Microsoft SQL Server SQL to Denodo VQL Quick Reference

Revision 20211103

Denodo Inc.: 525 University Avenue, Suite 31, Palo Alto, CA 94301. USA www.denodo.com

Goal

This document is a quick reference for migrating Microsoft SQL Server SQL to Denodo VQL. The document is aimed at administrators and developers that want to efficiently migrate their existing SQL Server queries to Denodo VQL queries.

Content

The Knowledge Base article VDP Conformance with Standard SQL contains a reference of the Virtual DataPort conformance with the SQL 92 standard. The document is focused on query capabilities and contains information about: Data Types, SQL Predicates Support and SQL Functions Support. The Query Expressions section lists the expressions defined by the standard SQL and their equivalent in Virtual DataPort, explaining the differences with the standard when appropriate.

Following, a group of tables where the correspondence between SQL Server functions and the Denodo equivalents is presented. This list is just a reference since more functions can be included in future Denodo versions.

Functions mapping

Denodo function	SQL Server function
`addday(daterelatedvalue, inc)`	`dateadd(day,inc,date)`
`addhour(timerelatedvalue, inc)`	`dateadd(hour,inc,date)`
`addminute(timerelatedvalue, inc)`	`dateadd(minute,inc,date)`
`addsecond(timerelatedvalue, inc)`	`dateadd(second,inc,date)`
`addmonth(daterelatedvalue, inc)`	`dateadd(month,inc,date)`
`addweek(daterelatedvalue, inc)`	`dateadd(week,inc,date)`
`addyear(daterelatedvalue, inc)`	`dateadd(year,inc,date)`
`cast(data_type, value)`	`cast(value as data_type)`
`firstdayofmonth(daterelatedvalue)`	`dateadd(day,1-datepart(day,date),date)`
`firstdayofweek(daterelatedvalue)`	`dateadd(day, 1-datepart(dw,date),date)`
`lastdayofmonth(daterelatedvalue)`	`dateadd(day, -(day(dateadd(month, 1, date))), dateadd(month, 1, date))`
`lastdayofweek(daterelatedvalue)`	`dateadd(day, 7-datepart(dw,date),date)`

nextweekday(daterelatedvalue, weekDayRef)	dateadd(day, 1 + ((7 + weekDayRef - datepart(weekday,date)) %7), date)
previousweekday(daterelatedvalue, weekDayRef)	dateadd(day, - 7 + ((8 + weekDayRef - datepart(weekday,date)) %7), date)
getmillisecond(timerelatedvalue)	datepart(millisecond,date)
getdaysbetween(daterelatedvalue1, daterelatedvalue2)	datediff(Day,date1,date2)
getmonthsbetween(daterelatedvalue1, daterelatedvalue2)	datediff(Month,date1,date2)
getdayofweek(daterelatedvalue)	datepart(weekday,date)
getdayofyear(daterelatedvalue)	datepart(dayofyear,date)
ceil(arg0)	ceiling(arg0)
concat(arg0 [,argi]*)	CAST(arg0 as varchar(8000))[+ CAST(argi as varchar(8000))]{1, n}
concat(arg0 [,argi]*)	concat(arg0[, argi]{1, n}) (SQL Server 2012+)
gethour(timerelatedvalue)	datepart(hour, date)
getminute(timerelatedvalue)	datepart(minute, date)
getsecond(timerelatedvalue)	datepart(second, date)
len(arg0)	len(arg0)
mod(arg0, arg1)	CAST(arg0 as int) % CAST(arg1 as int)
mod(arg0, arg1)	CAST(arg0 as numeric(38,10)) % CAST(arg1 as numeric(38,10))(SQL Server 2005+)
removeaccents(arg0)	replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(replace(arg0, ''Ý'', ''Y''), ''Û'', ''U''), ''Ô'', ''O''), ''Î'', ''I''), ''Ê'', ''E''), ''Â'', ''A''), ''Ü'', ''U''), ''Ö'', ''O''), ''Ï'', ''I''), ''Ë'', ''E''), ''Ä'', ''A''), ''Ù'', ''U''), ''Ò'', ''O''), ''Ì'', ''I''),

	''È'', ''E''), ''À'', ''A''), ''Ú'', ''U''), ''Ó'', ''O''), ''Í'', ''I''), ''É'', ''E''), ''Á'', ''A''), ''ý'', ''y''), ''ÿ'', ''y''), ''û'', ''u''), ''ô'', ''o''), ''î'', ''i''), ''ê'', ''e''), ''â'', ''a''), ''ü'', ''u''), ''ö'', ''o''), ''ï'', ''i''), ''ë'', ''e''), ''ä'', ''a''), ''ù'', ''u''), ''ò'', ''o''), ''ì'', ''i''), ''è'', ''e''), ''à'', ''a''), ''ú'', ''u''), ''ó'', ''o''), ''í'', ''i''), ''é'', ''e''), ''á'', ''a'')
round(arg0 [,precision])	round(arg0, PRECISSION_PARAM)
now	getDate()
substr(arg0, arg1)	substring(arg0,arg1,len(arg0)-1)
substr(arg0 from arg1)	substring(arg0,arg1,len(arg0)-1)
substr(arg0, arg1, arg2)	substring(arg0,arg1,arg2)
substr(arg0 from arg1 for arg2)	substring(arg0,arg1,arg2)
trim(arg0)	RTRIM(LTRIM(arg0))
trim(arg0)	TRIM(arg0)(SQL Server 2017+)
ltrim(arg0)	LTRIM(arg0)
rtrim(arg0)	RTRIM(arg0)
to_date(date_pattern, value)	value
atan2(arg0, arg1)	atn2(arg0,arg1)
log(arg0)	log10(arg0)BASE_CHANGE_EXPR
log(arg0, base)	log(arg0, BASE_PARAM)(SQL Server 2012+)
ln(arg0)	log(arg0)
current_date	convert(date,getdate())
xmlquery(arg0, arg1 [,xmlvalue, isxmlfile])	arg1.query(arg0)
xpath(arg0, arg1)	arg1.query(arg0)
cot(arg0)	cot(arg0)
position(arg0 IN arg1)	charindex(arg0, arg1)

repeat(arg0, count)	replicate(arg0, count)
instr(arg0, arg1)	charindex(arg1,arg0)-1

In the above list, there are different types of data and time functions:

- Functions that are related with datetime types that includes dates (date, localdate, timestamp, timestamptz), will be referred as daterelatedvalue
- Functions that are related with datetime types that includes times (date, time, timestamp, timestamptz), will be referred as timerelatedvalue
- Functions that receive any datetime values, except intervals, will be referred as datetimevalue

Aggregation functions

Denodo Function	SQL Server function
stdev(arg0)	STDEV(arg0)
stdevp(arg0)	STDEVP(arg0)
var(arg0)	VAR(arg0)
varp(arg0)	VARP(arg0)

Operators

Denodo function	SQL Server operator
arg0 is TRUE	arg0 = '1'
arg0 is FALSE	arg0 = '0'
minus	EXCEPT
xmlexists(arg0, arg1)	arg1.exist(arg0) = 1)