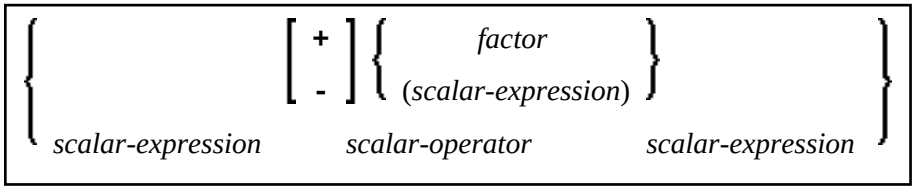
Scalar Expressions



This chapter covers the following topics:

- Scalar Expression
- Scalar Operator
- Factor

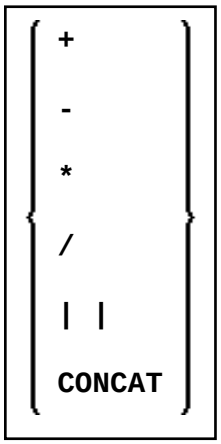
Scalar Expression

A *scalar-expression* consists of a factor or other scalar expressions including scalar operators.

Concerning reference priority, scalar expressions behave as follows:

- When a non-qualified variable name is specified in a scalar expression, the first approach is to resolve the variable name as column name of the referenced table.
- If no column with the specified name is available in the referenced table, Natural tries to resolve this variable as a Natural user-defined variable (host variable).

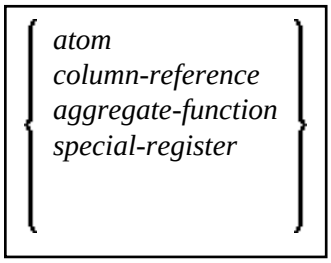
Scalar Operator



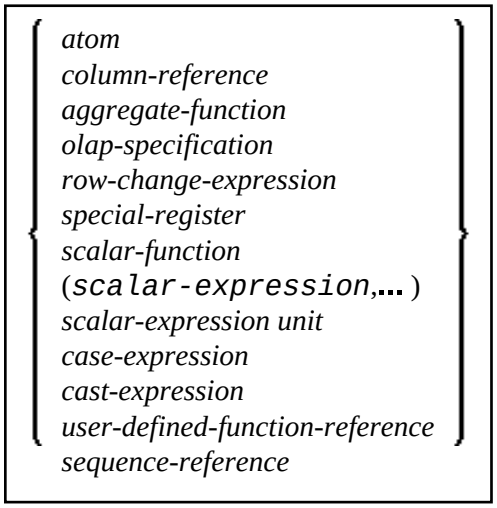
A *scalar-operator* can be any of the operators listed above; the minus (-) and slash (/) operators must be separated by at least one blank from preceding operators.

Factor

Common Set Syntax:

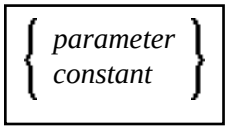


Extended Set Syntax:



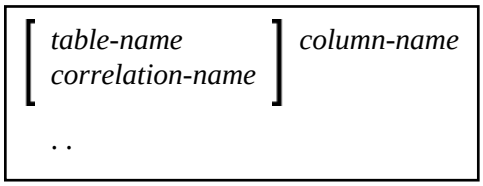
A *factor* can consist of one of the items listed in the above diagram and described in the text below.

Atom



An *atom* can be either a *parameter* or a *constant*; see also the section *Basic Syntactical Items*.

Column Reference



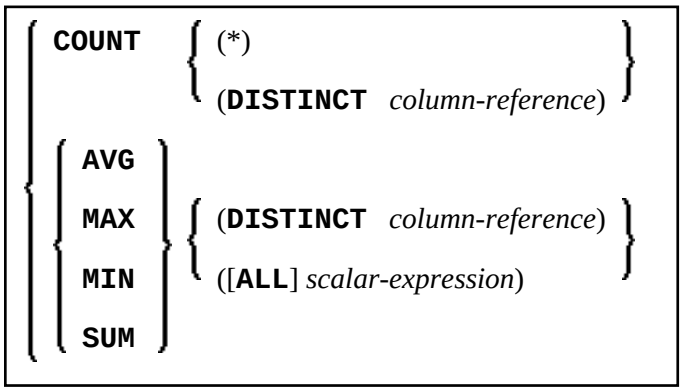
A *column-reference* is a column name optionally qualified by either a *table-name* or a *correlation-name* (see also the section *Basic Syntactical Items*). Qualified names are often clearer than unqualified names and sometimes they are essential.

Note:
A table name in this context must not be qualified explicitly with an authorization identifier. Use a correlation name instead if you need a qualified table name.

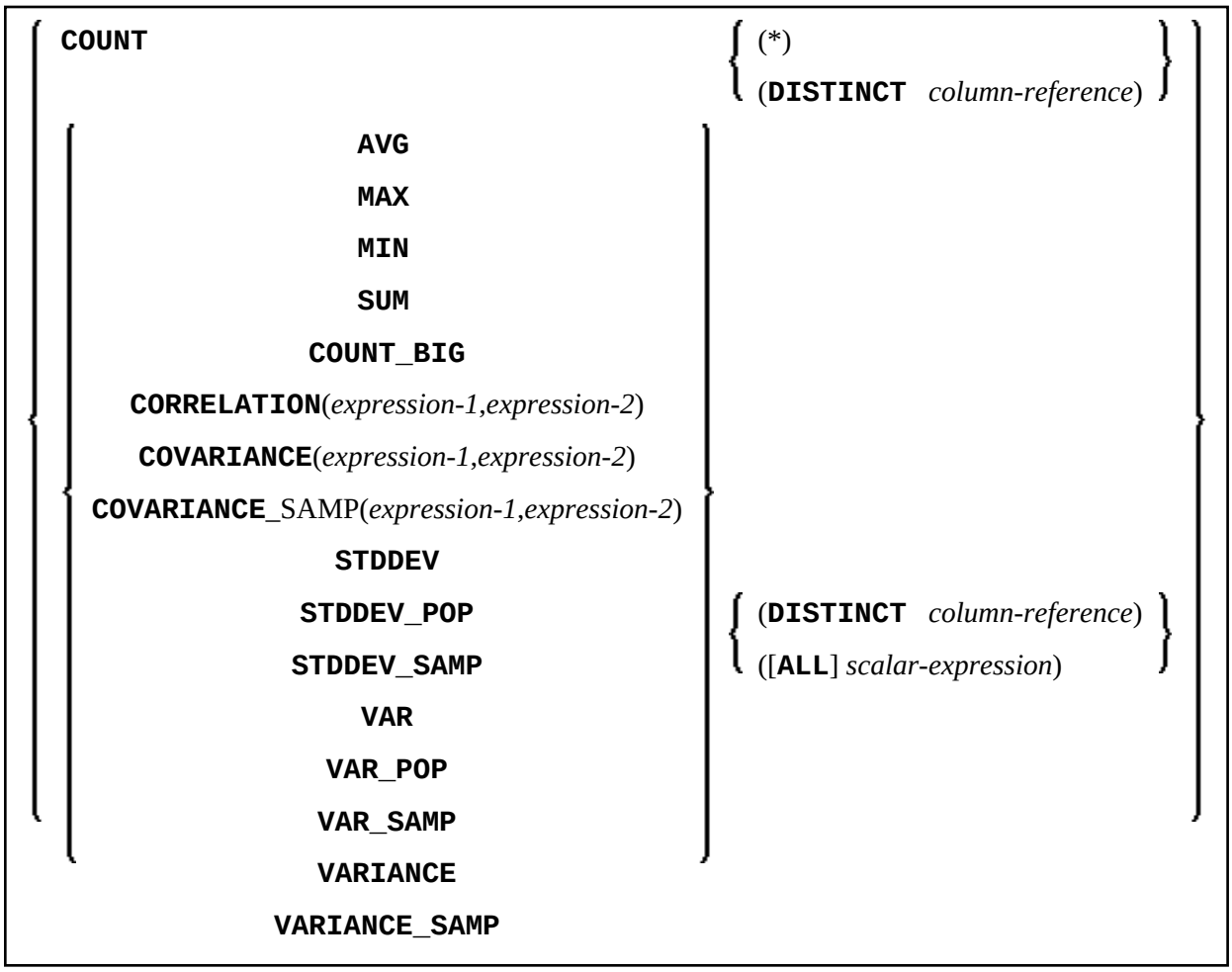
If a column is referenced by a *table-name* or *correlation-name*, it must be contained in the corresponding table. If neither a *table-name* nor a *correlation-name* is specified, the respective column must be in one of the tables specified in the FROM clause (see *Table Expression*).

Aggregate Function

Common Set Syntax:



Extended Set Syntax:



SQL provides a number of special functions to enhance its basic retrieval power. The so-called SQL aggregate functions currently available and supported by Natural are:

AVG	gives the average of the values in a column
COUNT	gives the number of values in a column
MAX	gives the highest value in a column
MIN	gives the lowest value in a column
SUM	gives the sum of the values in a column

Apart from `COUNT(*)`, each of these functions operates on the collection of scalar values in an argument (that is, a single column or a *`scalar-expression`*) and produces a scalar value as its result.

Example:

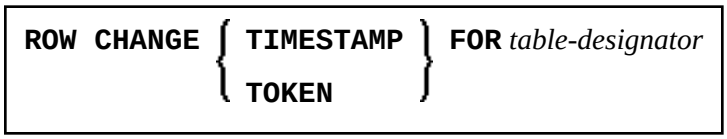
```
DEFINE DATA LOCAL
1 AVGAGE   (I2)
END-DEFINE
...
SELECT AVG (AGE)
  INTO AVGAGE
  FROM SQL-PERSONNEL
  ...
```

In general, the argument can optionally be preceded by the keyword `DISTINCT` to eliminate redundant duplicate values before the function is applied.

If `DISTINCT` is specified, the argument must be the name of a single column; if `DISTINCT` is omitted, the argument can consist of a general *scalar-expression*.

`DISTINCT` is not allowed with the special function `COUNT(*)`, which is provided to count all rows without eliminating any duplicates.

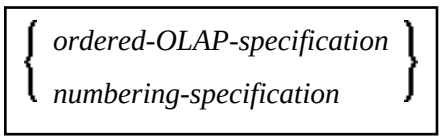
ROW CHANGE Expression



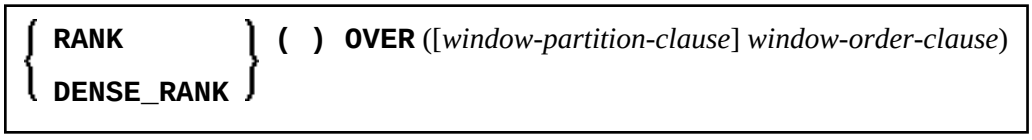
A `ROW CHANGE` expression retuns a token or a timestamp that represents the last change to a row.

`TIMESTAMP`	Specifies a timestamp is returned that represents the last time when a row was changed.
`TOKEN`	Specifies a token of type `BIGINT` is returned that represents a relative point in the modification sequence of a row.
`FOR` *`table-designator`*	Identifies the table in which the expression is referenced. *`table-designator`* has to be a valid Natural SQL DDM.

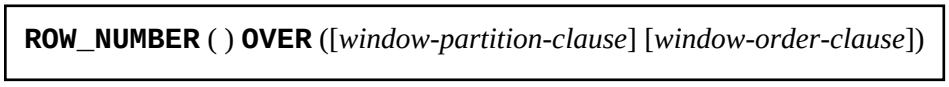
OLAP Specification



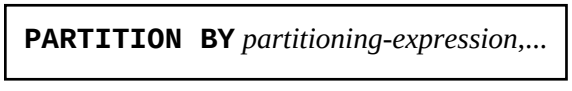
ordered-OLAP-specification



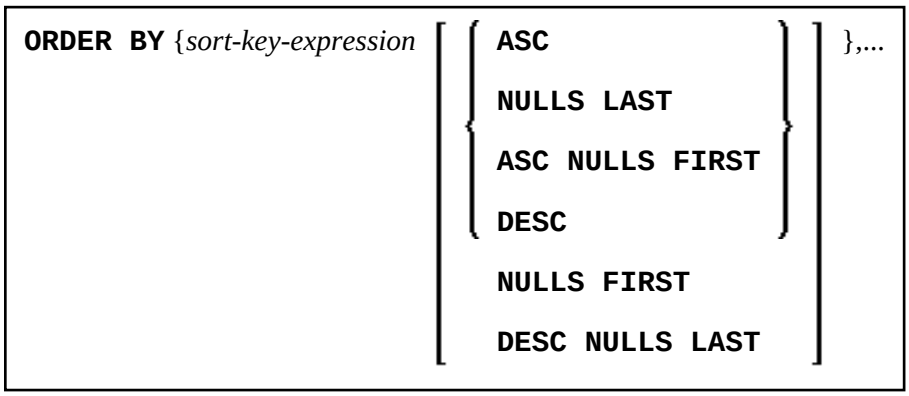
numbering-specification



window-partition-clause



window-order-clause



Online analytical processing (OLAP) specifications provide the ability to return ranking and row numbering information as a scalar value in the result of a query. An OLAP specification can be included in an expression, in a select-list, or in the `ORDER BY` clause of a `SELECT` statement. The query result to which the OLAP specifications are applied is the result table of the innermost subselect that includes the OLAP specifications.

`RANK`	Specifies that the rank of a row is defined as 1 plus the number of rows that strictly precede the row.
`DENSE_RANK`	Specifies that the rank of a row is defined as 1 plus the number of preceding rows that are distinct with respect to the ordering.
`ROW_NUMBER`	Specifies that a sequential row number is computed for the row that is defined by the ordering, starting with 1 for the first row.
`PARTITION BY`	Defines the partition within which the OLAP operation is applied.
`ORDER BY`	Defines the ordering of rows within a partition that is used to determine the value of the OLAP specification.
`ASC`	Specifies that the values of *`sort-key-expression`* are used in ascending order.
`DESC`	Specifies that the values of *`sort-key-expression`* are used in descending order.
`NULLS_FIRST`	Specifies that the window ordering considers null values before all non-null values in the sort order.
`NULLS LAST`	Specifies that the window ordering considers null values after all non-null values in the sort order.

Example:

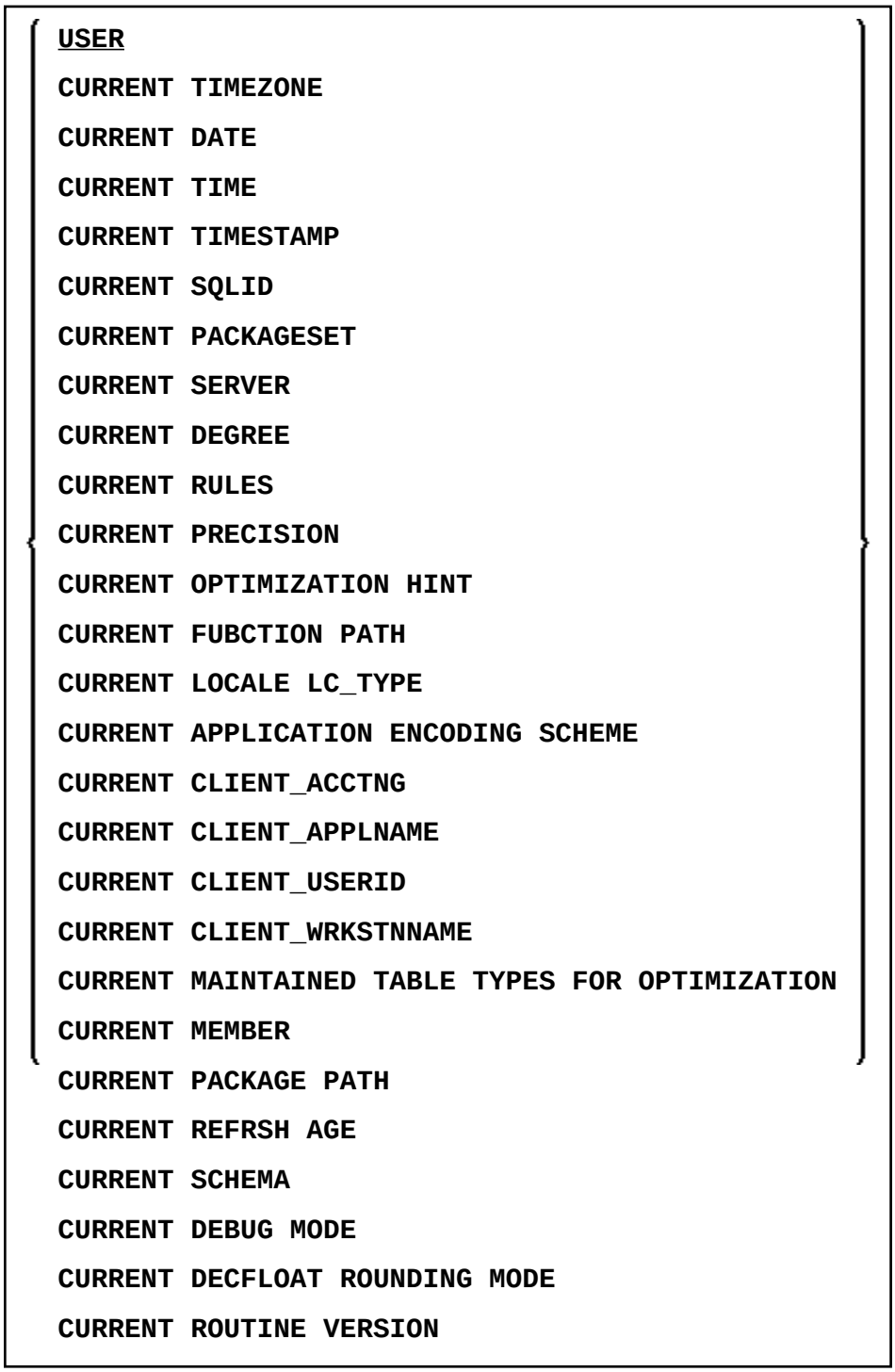
Display the ranking of employees that have a total salary of more than $30,000, in order by last name.

```
SELECT EMPNO, LASTNAME, FIRSTNME, SALARY+BONUS AS TOTAL_SALARY,
  RANK() OVER(ORDER BY SALARY+BONUS DESC) AS RANK_SALARY
  FROM DSN8910-EMP WHERE SALARY+BONUS > 30000
  ORDER BY LASTNAME;
```

Special Register

Common Set Syntax:

Extended Set Syntax:



A reference to a *special-register* returns a scalar value.

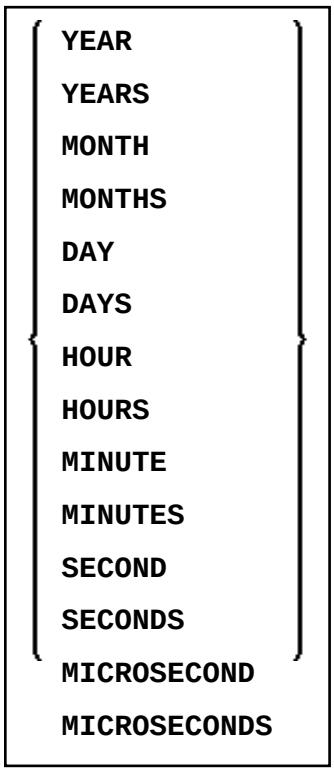
With the exception of USER, *special-registers* do not conform to standard SQL and are therefore supported by the Natural SQL Extended Set only.

Scalar Function

A *scalar-function* is a built-in function that can be used in the construction of scalar computational expressions.

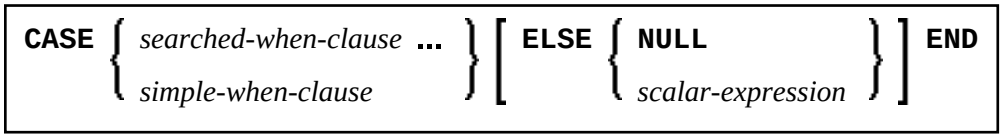
For information on the scalar-functions that are supported by the Natural SQL Extended Set, see *Syntactical Items Common to Natural SQL Statements*, *scalar-function* in the *Natural for DB2* documentation.

Scalar Expression Unit



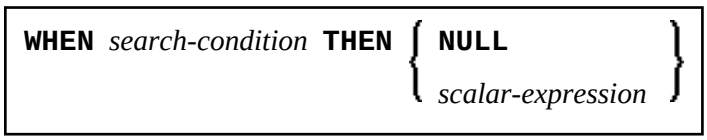
unit does not conform to standard SQL and is therefore supported by the Natural SQL Extended Set only.

Case Expression



A *case-expression* does not conform to standard SQL and is therefore supported by the Natural SQL Extended Set only.

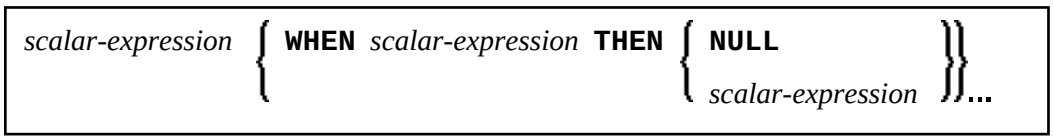
Searched WHEN Clause



A Searched `WHEN` clause does not conform to standard SQL and is therefore supported by the Natural SQL Extended Set only.

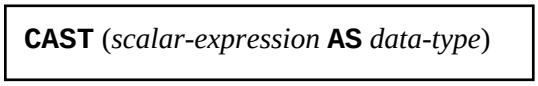
See details on *`search-condition`*.

Simple WHEN Clause



A Simple `WHEN` clause does not conform to standard SQL and is therefore supported by the Natural SQL Extended Set only.

Cast Expression

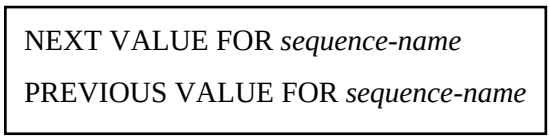


A `CAST` expression does not conform to standard SQL and is therefore supported by the Natural SQL Extended Set only.

User-Defined Function Reference

The option *`user-defined-function-reference`* belongs to the Natural SQL Extended Set. This options allows you to invoke any user-defined function. Arguments have to be placed in brackets and separated by commas. The user-defined function must be declared in the target RDBMS.

Sequence Reference



The option *`sequence-reference`* belongs to the Natural SQL Extended Set.

This option allows you to reference the next value or the previous value of a sequence object. The sequence object has to be created in the target RDBMS before it could be referenced at runtime.

Scalar Fullselect

(fullselect)

The option *scalar-fullselect* belongs to the Natural SQL Extended Set.

A *scalar-fullselect* as supported in an expression is a *fullselect* - enclosed in parentheses - that returns a single row consisting of a single column value. If the *fullselect* does not return a row, the result of the expression is the null value. If more than one row is to be returned for a *scalar-fullselect*, an error occurs.