



SAS® GLOBAL FORUM 2016



IMAGINE. CREATE. INNOVATE.

From SAS® Data Management to Big Data Appliances

How SAS/ACCESS® Makes Life Easier

#SASGF



From SAS® Data Management to Big Data Appliances: How SAS/ACCESS® Makes Life Easier

Magali Thésias
Senior Specialist - Deloitte

Context

90% of all the data
in the 
has been generated
over the last **2 years**

New technologies are developed to help companies handle this huge volume of data. Amongst them, the Massively Parallel Processing (MPP), enabling splitting data and queries across a large number of nodes in order to perform simultaneous computation.

What is the issue?

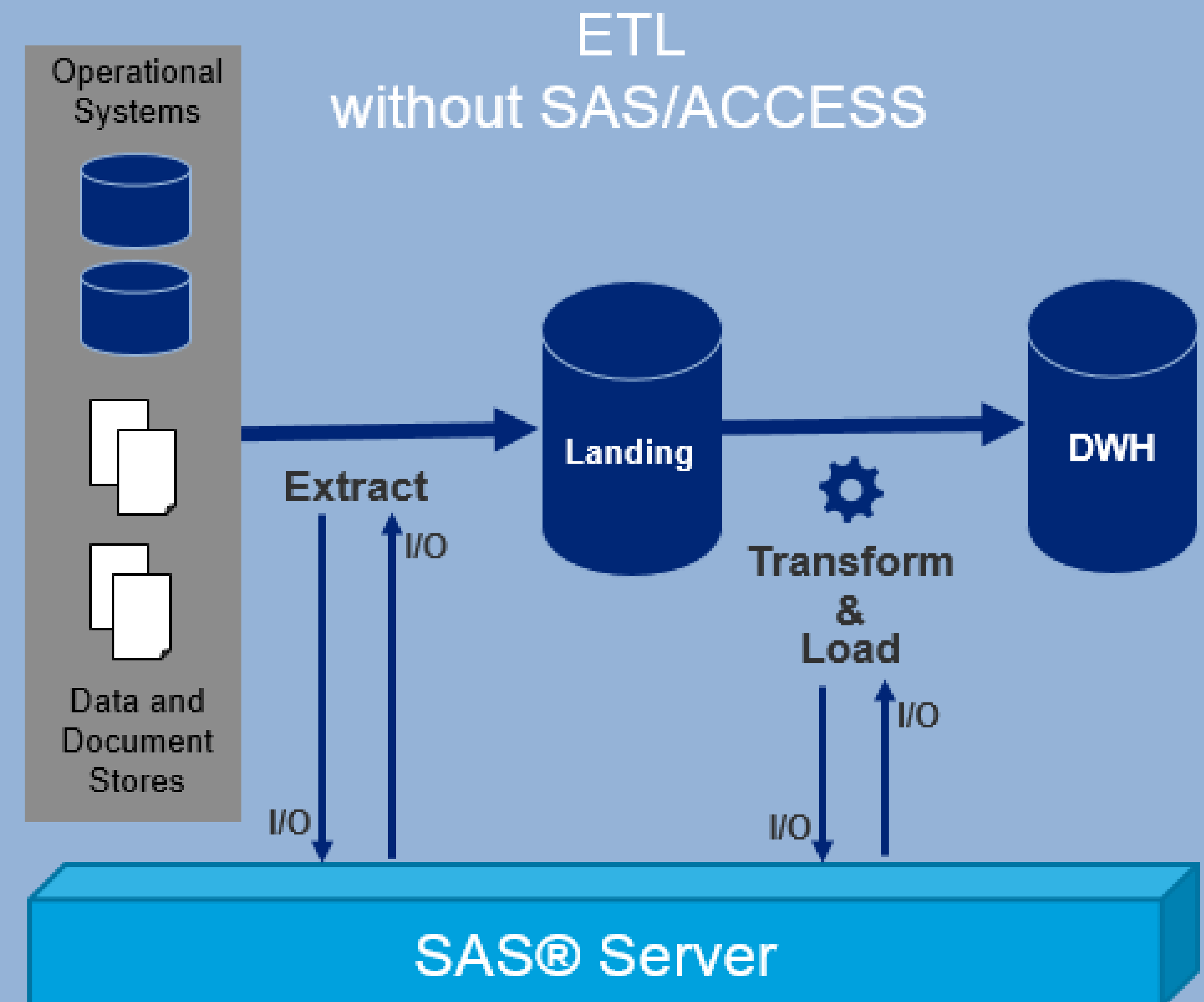
Many (large) companies are struggling in dealing with these technologies and more importantly on integrating them in their existing data management processes.

Moreover, they also want to rely on the knowledge built by their teams in existing products and implementing change to learn new technologies can therefore be a costly procedure.

How SAS is helping?

SAS® is perfectly fitting in this situation by offering a suite of software that can be set up to work with any third-party database through the usage of the corresponding SAS/ACCESS®. Indeed, for every new database technology SAS® is releasing a specific SAS/ACCESS® allowing users to develop and migrate SAS® solutions almost transparently. Only few techniques have to be known by your users to combine the power of SAS® with a third-party (big) database.

In-Database Processing



From SAS® Data Management to Big Data Appliances: How SAS/ACCESS® Makes Life Easier

Magali Thésias
Senior Specialist - Deloitte

Context

90% of all the data
in the 
has been generated
over the last **2 years**

New technologies are developed to help companies handle this huge volume of data. Amongst them, the Massively Parallel Processing (MPP), enabling splitting data and queries across a large number of nodes in order to perform simultaneous computation.

What is the issue?

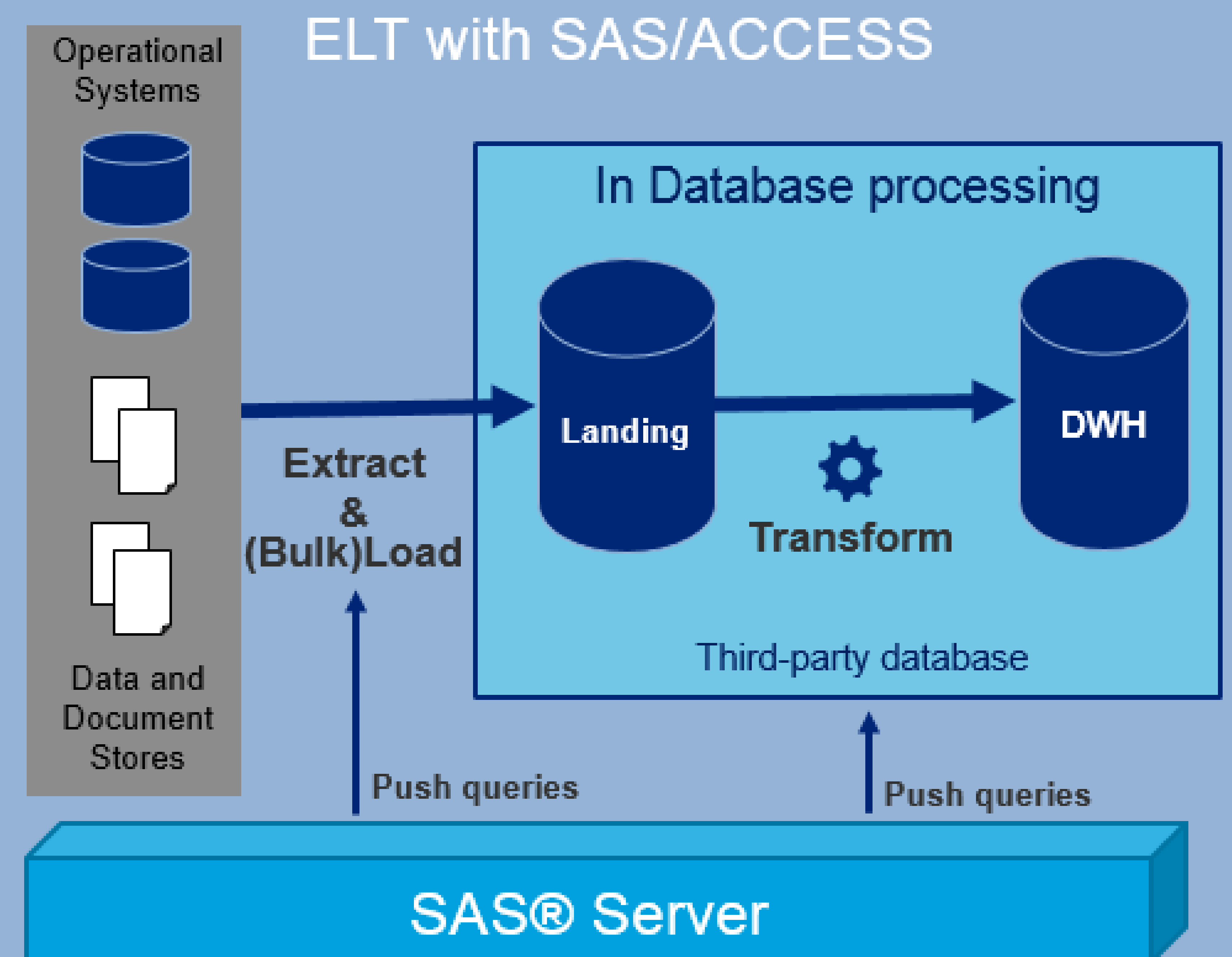
Many (large) companies are struggling in dealing with these technologies and more importantly on integrating them in their existing data management processes.

Moreover, they also want to rely on the knowledge built by their teams in existing products and implementing change to learn new technologies can therefore be a costly procedure.

How SAS is helping?

SAS® is perfectly fitting in this situation by offering a suite of software that can be set up to work with any third-party database through the usage of the corresponding SAS/ACCESS®. Indeed, for every new database technology SAS® is releasing a specific SAS/ACCESS® allowing users to develop and migrate SAS® solutions almost transparently. Only few techniques have to be known by your users to combine the power of SAS® with a third-party (big) database.

In-Database Processing



From SAS® Data Management to Big Data Appliances: How SAS/ACCESS® Makes Life Easier

Magali Thésias
Senior Specialist - Deloitte

How to push code for in-database processing?

Explicit pass-through

- Use the **PROC SQL CONNECT – EXECUTE** statements.
- The SQL query has to be able to work AS-IS in the database.
- Database will generate errors if a query contains anything SAS specific:
 - SAS formats.
 - SAS functions.
 - DATE or DATETIME actual numeric values.
 - INTO: macro variable.
 - SAS options.
 - Using multiple librefs.

```
PROC SQL;
```

```
CONNECT to NETEZZA (  
    DATABASE=SANDBOX  
    SERVER=&NZ_SERVER  
    USER=&NZ_USER  
    PASSWORD=&NZ_PSW);
```

```
EXECUTE (  
    CALL DROP_OBJECT_IF_EXISTS ('D_DATE', 'TABLE');  
) BY NETEZZA;
```

```
DISCONNECT FROM NETEZZA;
```

```
QUIT;
```


From SAS® Data Management to Big Data Appliances: How SAS/ACCESS® Makes Life Easier

Magali Thésias
Senior Specialist - Deloitte

How to push code for in-database processing?

Implicit pass-through

- Use the **LIBNAME** engine.
- Use the various DBMS data types and translate them into SAS formats.
- SAS attempts to generate a database specific SQL query that will be executed in-database.
- Not all SAS functions and procedures can be converted to DBMS-specific syntax. As example, Netezza engine supports:
 - 48 SAS functions
 - 7 SAS procedures

```
LIBNAME nz NETEZZA DATABASE='SANDBOX'  
                SERVER='XX.XX.XX.XXX'  
                SCHEMA='admin'  
                USER=admin  
                PASSWORD="XXXXXX";
```

From SAS® Data Management to Big Data Appliances: How SAS/ACCESS® Makes Life Easier

Magali Thésias
Senior Specialist - Deloitte

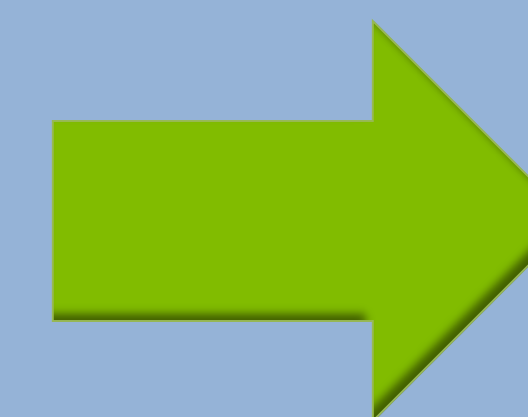
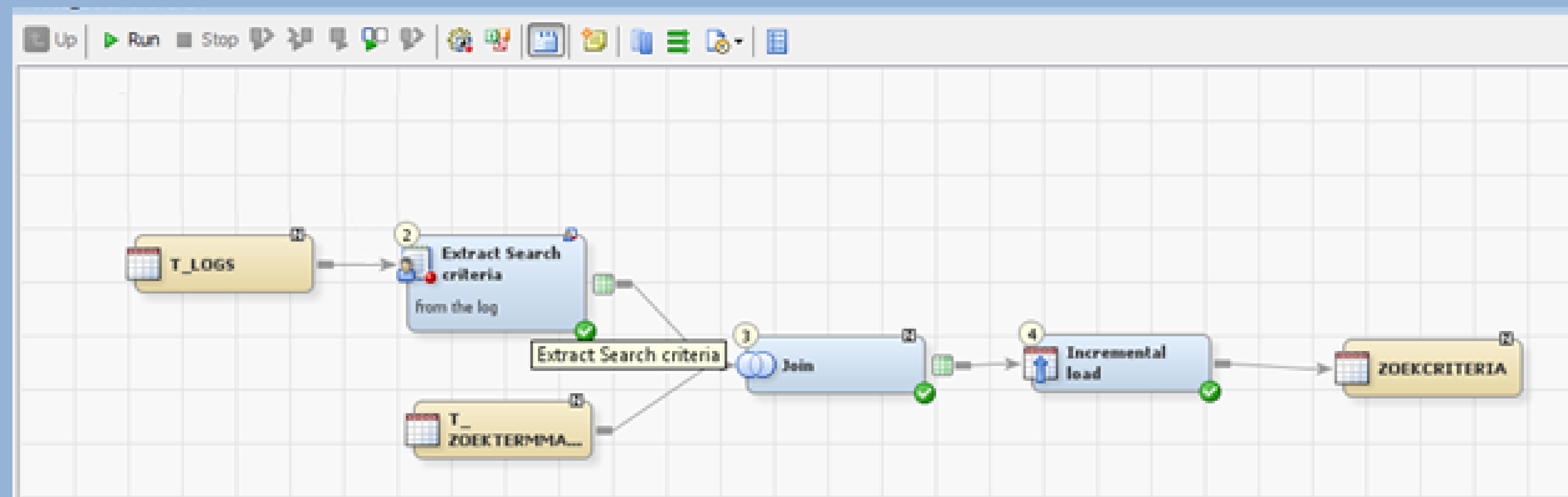
SAS® DATA INTEGRATION STUDIO

Data
Transforms

Apply
Business
Rules

Analytic
Transforms

Implicit pass-through code generation



Code generation mode: Automatic

```
91  * Target Table:   Extract - PDA.W10SB5C0           A5B1J9FM.BX000003 *
92  *-----*/
93
94  %let transformID = %quote(A5B1J9FM.BV000004);
95  %let trans_rc = 0;
96  %let etls_stepStartTime = %sysfunc(datetime(), datetime20.);
97
98  %let SYSLAST = %nrquote(PDA.Extract_Date);
99
100 %global etls_sql_pushDown;
101 %let etls_sql_pushDown = -1;
102 option DBIDIRECTEXEC;
103
104 /*---- Map the columns ----*/
105 %macro etls_deleteTable;
106   %local etls_tableExists;
107   %let etls_tableExists = %eval(%sysfunc(exist(PDA.W10SB5C0, DATA)) or
108     %sysfunc(exist(PDA.W10SB5C0, VIEW)));
109
110   %if (&etls_tableExists) %then
```

Diagram Code Log Output

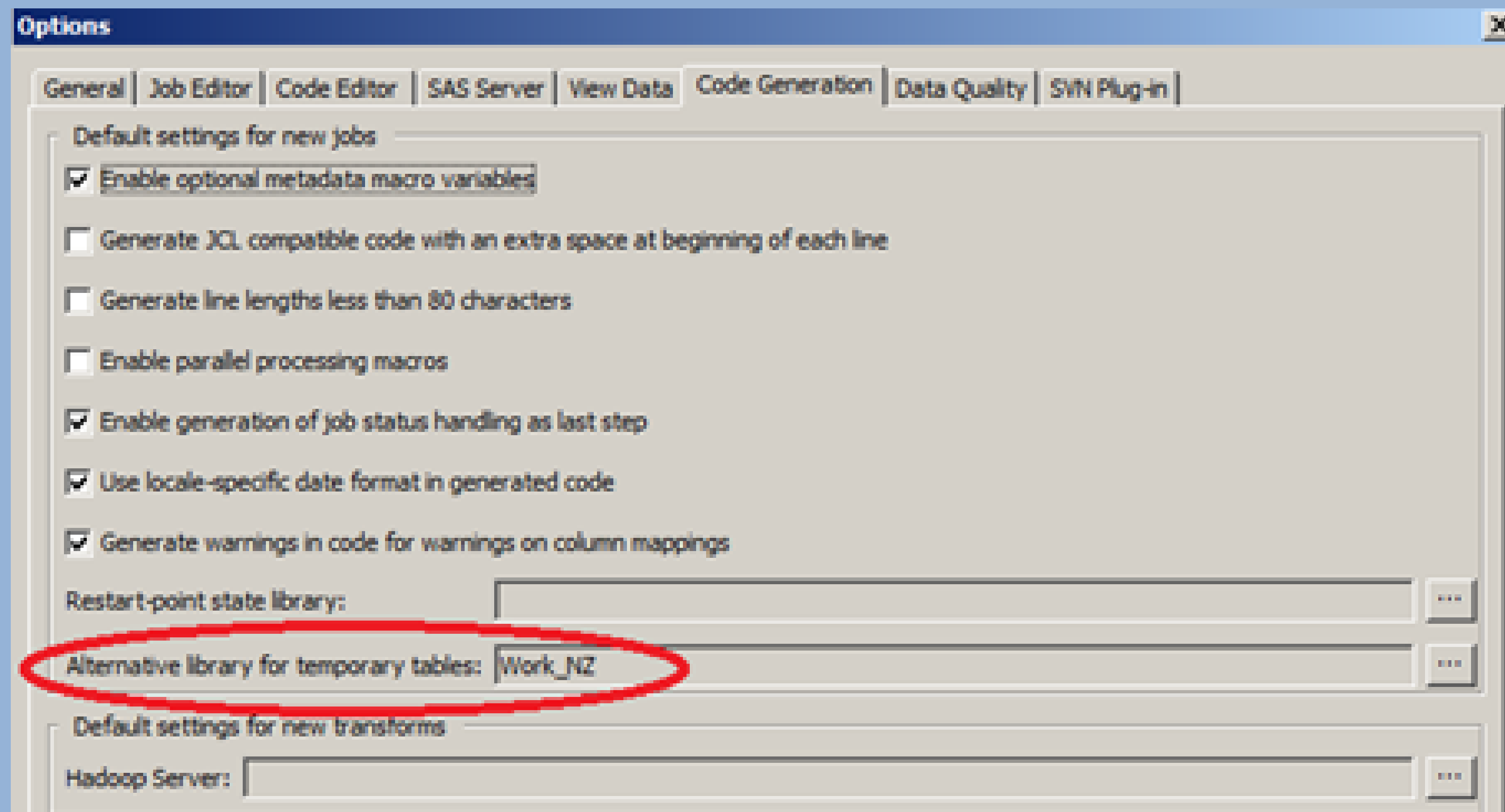
Details

From SAS® Data Management to Big Data Appliances: How SAS/ACCESS® Makes Life Easier

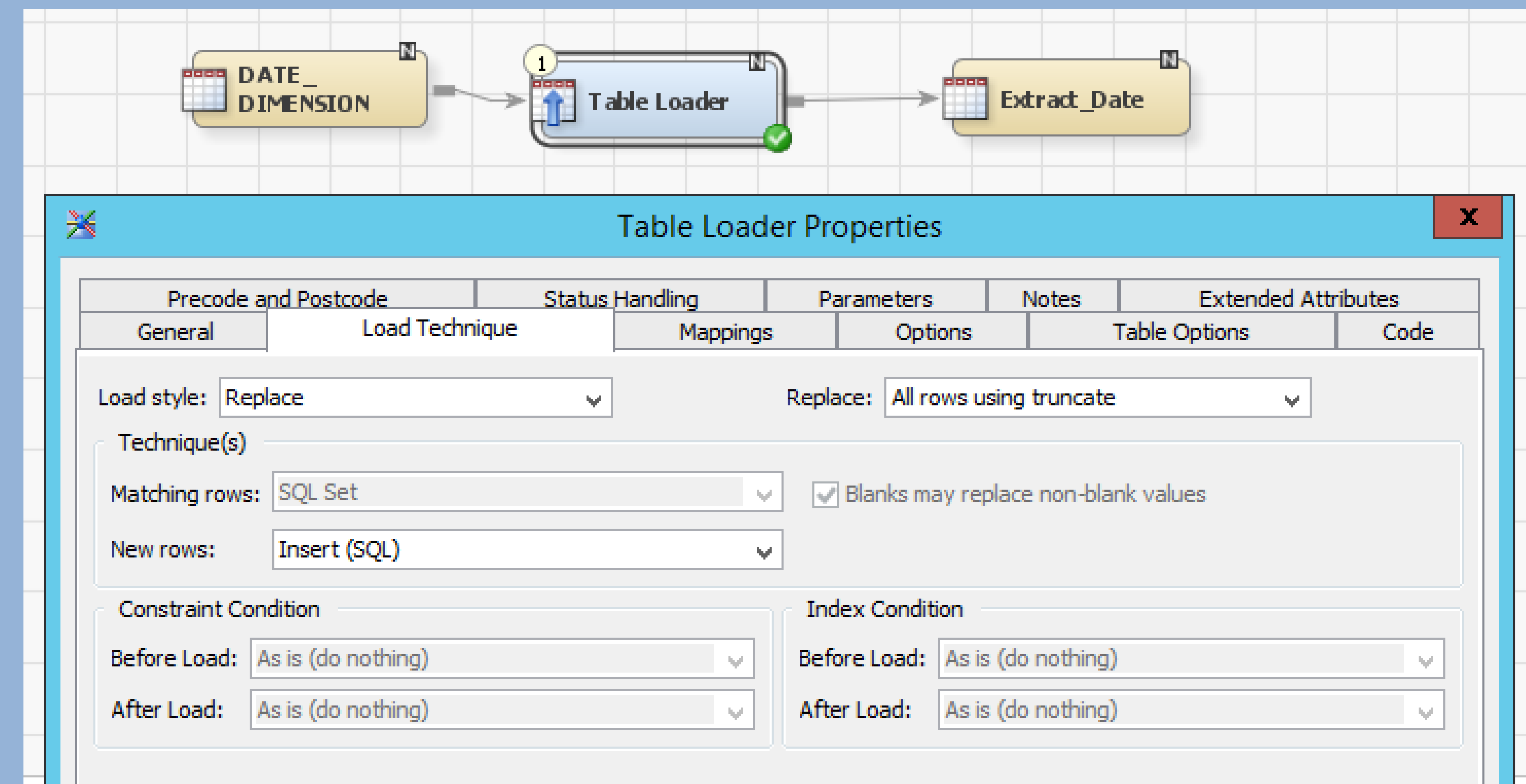
Magali Thésias
Senior Specialist - Deloitte

Best practices to create ELT data flows with SAS® DI Studio

Redirection of the default SAS work



Leverage bulk load capability

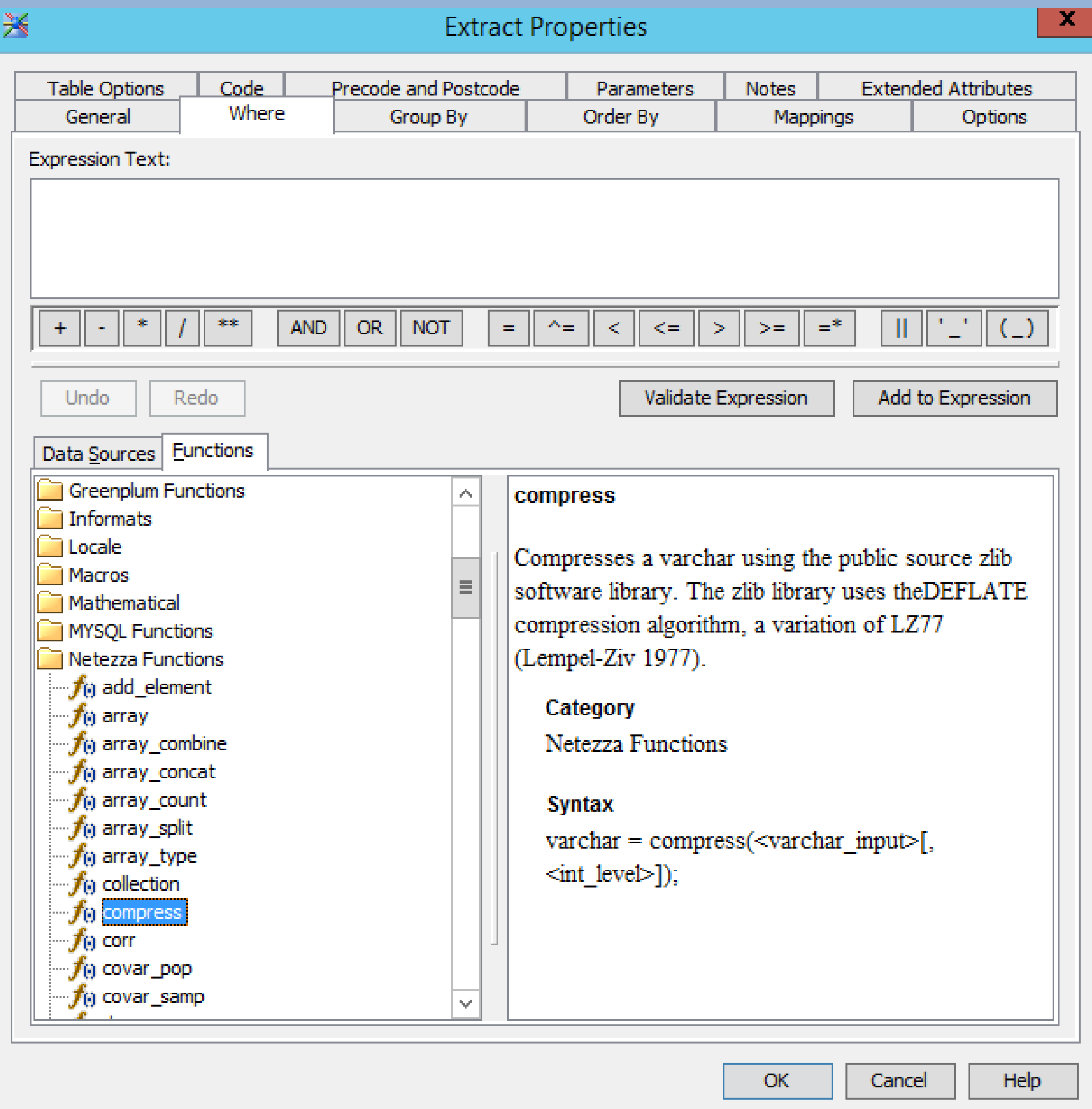


From SAS® Data Management to Big Data Appliances: How SAS/ACCESS® Makes Life Easier

Magali Thésias
Senior Specialist - Deloitte

Best practices to create ELT data flows with SAS® DI Studio

Use database specific SQL functions to generate
explicit SQL pass-through



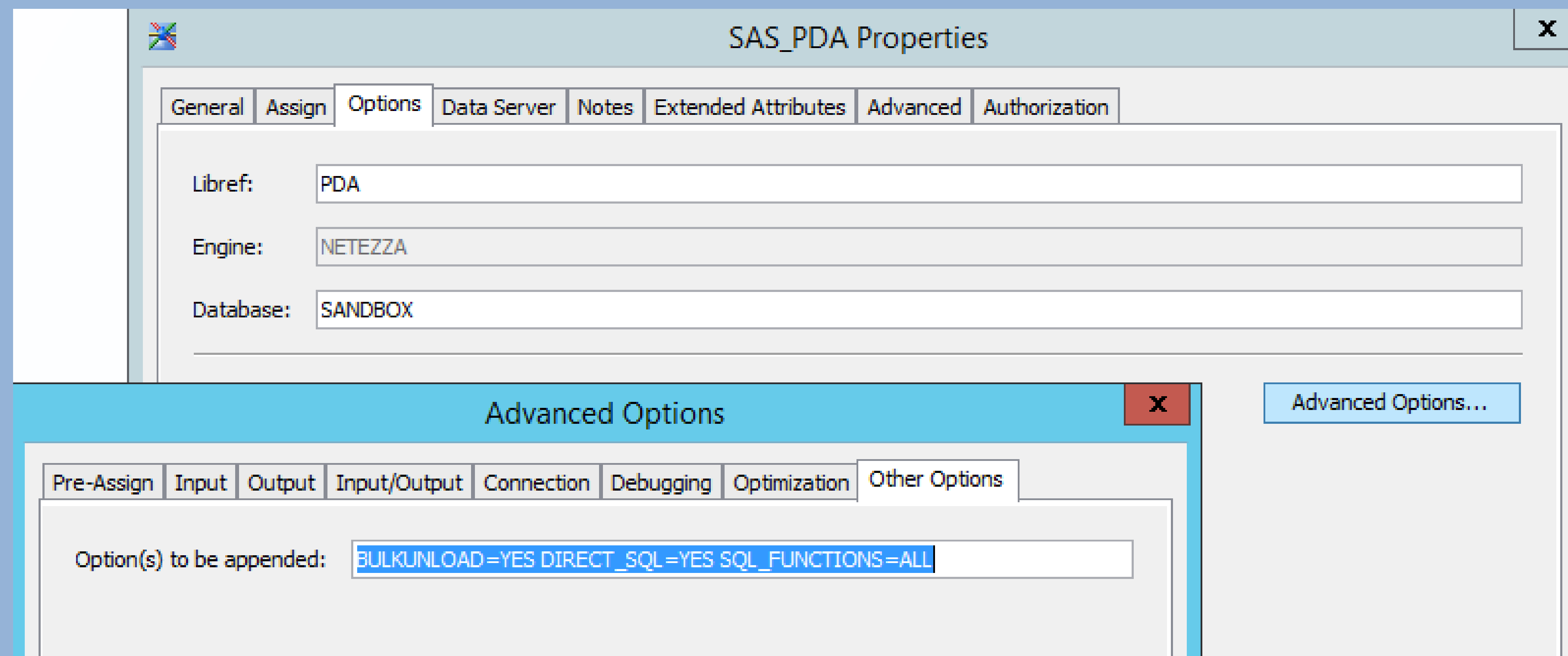
Create explicit pass-through SQL join

Join Properties	
Name	Value
Create SYSLAST Macro Variable	Yes
Automatically create join condition	No
System Options	
User Written	No
Pass Through	Yes
Target Table is Pass Through	Yes
Target Table Pass Through Action	Truncate
Debug	No
Suggest Sort Merge Join	No
Buffer Size	

From SAS® Data Management to Big Data Appliances: How SAS/ACCESS® Makes Life Easier

Magali Thésias
Senior Specialist - Deloitte

Libname options



BULKUNLOAD=YES

DIRECT_SQL=

SQL_FUNCTION=ALL

From SAS® Data Management to Big Data Appliances: How SAS/ACCESS® Makes Life Easier

Magali Thésias
Senior Specialist - Deloitte

Table Creation and Loading Options

BULKLOAD=YES

BL_OPTIONS=

DBCREATE_TABLE_OPTS=

BL_USE_PIPE=NO

BL_DELIMITER=

From SAS® Data Management to Big Data Appliances: How SAS/ACCESS® Makes Life Easier

Magali Thésias
Senior Specialist - Deloitte

See what is happening in database: System Option

In a transformation

The screenshot shows the 'Extract Properties' dialog box with the 'Options' tab selected. The left pane lists 'SQL Extract *', 'Additional Options *', and 'Checkpoint *'. The right pane contains the following settings:

- Use the optimized pass-through facility for SQL statements:** Lets the Pass-Through Facility optimize handling of SQL Statements by passing th...
☐ Yes
- * Use formats and informats in column definitions:** Generated code for this step includes column definitions with format and informat...
☐ Use job level setting
- System options:** Additional options to include on an OPTIONS statement, separated by spaces.
`OPTIONS SASTRACE=',,,d' SASTRACELOC=SASLOG NOSTSUFFIX;`

At job level

The screenshot shows the 'PDA Properties' dialog box with the 'Options' tab selected. The left pane lists 'General *', 'Checkpoint *', and 'Diagnostics/Statistics *'. The right pane contains the following settings:

- General**
- System Options:** Additional options to include on an OPTIONS statement, separated by spaces.



SASTRACE

From SAS® Data Management to Big Data Appliances: How SAS/ACCESS® Makes Life Easier

Magali Thésias
Senior Specialist - Deloitte

See what is happening in database: SASTRACE

The screenshot displays the SAS interface. The 'Create PDA Table Properties' dialog is open, showing the 'Code' tab. The code generation mode is set to 'Automatic'. The code being generated is as follows:

```
45  
46 proc sql  
47   _method  
48 ;  
49 create table PDA.NZ_TABLE  
50 (  
51   BULKLOAD=YES DBCREATE_TABLE_OPTS='DISTRIBUTE ON RANDOM' BL_USE_PIPE=NO BL_DELIMITER=',' BL_OPTIONS="maxerrors 100"  
52 ) as  
53 select *  
54 from  
55   SASDATA.WAC6IORA as WAC6IORA  
56 ;  
57 quit;
```

The 'PDA *' window is also open, showing the execution log. The log contains the following messages:

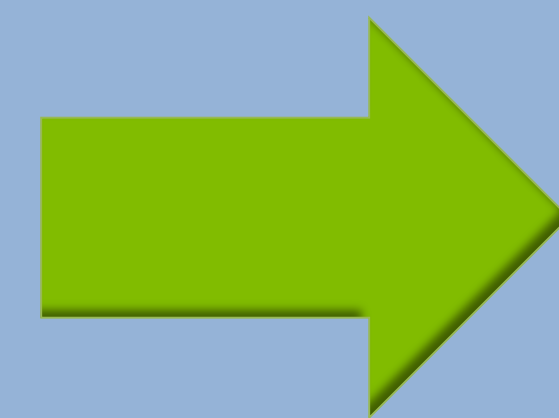
```
106 NETEZZA: AUTOCOMMIT turned OFF for connection id 2  
107 NOTE: SAS variable labels, formats, and lengths are not written to DBMS tables.  
108  
109 NETEZZA_5: Executed: on connection 2  
110 CREATE TABLE SANDBOX.ADMIN.NZ_TABLE (DDAT_SKEY BIGINT,DDAT_BKEY DATE,DDAT_DATE VARCHAR(10),DDAT_DATE_SHORT  
111  
112 ...  
130 VARCHAR(44),DDAT_SEMESTER_OFFSET BIGINT) DISTRIBUTE ON RANDOM |  
131  
132 NETEZZA: COMMIT performed on connection 2.  
133  
134 NETEZZA_6: Executed: on connection 2  
135 INSERT INTO SANDBOX.ADMIN.NZ_TABLE SELECT * FROM EXTERNAL 'BL_NZ_TABLE_5BEC915E-5E13-41BD-A1C5-29D30A264F38.dat' USING ( DELIMITER  
136 ', ' REMOTESOURCE 'ODBC' NULLVALUE '' maxerrors 100)  
137
```


From SAS® Data Management to Big Data Appliances: How SAS/ACCESS® Makes Life Easier

Magali Thésias
Senior Specialist - Deloitte

How to use these options? A concrete example

Log result thanks to SASTRACE system option



```
Netezza_20: Executed on connection 4
CREATE EXTERNAL TABLE '/tmp/tablename' using(DELIMITER '|' REMOTESOURCE
'ODBC' NULLVALUE '') AS SELECT
...
ERROR: CLI open cursor error: ERROR: found delim '|' in a data field,
specify escapeChar '\\' option in the external table definition.
```

- In-database processing cannot be executed, so an external table is created to download data on the SAS® server.
- The default delimiter (pipe) is used but this character is found in one of table columns.



Unload properly the data by using the bl_delimiter option.

From SAS® Data Management to Big Data Appliances: How SAS/ACCESS® Makes Life Easier

Magali Thésias
Senior Specialist - Deloitte

Conclusions

- SAS/ACCESS® is automatically generating SQL code compliant with your third-party database.
 - SAS® users can rely on their knowledge.
 - SAS® development is back-end independent.
- The only extra knowledge needed is a set of options to fine tune the in-database processing.
- Nevertheless, explicit pass-through SQL code can be sent to the third-party database.



SAS[®] GLOBAL FORUM 2016

IMAGINE. CREATE. INNOVATE.

LAS VEGAS | APRIL 18-21

#SASGF