Example of Value Constraints Used in Queries

Value Constraint by Number

If the fact with a numerical value didn't have the normalized numerical value with a single *UNIT_CD* for a particular concept, then the user can tell the service to do the unit conversion of the *NVAL_NUM* column before applying the value constraints in the query.
The unit conversion of *NVAL_NUM* is calculated using the concept's metadata xml defined in the Ontology cell (<ConvertingUnits/>, <MultiplyingFactor/>).
To enable the unit conversion, set the following project parameter in the Project Management cell.
CRC_ENABLE_UNITCD_CONVERSION = ON | OFF

Value	Description
ON	Unit conversion is enabled
OFF	Unit conversion is not enabled

- ***This unit conversion option will slow down the query. For better query performance load, the numerical fact values in the in the normalized units and do not enable this option.***

Greater than operator	
Query Numeric Value Constraint:	<constrain_by_value> <value_operator>GT</value_operator> <value_constraint>99.9</value_constraint> <value_type>NUMBER</value_type> </constrain_by_value>
Numeric Constraint SQL:	(valtype_cd = 'N' AND nval_num > 99.9 AND tval_char IN ('GE','E')) OR (valtype_cd = 'N' AND nval_num >= 99.9 AND tval_char = 'G') ***Unit Conversion Enabled*** (valtype_cd = 'N' AND case when unit_cd = 'mg/5ml' then nval_num > 99.9 * 5 when unit_cd = 'mg/15ml' then nval_num > 99.9 * 15 when unit_cd = 'mg/0.5ml' then nval_num > 99.9 * 0.5 AND tval_char IN ('GE','E')) OR (valtype_cd = 'N' AND case when unit_cd = 'mg/5ml' then nval_num > 99.9 * 5 when unit_cd = 'mg/15ml' then nval_num > 99.9 * 15 when unit_cd = 'mg/0.5ml' then nval_num > 99.9 * 0.5 AND tval_char = 'G')
Less than operator	
Query Numeric Value Constraint:	<constrain_by_value> <value_operator>LT</value_operator> <value_constraint>99.9</value_constraint> <value_type>NUMBER</value_type> </constrain_by_value>
Numeric Constraint SQL:	(valtype_cd = 'N' AND nval_num < 99.9 AND tval_char IN ('LE','E')) OR (valtype_cd = 'N' AND nval_num <= 99.9 AND tval_char = 'L')
Between operator	
Query Numeric Value Constraint:	<constrain_by_value> <value_operator>BETWEEN</value_operator> <value_constraint>1 and 100</value_constraint> <value_type>NUMBER</value_type> </constrain_by_value>
Numeric Constraint SQL:	(valtype_cd = 'N' AND nval_num BETWEEN 1 and 100 AND tval_char = 'E')
Equal to operator	

Query Numeric Value Constraint:	<constrain_by_value> <value_operator>EQ</value_operator> <value_constraint>99.9</value_constraint> <value_type>NUMBER</value_type> </constrain_by_value>
Numeric Constraint SQL:	(valtype_cd = 'N' AND nval_num = 99.9 AND tval_char = 'E')
Less than and Equal to operator	
Query Numeric Value Constraint:	<constrain_by_value> <value_operator>LE</value_operator> <value_constraint>99.9</value_constraint> <value_type>NUMBER</value_type> </constrain_by_value>
Numeric Constraint SQL:	(valtype_cd = 'N' AND nval_num <= 99.9 AND tval_char IN ('L','E','LE')
Greater than and Equal to operator	
Query Numeric Value Constraint:	<constrain_by_value> <value_operator>GE</value_operator> <value_constraint>99.9</value_constraint> <value_type>NUMBER</value_type> </constrain_by_value>
Numeric Constraint SQL:	(valtype_cd = 'N' AND nval_num >= 99.9 AND tval_char IN ('G','E','GE')
Not Equal operator	
Query Numeric Value Constraint:	<constrain_by_value> <value_operator>NE</value_operator> <value_constraint>99.9</value_constraint> <value_type>NUMBER</value_type> </constrain_by_value>
Numeric Constraint SQL:	(valtype_cd = 'N' AND nval_num <> 99.9 AND tval_char <> 'NE') OR (valtype_cd = 'N' AND nval_num = 99.9 AND tval_char = 'NE')

Value Constraint by Text

Equals operator	
Query Text Value Constraint:	<constrain_by_value> <value_operator>EQ</value_operator> <value_constraint>H</value_constraint> <value_type>TEXT</value_type> </constrain_by_value>
Text Value Constraint SQL:	valtype_cd = 'T' AND tval_char = 'H'
Not equals operator	
Query Text Value Constraint:	<constrain_by_value> <value_operator>NE</value_operator> <value_constraint>L</value_constraint> <value_type>TEXT</value_type> </constrain_by_value>
Text Value Constraint SQL:	valtype_cd = 'T' AND tval_char <> 'L'
Like operator	
Query Text Value Constraint:	<constrain_by_value> <value_operator>LIKE</value_operator> <value_constraint>L</value_constraint> <value_type>TEXT</value_type> </constrain_by_value>
Text Value Constraint SQL:	valtype_cd = 'T' AND tval_char LIKE 'L%')

In operator	
Query Numeric Value Constraint:	<constrain_by_value> <value_operator>IN</value_operator> <value_constraint>'A','B'</value_constraint> <value_type>TEXT</value_type> </constrain_by_value>
Text Value Constraint SQL:	valtype_cd = 'T' AND tval_char = ('A','B')
Between operator	
Query Text Value Constraint:	<constrain_by_value> <value_operator>BETWEEN</value_operator> <value_constraint>'A' and 'B' </value_constraint> <value_type>TEXT</value_type> </constrain_by_value>
Text Value Constraint SQL:	valtype_cd = 'T' tval_char BETWEEN 'A' AND 'B'

Value Constraint by Flag

Equals operator	
Query Flag Value Constraint:	<constrain_by_value> <value_operator>EQ</value_operator> <value_constraint>H</value_constraint> <value_type>FLAG</value_type> </constrain_by_value>
Flag Value Constraint SQL:	valueflag_cd = 'H'
Not equals operator	
Query Flag Value Constraint:	<constrain_by_value> <value_operator>NE</value_operator> <value_constraint>L</value_constraint> <value_type>FLAG</value_type> </constrain_by_value>
Flag Value Constraint SQL:	valueflag_cd <> 'H'
In operator	
Query Flag Value Constraint:	<constrain_by_value> <value_operator>IN</value_operator> <value_constraint>'A','B' </value_constraint> <value_type>FLAG</value_type> </constrain_by_value>
Flag Value Constraint SQL:	valueflag_cd IN ('A', 'B')