DATA VIRTUALITY MASTERCLASS

Topic: Applying community solutions

What to expect from this session?

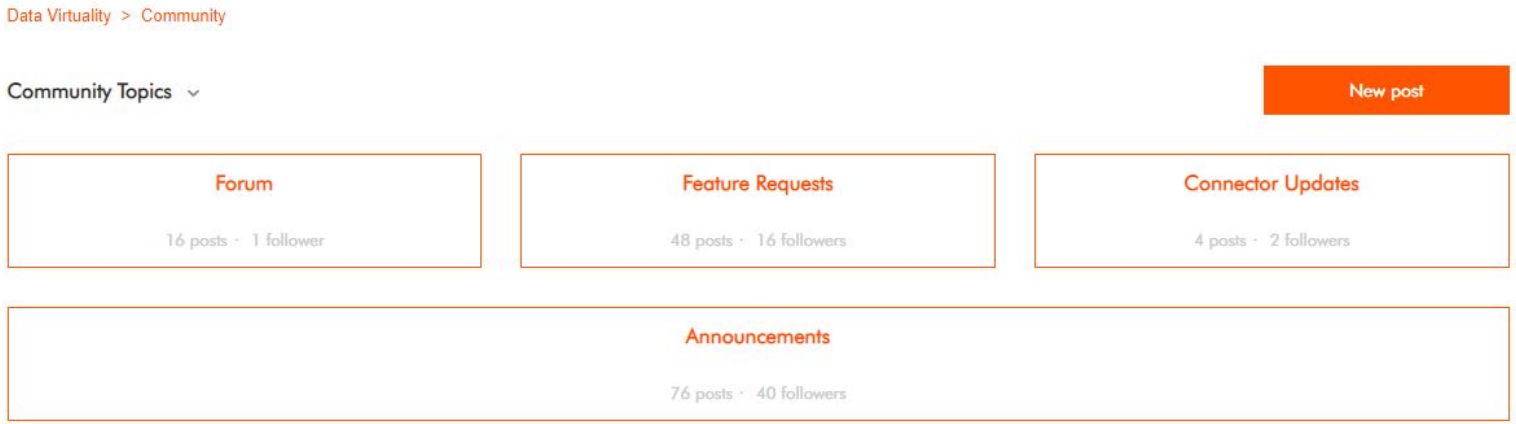
There are many available resources on our community website. We realized that they are not known very well to our community. This track will take a look at some of our solutions and show their application in practical examples. The following Community solutions will be covered:

- XML multi-values
- Using the Upsert Stored Procedure to Reproduce the SQL MERGE
- Key Value Pairs as Stored Procedure Parameters
- Regex Replace Stored Procedure
- Application of sprocs in UTILS
- Moving tables across servers
- Date axis
- Drop old MAT tables
- Find Recopts without a schedule
- Optional Joins

Some of the examples did not appear in the community, but in support requests. We want to share these solutions as well in the course of this masterclass.

Some remarks on our community page

- Community and Help Center
- Community is for announcements, and user posts
- Help Center articles created by DV staff
- We encourage you to post your own interesting solutions in the forum
- Current sections:
  - Forum - share cool user stories here
  - Feature requests - you can request new features here
  - Connector updates - subscribe here to see changes for REST API connectors
  - Announcements - subscribe here to get information about new releases and important announcements from the DV side



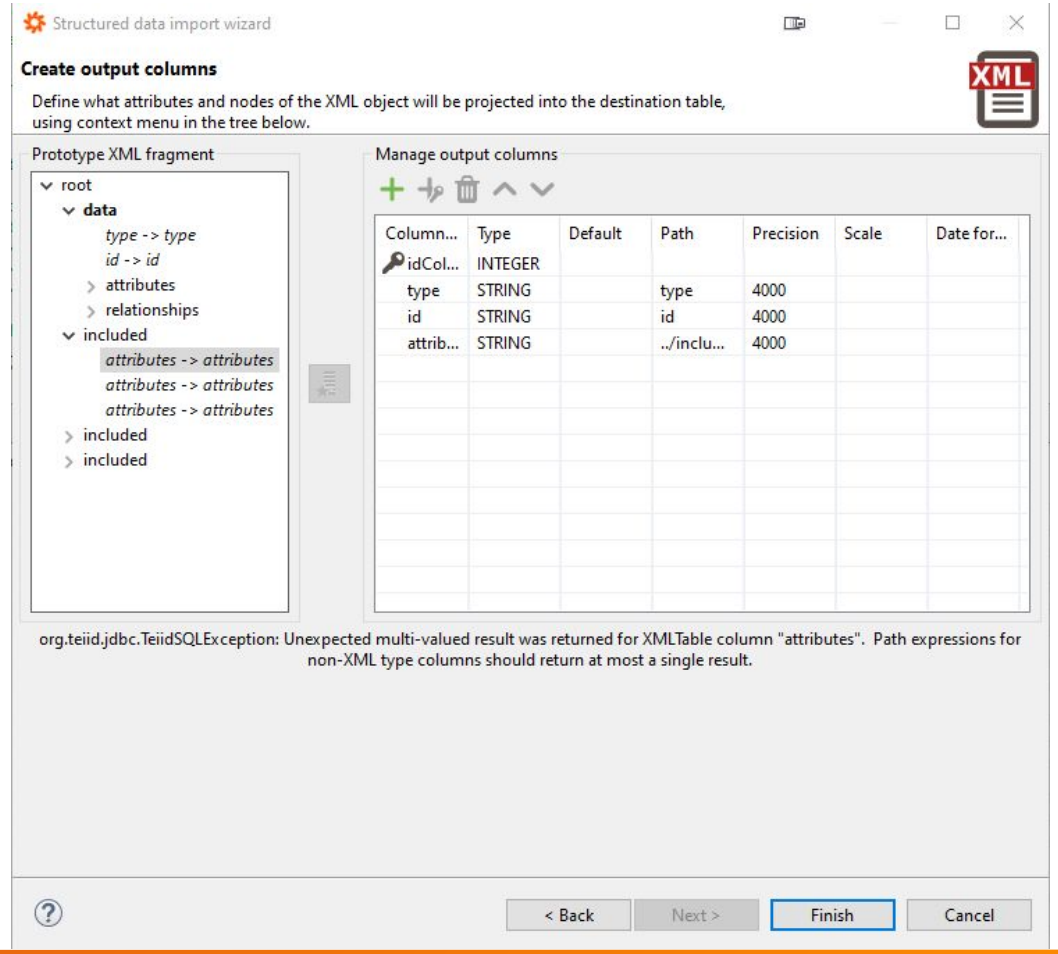
data virtuality

XML multi-values

XML multi values

- Community link:
  https://support.datavirtuality.com/hc/en-us/articles/210854723-handle-multi-valued-XML-paths
- Problem: when using the DV XML/JSON wizard, an error message will appear when dragging in a field with this structure:

```
<included>
      <attributes>a</attributes>
      <attributes>b</attributes>
      <attributes>c</attributes>
</included>
```



- Solutions:
  - indexed path expression
  - multiple XMLTABLE parsing
  - array approach
- DEMO

data virtuality

Using the Upsert Stored Procedure to Reproduce the SQL MERGE

Using Upsert to Reproduce the SQL MERGE - problem

- Community Link: https://support.datavirtuality.com/hc/en-us/articles/360014175311-Using-the-Upsert-Stored-Procedure-to-Reproduce-the-SQL-MERGE
- Problem: SQL Merge is not supported in Data Virtuality

```
merge into "dwh.fact_soc_tt_posts" dest
using
(select * from "dwh.staging_fact_tt_posts" ) src
on (src.post_id = dest.post_id)
when matched then update set
 dest.createtime    = src.createtime
;;
```

Using Upsert to Reproduce the SQL MERGE - solution

data virtuality

- Solution: use the upsert procedure
- Upsert has developed - it is possible to upsert to any writable data source, not only dwh - this is true for all replication jobs, now also in the wizards
- When creating a job, the definition will be an upsert procedure call
- Let’s review the parameters:

```
exec UTILS.upsert (
source_table => '"mssql.AdventureWorks.HumanResources.Employee"'
,keyColumnsArray => array ('EmployeeID')
,updateColumns => array ('EmployeeID')
,invertUpdateColumns => true
,target_table => '"oracle.employee"'
,surrogateKeyType => 'UUID'
,surrogateKeyName => 'SurrogateUUID'
,dbmsTableCreationOptions => NULL
,checkMaxField => 'ModifiedDate'
,defaultvalueIfCheckMaxFieldIsNull => '''2000-01-01'''
);;
```

data virtuality

Key Value Pairs as Stored Procedure Parameters

Key Value Pairs as Stored Procedure Parameters

- Community Link: https://support.datavirtuality.com/hc/en-us/articles/210180266-Pass-Key-Value-Pairs-to-a-Procedure-Using-Multi-Dimensional-Arrays
- Problem: usually, you have to provide named parameters for procedure calls - this might be limiting in some scenarios
- Solution: stored procedure to parse the parameters

Key Value Pairs as Stored Procedure Parameters - code

- Parameters are put into a temp table
- Procedure can also be used to get the value for a key

```
CREATE VIRTUAL PROCEDURE views.parse_params (
    IN params object,IN "key" string
) RETURNS (
    "argument_number" integer, "key" string, "value" object)
    AS
BEGIN
    DECLARE integer VARIABLES.i = 0 ;
    CREATE LOCAL TEMPORARY TABLE "#__LOCAL__keyvalue_store" (
        "argument_number" integer, "key" string, "value" object) ;
    WHILE (
        i < array_length (params)
    )
    BEGIN
        INSERT
            INTO "#__LOCAL__keyvalue_store" SELECT
                    VARIABLES.i + 1 as "argument_number"
                    ,cast (
                        params[i + 1 ][1] as string
                    ) AS "key"
                    ,params[i + 1 ][2] as "value" ;
        VARIABLES.i = VARIABLES.i + 1 ;
    END
    IF ("key" IS NOT NULL)
          SELECT * from "#__LOCAL__keyvalue_store" WHERE "key" = "parse_params.key" ;
    ELSE
          SELECT * from "#__LOCAL__keyvalue_store" ;
END;;
```

Key Value Pairs as Stored Procedure Parameters - examples

- Examples:

```
call views.parse_params (
    ARRAY (
        ARRAY ('master','class')
        ,ARRAY ('stored','procedure')
    )
);;
```

```
call "views.parse_params" (
    "params" =>     ARRAY (
        ARRAY ('master','class')
        ,ARRAY ('stored','procedure')
    )
    ,"key" => 'master'
);;
```

Results

	argument_number	key	value
1	1	master	class
2	2	stored	procedure

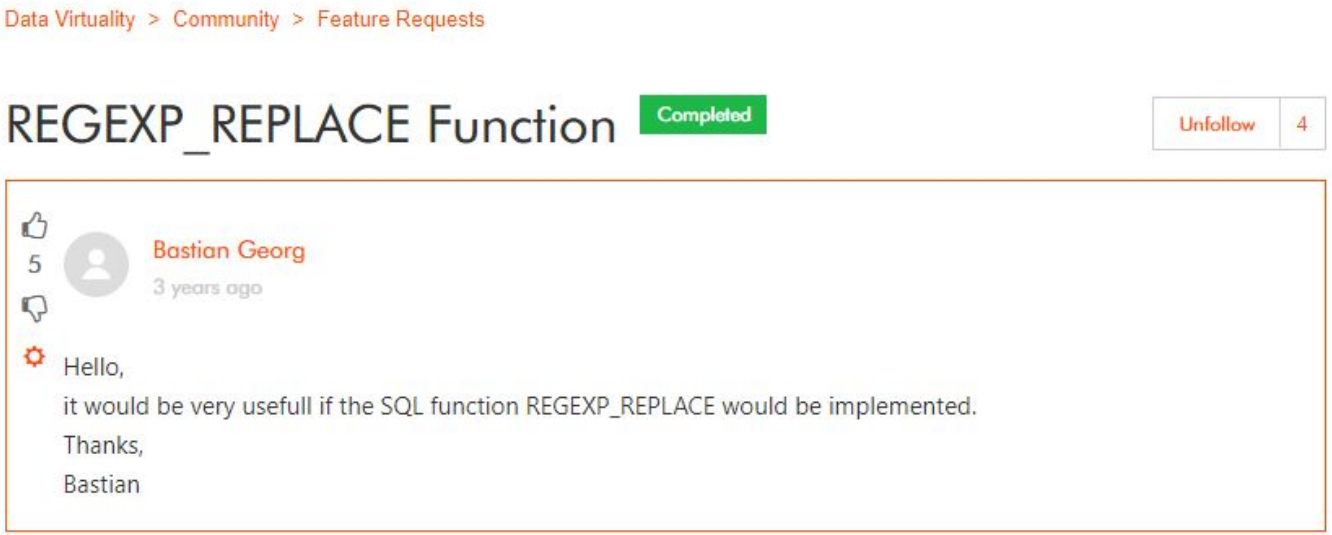
Results

	argument_number	key	value
1	1	master	class

Regex Replace Stored Procedure

Regex Replace Stored Procedure

- Problem: REGEXP_REPLACE did not exist as a stored procedure in 2.1
- Solution: Feature Request in the community
- https://support.datavirtuality.com/hc/en-us/articles/360001170443-Regex-Replace-via-ObjectTable



Regex Replace Stored Procedure - objecttable solution

- Let us still have a look at the community solution
- It shows a demo of using javascript with ObjectTable

```
Create Virtual Procedure demos.RegexReplace (
      IN initialString string not null,
      IN regex string not null,
      IN replacement string not null
      )
Returns (resultString string)
As
Begin
      Select
            resultString
      From
            ObjectTable (language 'javascript' '
                  (new java.lang.String(initialString)).replaceAll(regex, replacement)
                  'Passing
                  initialString as initialString,
                  regex as regex,
                  replacement as replacement
                  Columns
                  resultString string 'dv_row'
            ) o;
End ;;
```

Regex Replace Stored Procedure - usage and outlook

- Usage:

```
select
      k.Name,
      (call views.RegexReplace(
                "initialString" => k.Name,
                "regex" => '[aeiouAEIOU]',
                "replacement" => '**'
      )) as regex_replacement
from
      SYS.DataTypes AS k ;;
```

Results

	Name	regex_replacement
1	ENTITIES	**NT**T****S
2	ENTITY	**NT**TY
3	ID	**D
4	IDREF	**DR**F
5	IDREFS	**DR**FS
6	NCName	NCN**m**
7	NMTOKEN	NMT**K**N
8	NMTOKENS	NMT**K**NS
9	NOTATION	N**T**T****N
10	Name	N**m**
11	QName	QN**m**
12	XMLLiteral	XMLL**t**r**l
13	anyURI	**ny**R**
14	base64Binary	b**s**64B**n**ry
15	bigdecimal	b**gd**c**m**l
16	biginteger	b**g**nt**g**r

- This opens up a huge set of possibilities

  https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference

- Be aware, performance will be taken from the DV Server, as this operation will not be pushed down

data virtuality

Application of sprocs in UTILS

Application of sprocs in UTILS

- Problem: not all UTILS functions are known to our community, documented here: https://documentation.datavirtuality.com/24/reference-guide/system-schema/system-procedures/utils-procedures
- It’s worth checking the documentation

Application of sprocs in UTILS - examples (1)

- Get a table from an array:

```
Select * From (
    call UTILS.arrayToTable(
        items => ('a','b','c')
    )
)x;;
```

Results

	id	item
1	1	a
2	2	b
3	3	c

- Create a table, as DV does not allow the CREATE TABLE statement

```
call UTILS.createTable (
        tableName => 'dwh.TableName',
        columnsAndTypes => 'stringColumn,intColumn|integer,decimalColumn|decimal,timestampColumn|timestamp,_crazyName'
    )
```

Columns: 5

Name	Type	Size	Scale	Nullable
stringcolumn	string	4000	0	NULL
intcolumn	integer	4	0	NULL
decimalcolumn	bigdecimal	2147483647	20	NULL
timestampcolumn	timestamp	12	0	NULL
_crazyname	string	4000	0	NULL

Application of sprocs in UTILS - examples (2)

- Convert a table to JSON (XML version also available):

```
call "UTILS.tableToJson"(
    "tableName" => '"MySQL.salesorderdetailregular"'
);;
```

Results

	json
1	{"salesorderdetailregular":{"row":[{"salesorderid":5001,"linenumber":1,"productid":121

- Avoid casting problems, UTILS.tryCastDate also available

```
call UTILS.tryCast(
    originalValue => 'abc',
    targetType => 'decimal'
);;
```

Results

	newValue
1	<null>

data virtuality

Moving tables across servers

Moving tables across servers

- Problem: when changing to a different Analytical Storage or if you just want to move data, data needs to be copied.
- Solutions:
  - if the target system is the same database type (e.g. setting up a second server for DV Sync), use the databases capabilities instead of DV
  - if the tables are not too large you can loop on the metadata and SELECT INTO
  - if the tables are large, do the loading in chunks

Moving tables across servers - solution

- loop on the metadata and SELECT INTO and use EXECUTE IMMEDIATE to execute the SELECT INTO dynamically
- Can also be used for backup purposes: https://support.datavirtuality.com/hc/en-us/articles/201445825-Backup-the-Data-Warehouse-by-Copying-It-to-a-Parallel-Instance

```
BEGIN
    LOOP
    ON (SELECT * FROM "SYS.Tables" WHERE SchemaName = 'dwh') AS dwh_table
    BEGIN
        EXECUTE IMMEDIATE 'SELECT * INTO dwh_new."' || dwh_table."Name" || '" FROM
dwh."' || dwh_table."Name" || '"' ;
    END
END;;
```

Moving tables across servers - large dataset example

- Doing it for huge tables

```
DECLARE BIGINTEGER jobId123 ;
  jobId123 = EXEC SYSADMIN.CreateSQLJob(script => 'begin
  declare integer numruns=40 ;
  declare integer i =0;
  while (i<numruns)
  begin atomic
  INSERT into dwh.compass_owi_trans
  SELECT * FROM "export_views.compass_owi_trans"
  where owi_transid
  >(select coalesce(max(owi_transid),-1) from dwh.compass_owi_trans ) order by owi_transid limit 1000000;
  i=i+1;
  end
  end', description => 'OneOff chunked: export_views.compass_owi_trans') ;
```

data virtuality

Date axis

Date Axis

- Problem: DWH developers are used to having date axis tables
- Solution: we do not need static tables here, we can do it in a procedure
- Example:

```
CREATE virtual procedure views.dateaxis (
    IN startdate date
    ,IN enddate date
) returns (
    xdate date
) as
begin
    DECLARE date idate ;
    idate = startdate ;
    CREATE LOCAL TEMPORARY TABLE #x (
        xdate date
    ) ;
    WHILE (
        idate <= enddate
    )
    BEGIN
        INSERT
            INTO #x (xdate)
            VALUES (idate) ;
        idate = timestampadd (
            SQL_TSI_DAY
            ,1
            ,idate
        ) ;
    END
    SELECT
            *
        from
            #x ;
end;;
```

```
call "views.dateaxis"(
    "startdate" => '2021-01-01',
    "enddate" => curdate()
);;
```

Results

	xdate
1	2021-01-01
2	2021-01-02
3	2021-01-03
4	2021-01-04
5	2021-01-05
6	2021-01-06
7	2021-01-07
8	2021-01-08
9	2021-01-09
10	2021-01-10
11	2021-01-11

data virtuality

Drop old MAT tables

Drop old MAT tables

- Community Link: https://support.datavirtuality.com/hc/en-us/articles/201446295-Get-the-Recent-N-Stages-of-Mat-Tables-and-Create-a-Procedure-to-Drop-Old-Ones
- Problem: while the **Clean stale replicator tables task** job works very well and can be configured to *keep n stages* and *keep n days*, users might still want to have fine grained control over it.
- Solution: list the latest stages and create a procedure to drop stages before it

Drop old MAT tables

- Challenge we are solving here is to get the last n objects by order
- This is not a problem for a single mat table (example: mat_table_25_st1..20)
- Let us use a window function here

```
SELECT a.*

FROM (SELECT "t.Name" as "TableName", "mt.accessState",
SUBSTRING("t.Name",0,LOCATE('_st',"t.Name")-1)as "TablePrefix",
CAST(SUBSTRING("t.Name",11,LOCATE('_st',"t.Name")-11) as integer) as "TableNumberInfix",

CAST(SUBSTRING("t.Name",LOCATE('_st',"t.Name")+3) as integer)as "MatTableStage"RANK() OVER
(PARTITION BY SUBSTRING("t.Name",0,LOCATE('_st',"t.Name")-1) ORDER BY
CAST(SUBSTRING("t.Name",LOCATE('_st',"t.Name")+3) as integer) DESC) as "StagePriority"

FROM  "SYS.Tables" t INNER JOIN "SYSADMIN.MaterializedTable" mt ON("mt.name" = "t.name")

WHERE "t.SchemaName" = 'dwh' AND "mt.accessState" = 'READY'

ORDER BY CAST(SUBSTRING("t.Name",11,LOCATE('_st',"t.Name")-11) as integer)) as a

WHERE "a.StagePriority" <= 10

ORDER BY "a.TableNumberInfix", "a.StagePriority" ASC

;;
```

Results

	TableName	accessState	TablePrefix	TableNumberInfix	MatTableStage	StagePriority
1	mat_table_0_st0	READY	mat_table_0	0	0	2
2	mat_table_0_st1	READY	mat_table_0	0	1	1
3	mat_table_1_st17	READY	mat_table_1	1	17	10
4	mat_table_1_st18	READY	mat_table_1	1	18	9
5	mat_table_1_st19	READY	mat_table_1	1	19	8
6	mat_table_1_st20	READY	mat_table_1	1	20	7
7	mat_table_1_st21	READY	mat_table_1	1	21	6
8	mat_table_1_st22	READY	mat_table_1	1	22	5
9	mat_table_1_st23	READY	mat_table_1	1	23	4
10	mat_table_1_st24	READY	mat_table_1	1	24	3

Find Recopts without a schedule

Find Recopts without a schedule

- Community Link: https://support.datavirtuality.com/hc/en-us/articles/202892369-Find-Enabled-Recommended-Optimizations-Which-Have-No-Schedule
- Problem: there might be enabled recommended optimizations without refresh schedule
- You will get old data without even noticing on bigger systems
- Solution: use DV's metadata to identify them

Find Recopts without a schedule - code

data virtuality

```
SELECT ro.id
  ,ro.Matchdescriptor
  ,sj.description
  ,sd.*
FROM
  "SYSADMIN.RecommendedOptimizations" ro
  left join "SYSADMIN.ScheduleJobs" sj
  on ro.id = sj.groupId
  left join "SYSADMIN.Schedules" sd
  on sj.ID = sd.jobID
WHERE
  "ro"."Enabled" = true
  and sd.ID IS NULL;;
```

data virtuality

Optional Joins

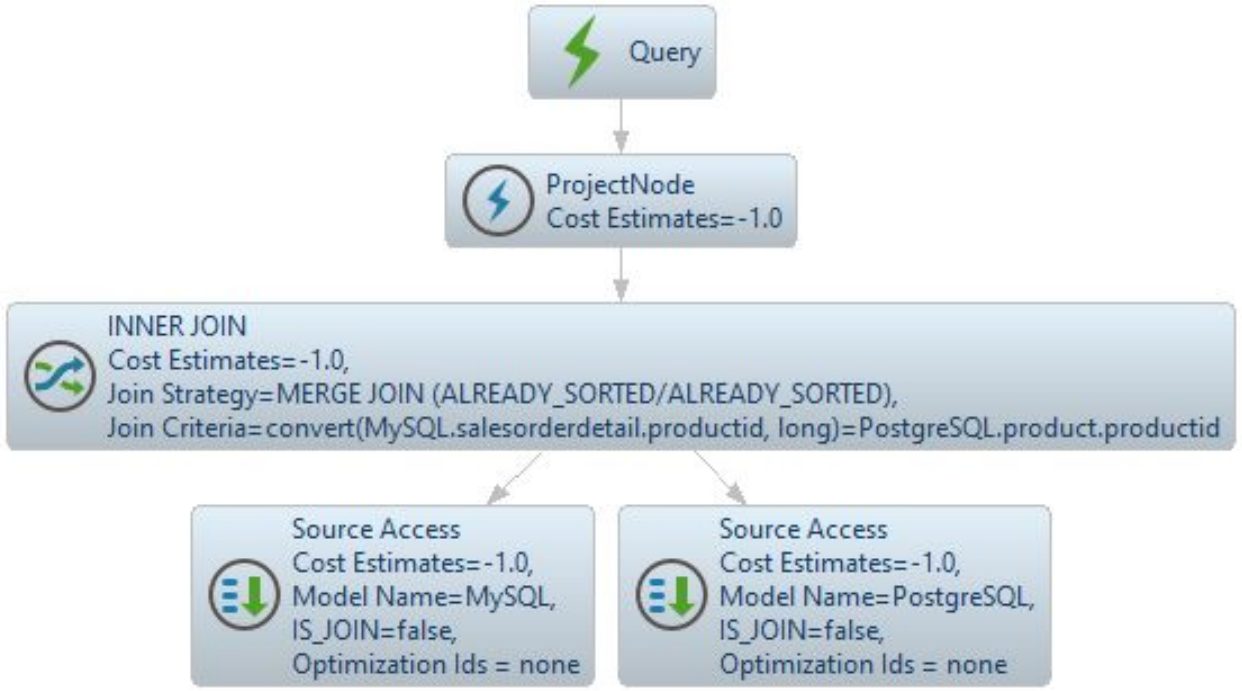
Optional Joins

- Link: https://documentation.datavirtuality.com/24/reference-guide/federated-planning/federated-optimizations
- Problem:
  - View consists of A JOIN B
  - User queries fields of A only
  - JOIN to B is still executed
- Solution:
  - Declare the join as optional

Optional Joins - example with fields from both tables

- Fields from both tables are queried - Execution plan as expected

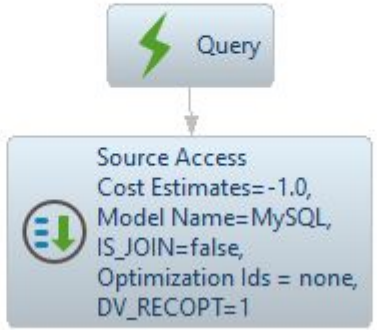
```
SELECT
    "salesorderdetail.orderqty",
    "salesorderdetail.linetotal",
    "product.name"
FROM
    "MySQL.salesorderdetail" INNER JOIN /*+ optional */ "PostgreSQL.product" ON "salesorderdetail.productid" = "product.productid";;
```



Optional Joins - example with querying only one table

- Fields from one table queried - Execution plan changes

```
SELECT
    "salesorderdetail.orderqty",
    "salesorderdetail.linetotal"
FROM
    "MySQL.salesorderdetail" INNER JOIN /*+ optional */ "PostgreSQL.product" ON "salesorderdetail.productid" = "product.productid"
```



Summary

- Our community and help center offer a variety of solutions, but they are often overlooked
- Subscribe to topics
- SHARE - if you create a nice solution, let the world know in our community forum

Any feedback / questions?

Thank you!

Please feel free to contact us at:
presales@datavirtuality.com

or

visit us at:
datavirtuality.com