AND | OR condition  |
NOT condition

--simple_condition (returns TRUE/FALSE/NULL)
expression = | != | <> | > | >= | < | <= expression |
expression BETWEEN expression AND expression |
expression LIKE 'string_pattern'                    |
expression IS [NOT] NULL                            |
column [NOT] IN ( query_block | list_of_value )

--orderby_item
expression [ASC | DESC]
```

Parameter Details

Parameter	Details
Joins	there is no limit to the number of tables or sub-queries in a join. Only join conditions that use "equals to" conditions (equi-joins) are supported. Only ANSI syntax is supported (**a join b on condition**). In the ON clause, only join conditions are supported, not single-table filter conditions. Self joins are not supported in the same FROM clause; you can, however, use the same table in different sub-queries. The Join clause in Jethro supports the NULL -safe equal comparison operator (<=>). The expression: **T1 JOIN T2 ON T1.C<=>T2.C** is equivalent to the expression: *T1 JOIN T2 ON T1.C=T2.C OR (T1.C IS NULL AND T2.C IS NULL*)
IN	IN accepts a list of values or an uncorrelated sub-query only a single left column is supported. Correlated sub-queries are not supported.