SAP HANA CE Functions - Calculation Engine Plan Operators

Calculation engine plan operators encapsulate data-transformation functionality and can be used in the definition of functions. They constitute an alternative to using SQL statements as their logic is directly implemented in the calculation engine, i.e. the execution environment of SQLScript.

There are different categories of operators.

- Data Source Access operators that bind a column table or a column view to a table variable.
- Relational operators that allow a user to bypass the SQL processor during evaluation and to directly interact with the calculation engine.
- Special extensions implement, e.g., crucial business functions inside the database kernel.

Data Source Access Operators

1. **CE_COLUMN_TABLE**

   The CE_COLUMN_TABLE operator provides access to an existing column table.

   Example:

```
ot_books1 = CE_COLUMN_TABLE("BOOKS");
ot_books2 = CE_COLUMN_TABLE("BOOKS", ["TITLE","PRICE","CRCY"]);
```

   This example only works on a column table and does not invoke the SQL processor.

   It is semantically equivalent to the following:

```
ot_books3 = SELECT * FROM books;
ot_books4 = SELECT title, price, crcy FROM books;
```

2. **CE_JOIN_VIEW**

   The CE_JOIN_VIEW operator returns results for an existing join view (also known as Attribute View).

```
out = CE_JOIN_VIEW("PRODUCT_SALES", ["PRODUCT_KEY", "PRODUCT_TEXT", "SALES"]);
```

Retrieves the attributes PRODUCT_KEY, PRODUCT_TEXT, and SALES from the join view PRODUCT_SALES.

It is equivalent to the following SQL:

out = SELECT product_key, product_text, sales FROM product_sales;

3. **CE_OLAP_VIEW**

   The CE_OLAP_VIEW operator returns results for an existing OLAP view (also know as an Analytical View).

   out = CE_OLAP_VIEW("OLAP_view", ["DIM1", "KF"]);

   Is equivalent to the following SQL:

   out = select dim1, SUM(kf) FROM OLAP_view GROUP BY dim1;

4. **CE_CALC_VIEW**

   The CE_CALC_VIEW operator returns results for an existing calculation view.

   Example:
   out = CE_CALC_VIEW("_SYS_SS_CE_TESTCECTABLE_RET", ["CID", "CNAME"]);

   Semantically equivalent to the following SQL:

   out = SELECT cid, cname FROM "_SYS_SS_CE_TESTCECTABLE_RET";

Relational operators

The calculation engine plan operators presented in this section provide the functionality of relational operators which are directly executed in the calculation engine.

5. **CE_JOIN**

   The CE_JOIN operator calculates a natural (inner) join of the given pair of tables on a list of join attributes. For each pair of join attributes, only one attribute will be in the result. Optionally, a projection list of attribute names can be given to restrict the output to the given attributes. If a projection list is provided, it must include the join attributes. Finally, the

plan operator requires each pair of join attributes to have identical attribute names. In case of join attributes having different names, one of them must be renamed prior to the join.

Example:

```
ot_pubs_books1 = CE_JOIN (:lt_pubs, :it_books,["PUBLISHER"]);
ot_pubs_books2 = CE_JOIN (:lt_pubs, :it_books,["PUBLISHER"],  ["TITLE","NAME","PUBLISHER" ,"YEAR" ]);
```

This example is semantically equivalent to the following SQL but does not invoke the SQL processor.

```
ot_pubs_books3 = SELECT P.publisher AS publisher, name, street, post_code, city, country,                    isbn, title, edition,
year, price, crcy FROM :lt_pubs AS P,
                                                :it_books AS B
                                                  WHERE P.publisher = B.publisher;
ot_pubs_books4 = SELECT title, name, P.publisher AS publisher, year FROM :lt_pubs AS P, :it_books AS B WHERE
P.publisher = B.publisher;
```

6. **CE_LEFT_OUTER_JOIN**

Calculate the left outer join. Besides the function name, the syntax is the same as for CE_JOIN.

7. **CE_RIGHT_OUTER_JOIN**

Calculate the right outer join. Besides the function name, the syntax is the same as for CE_JOIN.

8. **CE_FULL_OUTER_JOIN** is not supported.

9. **CE_PROJECTION**

Restricts the columns in the schema of table variable var_table to those mentioned in the projection list.

1. A variable of type table which is subject to the projection. Like CE_JOIN, CE_PROJECTION cannot handle tables directly as input.

2. A list of attributes which should be in the resulting table. The list must at least have one element. The attributes can be renamed using the SQL keyword AS, and expressions can be evaluated using the CE_CALC function.

3. An optional filter where Boolean expressions are allowed, as defined for the CE_CALC operator below.

Example:
ot_books1 = CE_PROJECTION (:it_books,["TITLE","PRICE", "CRCY" AS "CURRENCY"], '"PRICE" > 50');

Semantically equivalent to the following SQL:

ot_books2= SELECT title, price, crcy AS currency FROM :it_books WHERE price > 50;

10. **CE_CALC**

CE_CALC is used inside other operators discussed in this section. It evaluates an expression and is usually then bound to a new column.

Example:
with_tax = CE_PROJECTION(:product, ["CID", "CNAME", "OID", "SALES", CE_CALC('"SALES" * :vat_rate', decimal(10,2)) AS "SALES_VAT"], '"CNAME" = '':cname'');

Semantically equivalent to the following SQL:

with_tax2 = SELECT cid, cname, oid, sales, sales * :vat_rate as sales_vat FROM :product WHERE cname = ':cname';

Another frequent use case of CE_CALC is computing row numbers:

```
CREATE PROCEDURE ceGetRowNum(IN it_books books,
OUT ranked_books ot_ranked_books)
LANGUAGE SQLSCRIPT READS SQL DATA AS
BEGIN
ordered_books = SELECT title, price, crcy
FROM :it_books ORDER BY price DESC;
ranked_books = CE_PROJECTION(:it_books, ["TITLE", "PRICE",
CE_CALC('rownum()', integer) AS "RANK",
    "CRCY" AS "CURRENCY"]);
    END;
```

11. **CE_AGGREGATION**

Groups the input and computes aggregates for each group.

Example:
ot_books1 = CE_AGGREGATION (:it_books, [COUNT ("PUBLISHER") AS "CNT"], ["YEAR"]);
Semantically equivalent to the following SQL:
ot_books2 = SELECT COUNT (publisher) AS cnt, year FROM :it_books GROUP BY year;

12. **CE_UNION_ALL**

The CE_UNION_ALL function is semantically equivalent to SQL UNION ALL statement. It computes the union of two tables which need to have identical schemas. The CE_UNION_ALL function preserves duplicates

Example:
ot_all_books1 = CE_UNION_ALL (:lt_books, :it_audiobooks);
Semantically equivalent to the following SQL:
ot_all_books2 = SELECT * FROM :lt_books UNION ALL SELECT * FROM :it_audiobooks;

Special Operators

In this section we discuss operators that have no immediate counterpart in SQL.

13. **CE_VERTICAL_UNION**

For each input, applies the concatenation of their columns. Optionally columns can be renamed. Clearly, all input tables must have the same cardinality.
Syntax:
CE_VERTICAL_UNION(:input1, [project_att1 {AS new_param_name}, ...], :input2, [project_att1 {AS new_param_name}, ...], ...)

Example:
out = CE_VERTICAL_UNION(:firstname, ["ID", "FIRSTNAME" AS "GIVENNAME"], :lastname, ["LASTNAME" AS "FAMILYNAME"]);

The vertical union is sensitive to the ordering of its input. SQL statements and many calculation engine plan operators may reorder their input or return their result in different orders across invocations. This may lead to unexpected results.

14. **CE_CONVERSION**

Applies a unit conversion to input table 'input1' and returns the converted values. Result columns can optionally be renamed.

Key	Values	Type	Mandatory	Default	Documentation
'family'	'currency'	key	Y	none	the family of the conversion to be used
'method'	'ERP'	key	Y	none	the conversion method
‘error_handling’	'fail on error', 'set to null', 'keep unconverted'	key	N	'fail on error'	The reaction if a rate could not be determined for a row
'output'	combinations of 'input', 'unconverted', 'converted', 'passed_throug h', 'output_unit', 'source_unit',	key	N	'converted, passed_through , output_unit'	which attributes should be included in the output

Key	Values	Type	Mandatory	Default	Documentation
'target_unit', 'reference_date'					
'source_unit'	Any	Constant	N	None	the default source unit for any kind of conversion
'target_unit'	Any	Constant	N	None	the default target unit for any kind of conversion
'reference_date'	Any	Constant	N	None	the default reference date

					for any kind of conversion
'source_unit_column'	column in input table	column name	N	None	the name of the column containing the source unit in the input table
'target_unit_column'	column in input table	column name	N	None	the name of the column containing the target unit in the input table
'reference_date_column'	column in input table	column name	N	None	the default reference date for any kind of conversion
'output_unit_column'	Any	column name	N	"OUTPUT_UNIT"	the name of the column containing the target unit in the output table

And for ERP-Conversion in particular also:

Key	Values	Type	Mandatory	Default	Documentation
'client'	Any	Constant	Y	None	the client as stored in the tables
'conversion_type'	Any	Constant	N	'M'	the conversion type as stored in the tables
'schema'	Any	schema name	N	current schema	the default

schema in which the conversion tables should be looked up

Example:
conv_tab = CE_CONVERSION(:input, [family = 'currency', method = 'ERP', client = '004', conversion_type = 'M', target_unit = 'EUR', source_unit_column = "WAERK", reference_date_column = "ERDAT", output_unit_column = "TRGCUR"]);

15. **TRACE**

The TRACE operator is used to debug SQLScript procedures. It traces the tabular data passed as its argument into a local temporary table and returns its input unmodified. The names of the temporary tables can be retrieved from the SYS.SQLSCRIPT_TRACE view.

out = TRACE(:input);

Posted 6th November 2012 by JD

17 View comments

Anonymous January 24, 2014 at 1:48 AM
nice post thank you
Reply

yektek January 26, 2014 at 11:54 PM
valuable information from you thanks a lot swathi
Reply

Anonymous January 28, 2014 at 11:55 PM
thanks for sharing this valuable information t bindu

Reply

Anonymous February 13, 2014 at 10:16 PM

very useful information thanks

Reply

yektek.com March 12, 2014 at 2:49 AM

Very much precious information thank you Latha Pallavi

Reply

Anonymous June 4, 2014 at 2:09 AM

very well articulated. Thanks a lot..

Reply

Anonymous August 25, 2015 at 11:45 PM

Nice post!
Thanks for sharing this information.
Post more articles like this.....
Hamsini Technologies predominantly gives preference for every student's future. For that it was established with the industry experts to provide best and 100% job oriented Online Trainings around the globe in their convenience. We give SAP HANA Online Training by highly experienced market experts. Our all faculty have almost 10+ years real time experience and 4-5 years live online training experience. You will get life time support with one time payment.
Email Id: info@hamsinitechnologies.com,
Phone: +91 9666733320.
SAP HANA Online Training in Hyderabad

Reply

jaya February 28, 2016 at 4:19 AM

Very useful post.
SAP HANA Online training in Hyderabad

Reply

Anonymous March 8, 2016 at 2:22 AM

Thanks for posting this valuable information. It is useful to beginners for SAP ..... SAP HANA Online Training | SAP online Training

Reply

Anonymous July 24, 2016 at 11:41 AM

Nice article.. What is the meaning of "operator returns results for an existing **** view"? It creates a completely new view when we use "CE_****_View", doesnt it?

Reply

Anonymous July 25, 2017 at 12:08 AM

Nice Article on HANA. Useful Information about SAP HANA Implementation and Services.
Do you want free demo on SAP HANA Implementation, then please contact SAP HANA Services

Reply

Anonymous September 9, 2017 at 3:34 AM

Hi,
Actually I am fresher for sap hana with this post i got some basic idea Thank you(https://tekslate.com/sap-hana-training/)

Reply

Anonymous January 31, 2018 at 8:38 PM

Really I like your post.thank you for sharing.
SAP HANA Online Access

Reply

Anonymous March 22, 2018 at 2:42 AM

Really I like your post.thank you for sharing.
SAP HANA Online Access

Reply

---

narayana June 30, 2018 at 3:46 AM

Thanks for sharing amazing information about SAP.Gain the knowledge and hands-on experience in Sap Online Access

Reply

---

un known November 10, 2019 at 9:47 PM

I just loved your article on the beginners guide to starting a blog.If somebody take this blog article seriously in their life, he/she can earn his living by doing blogging.thank you for thizs article. best sap hana online training

Reply

IICT Technologies March 20, 2020 at 2:48 AM

Wonderful Information have been Shared...

SAP Training in Chennai

SAP ABAP Training in Chennai

SAP BASIS Training in Chennai

SAP FICO Training in Chennai

SAP PP Training in Chennai

SAP PM Training in Chennai

SAP QM Training in Chennai

SAP SD Training in Chennai

SAP MM Training in Chennai

SAP MDG Training in Chennai

Reply

Enter your comment...

Comment as: Sathish B (Goc ▼

Sign out

Publish Preview

Notify me