ABSTRACT
If you haven’t heard about SAS® with Teradata Vantage, you will soon. Teradata’s offering has evolved beyond a leading enterprise data warehouse into Teradata Vantage, a modern architecture that integrates analytic engines, analytic tools & languages with seamless data integration. Vantage extends support for data types, formats and sources beyond traditional structured data. SAS® is an integral and essential component in a Vantage deployment, as SAS® delivers analytic tools and languages that extend the new Vantage platform to create a complete analytic ecosystem.

In this session, we will focus on answering these questions:

• How does SAS® fit into the Teradata Vantage analytics architecture?
• What are the benefits of using SAS® with Teradata Vantage?
• Can I put my analytic solution in the cloud?
• Do I need to change my SAS® code? Or my SQL?
• Can you show me some sample use cases?

INTRODUCTION
The Teradata Database has evolved into Teradata Vantage. Vantage is a modern analytics platform that provides access to data from across the enterprise for use with open source and commercial analytic technologies. Applications that leverage Vantage are solving complex business problems, operationalizing insights, and enabling analytics that lead to autonomous decision-making. Within Vantage, the legacy Teradata Database is now referred to as the “NewSQL Engine”. Both SAS® and the NewSQL Engine are essential components of Vantage. SAS® provides both analytics tools and an analytic language that integrates easily with Vantage. The NewSQL Engine provides persistent data storage, and a SQL interface. It enables users and applications to leverage powerful transformations, which prepare the data for further analytic processing.
Within Vantage, SAS® is both an analytic tool and language.

**Highlights of the Vantage Architecture:**

- The NewSQL Engine is both the SQL Engine and the persistent data storage
- Integration from SAS® to the NewSQL Engine is provided by SAS/ACCESS to Teradata
- Analytic Engines in Vantage Release 1.0 include:
  - Teradata Machine Learning Engine
  - Teradata Graph Engine
- Additional Analytic Engines are planned for future releases of Vantage
- SAS® provides analytic tools and a primary analytic language
- The data storage layer, NewSQL Engine, and Teradata Analytic Engines are interconnected using a high-speed fabric, which accelerates performance
- SAS® servers and SAS® In-Memory technologies can also be added to the same high-speed fabric
SAS Viya in Teradata Vantage

SAS Viya can be co-located with the Analytic Engines and connected via a high-speed fabric to the NewSQL Engine & Data Store.

**BENEFITS FOR JOINT SAS® AND TERADATA CUSTOMERS**

Vantage extends the capabilities available in SAS® by leveraging In-Database processing, as well as providing access to additional analytic functions, data types & data sources.

The key benefits of In-Database processing:

- Pushes SAS® processing down to the NewSQL Engine nodes using the SAS® Embedded Process for Teradata, and feature-specific user defined functions from SAS®

- Reduces or eliminates the lag time of transferring large amounts of data
  - Pushes the work where the data resides, instead of moving the data to the processing
  - SAS® server nodes can be deployed close to the Teradata nodes on the high-speed fabric, to enable high speed data movement when needed.

- Leverages the scalability and efficiency of the NewSQL Engine’s optimizer for processing large volumes of data in parallel

- Accesses all the functions in the NewSQL Engine, including 100’s of existing functions and many new analytic functions

Further benefits of In-Database processing are improved data governance and security, because there are fewer copies of sensitive data.
**SAS® IN-DATABASE PRODUCTS AVAILABLE WITH SAS® & VANTAGE**

A wide range of In-Database functions developed jointly by SAS® and Teradata are available, including:

- SAS® Scoring Accelerator
- SAS® Code Accelerator
- SAS® Analytics Accelerator
- SAS® In-Database Formats
- SAS® Data Quality Accelerator
- SAS® Base Procedures
- SAS/ACCESS Interface to Teradata
- SAS/ACCESS Interface to Teradata (on SAS® Viya)
- SAS® In-Database technologies for Teradata (on SAS® Viya)

**Figure 3 — SAS® In-Database Functions:**

<table>
<thead>
<tr>
<th>SAS/ACCESS to Teradata and SAS® Base Procedures:</th>
<th>SAS® Code Accelerator for Teradata</th>
<th>SAS® Scoring Accelerator for Teradata</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROC APPEND</td>
<td>PROC D62 with Thread Programs</td>
<td>SAS/EM and SAS/STAT Models</td>
</tr>
<tr>
<td>PROC CONTENTS</td>
<td>PROC Transpose</td>
<td></td>
</tr>
<tr>
<td>PROC COPY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROC DATASETS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROC DELETE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROC FORMAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROC FREQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROC MEANS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROC PRINT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROC RANK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROC REPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROC SORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROC SQL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROC SUMMARY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROC TABULATE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SAS Analytics Accelerator for Teradata**

**Statistical Analysis Procedures:**

- PROC CANCORR
- PROC CORR
- PROC FACTOR
- PROC PRINCOMP
- PROC REG
- PROC SCORE
- PROC TIMESERIES
- PROC VARCLUS

- PROC SCORE works with coefficients from:
  - PROC ACECLUS
  - PROC CALIS
  - PROC CANDISC
  - PROC DISCRIM
  - PROC FACTOR
  - PROC PRINCOMP
  - PROC TCALIS
  - PROC YARCLUS
  - PROC ORTHOREG
  - PROC QUANTREG
  - PROC REG
  - PROC ROBUSTREG

**SAS® Enterprise Miner**

- PROC DMDB
- PROC DMINE
- PROC DMREG (Logistic Regression)
- Nodes for:
  - Input, Sample, Partition, Filter, Merge, Expand

**EXTENDED IN-DATABASE FUNCTIONS IN VANTAGE**

Vantage adds new analytic functions within the NewSQL Engine and optional Machine Learning and Graph Engines. Function aliasing technology hides the details from the end user about where the functions run. All functions are accessed using SQL, whether they run within the NewSQL Engine or another analytic engine.
The NewSQL Engine is the component of Vantage formally called Teradata Database. The NewSQL Engine has all the same capabilities and compatibility guarantees as all prior versions of the Database. In addition, your existing SAS® In-Database products will continue to function with the NewSQL Engine.

As mentioned, the Machine Learning and Graph Engines are Optional Analytic Engines within Vantage. Vantage also supports the more robust SAS Graph and Machine Learning capabilities.
**VANTAGE DEPLOYMENT OPTIONS**

For seamless application portability between environments, the same software is used across all deployment options. When SAS® supports a specific Teradata version or capability, you can be assured that it will work the same regardless of where Vantage is installed.

**Figure 5 – Locations Vantage can be deployed**

Vantage Deployment options include:

- **Teradata Cloud** – Managed by Teradata in a Teradata center (with multiple locations worldwide)
- **Public Cloud** – Vantage with the NewSQL Engine is available on AWS & Azure. Full deployments of Vantage with ML/Graph Engines on AWS & Azure are planned in a future release.
- **Teradata Hardware** – On-Premise at customer site using Teradata IntelliFlex 2.1 hardware, with plans to release ML/Graph Engine add-ons for existing Teradata hardware deployments.
- **Commodity Hardware** – Uses VMware on customer commodity hardware. Vantage with the NewSQL Engine is available today. Full deployments of Vantage with ML/Graph Engines are also planned.

When Teradata is deployed on-premise or in the Teradata Managed Cloud, everything can be installed on the same high-speed fabric for efficient data management and low latency.

**DO I NEED TO CHANGE MY SAS® CODE OR MY SQL FOR VANTAGE?**

NO. There is no need to change any of your existing SAS® code or SQL with Vantage. All the existing SAS® code and Teradata SQL will function as before & produce the same results.

For Example:
LIBNAME statements do not change.

```
libname td teradata server=<server name> user=<user name>
   password=<password> database=<database name>;
```

Implicit Pass-Through SQL does not change. PROC SQL Example:

```
proc sql;
create table work.some_table1 as
   select a,b,c, count(*) as cnt
```
from td.some_table group by a,b,c;
quit;

Explicit Pass-Through SQL statement does not change. PROC SQL Example:

proc sql;
  connect using td as tdx;
  create table work.some_table2 as select * from connection to tdx
  (select a,b,c from sasdemo.some_table where a < 100);
  disconnect from tdx;
quit;

PROC MEANS does not change. This and other Base SAS Procedures continue to run In-Database:

proc means
data=td.some_table; /* columns a,b,c */
quit;

VANTAGE SAMPLE USE CASES
For SAS® to utilize the extended In-Database Functions in Vantage, you must use the Explicit Pass-Through SQL functionality available with PROC SQL and/or PROC FEDSQL. The syntax for the SAS® Code and SQL statements are defined in the Teradata documentation (See References for a link to the documentation). The following three sample use cases demonstrate using SAS® with Vantage.

USE CASE #1 – SESSIONIZE Function in Vantage
Background: The customer, a well-known band, hosts a website where fans can download their music (for a fee, of course!). Keeping track of the band’s most popular songs and driving more fans to the website are always important. The band also wants to use the website data to plan their next concert tour.

Data: The web server gathers each user’s id, their activity while on the site and the referral site/source (i.e. how the customer was driven to the website: Direct Access, Search Engine or Social Media).

Question #1: What songs should be band include in their upcoming tour? The band wants to determine what songs were downloaded around the times they played their previous concert in each city. They will use this information to build a tailored playlist for each location.

SAS® Code to SESSIONIZE the data: Uses PROC SQL Explicit Pass-Through with the SESSIONIZE function in Vantage:

```sql
proc sql;
  connect to teradata(server=TDProd username=sas password=sas database=sas);

  /* Sessionize the data & store in a new table */
  execute (CREATE TABLE Sessionized AS (SELECT *
      FROM SESSIONIZE (ON SessionizeTest
      PARTITION BY UUID
      ORDER BY DownloadTimestamp
      USING TimeColumn ('DownloadTimestamp')TimeOut(60) ClickLag (0.2))
    ) WITH DATA;
  ) by teradata;
  execute (COMMIT) by teradata;
  disconnect from teradata;
quit;
```
**SAS® Code to graph downloads:** Uses PROC G3D to create a 3-D graph to show many times each music file was downloaded during the last year. The concert dates are associated with the data:

```sas
/* Load the data into SAS datasets from TD & Graph it */
libname td teradata server=TDProd user=sas password=sas database=sas;
proc sort
  data=td.MonthDownloads
  out=downloadgraph;
by MonthID DocID ;
run;
/* Generate a format to be used for the City Locations in the 3D Graph */
PROC FORMAT;
VALUE XX
  1 - 2 = 'Boston'
  /*more formats here for 3 through 7 */
  7 - 8 = 'Chicago'
  8 - 9 = 'Los Angeles'
  9 - 10 = 'Denver';
run;
/* Generate the 3D Graph
The 3D Graph shows the Number of times each Song was downloaded.
The dates of the concerts from the previous tour were added in */
goptions reset=all reset=global;
title1 'Song Download History';
title2 'With Associated Tour Dates';
proc g3d
  data=downloadgraph;
  label MonthID='Location' DocID='Song ID' DownloadCnt='Number of Downloads';
  plot MonthID*DocID=DownloadCnt /
    rotate=45
    tilt=45
    xticknum=6
    yticknum=10
    zticknum=5;
  format MonthID XX.
;run;
```
Graph 1 - The number of downloads of each song, correlated with their Concert Tour Dates

Observations: This shows that the download activity on the website has increased since the beginning of the year. Also, looking at the peaks around concert times, you can pick out popular songs. To improve popularity of some of their music, the band could play the song at the concert and/or offer a free download of the song to fans that attend.

Question #2: How can the band increase traffic to their website?

SAS® Code to graph referrals: Uses PROC GCHART to create a Pie Chart showing what percentage of sessions were driven to the site by referral type, for the last year:

```sas
/* Load the data into SAS datasets from TD & Graph it */
libname td teradata server="TDProd" user=sas password=sas database=sas;

proc sort
    data= td.References
    out=Sort2DGraph;
    by ReferralNum ReferralYear;
run;
```
/* Generate a Pie Chart
   The Chart shows % of Sessions by Referral Type for the last year;
*/
goptions reset=all reset=global;
title 'Percent of Sessions by Referral Type';
pattern1 color=purple;
pattern2 color=blue;
pattern3 color=yellow;
proc gchart data=Sort2DGraph;
  pie ReferralDesc / sumvar=AverageSessionCnt
    type=mean
    value=none
    noheading
    percent=arrow
    slice=arrow
    plabel=(font='Albany AMT/bold' h=1.5 color=black);
run;

Graph 2 – Shows the Percentage of sessions by each referral source/type for the last year
Observations: Very little traffic is coming from Social Media sites. This is an area that can be capitalized on to drive more traffic to the band’s website. The band can offer a free download if their fan shares their website on Facebook or Tweets about them.

USE CASE #2 - NPATH
Background: A retailer has a small online business which sells popular cell phones and accessories.

Data: The retailer tracks the traffic on their website; capturing all visitor activity from viewing the home page through to check out or drop off.

Question: Which promotion from last month drove the most site visitors to purchase a product? The 3 promotions were: A Free Cell Phone Case with Purchase, 10% Off your Purchase, and 3 Months No Interest on a Purchase.

SAS® code to use the NPATH function in Vantage: Uses SAS Studio in Viya, to run the NPATH function in Vantage using PROC FEDSQL. The result is returned to the CAS Server. Then you can open Visual Analytics in Viya and graph the results:

```sas
/* run the highlighted query in TD using PROC FedSQL explicit pass-through, results stored in CAS */
proc fedsql sessref=expt _method ;
create table tdlib.SankeyTest{options replace=true} as
  select sessionid, clickcount, purcahsetotals, firstclick, lastcheckout, path
  from connection to tdlib(
    SELECT * FROM NPATH(
      ON clickstream PARTITION BY SessionNumber ORDER BY clicktime
      USING
        SYMBOLS(Pagetype='Home' AS Home,
        Pagetype <> 'Home' AND Pagetype <> 'Checkout' AS ClickView,
        Pagetype <> 'Home' AS Checkout)
      PATTERN ('Home.ClickView*.Checkout')
      RESULT (  
        FIRST (SessionNumber of ANY(Home, Checkout, ClickView)) AS SessionId,  
        COUNT (* of any(Home, Checkout, ClickView)) AS ClickCount,  
        LAST(PurchaseTotal of ANY(Checkout)) AS Purcahsetotals,  
        FIRST (clicktime of ANY(Home,ClickView)) AS FirstClick,  
        LAST (clicktime of ANY(Checkout)) AS LastCheckout,  
        ACCUMULATE (itempage of ANY(Home,ClickView,Checkout)) AS PATH
      )
    ) MODE (OVERLAPPING)
  )
quit;
/* save the results of the query from CAS lib back into TD as a perm table */
proc casutil;
  save incaslib="tdlib" casdata="SankeyTest"
  outcaslib="tdlib" casout="SankeyTest"
run;
/* print sample rows */
```
Next, you can use SAS® Visual Analytics in Viya to map Sankey graph of data.

**Graph 3 – Sankey Diagram showing path of user sessions**

**Observations:** The 2 most successful offers were 10% Off & 3 Months No Interest. Only 1 customer made a purchase after viewing the Free Phone Case offer. The retailer may not want to run the Free Phone Case offer again in the future.

**USE CASE #3 – ARIMA**

**Background:** A manufacturing company produces bottled water.

**Data:** The company has all their historical inventory data.

**Question:** How much inventory does the company need to keep on-hand to fulfill orders in the next 2 years (what quantities are needed by month)?

**SAS® code to use ARIMA functions in Vantage:** Using Explicit Pass-Through with the new ARIMA function which is part of the Machine Learning Engine in Vantage:

```sas
proc sql;
  connect to teradata(server="TDProd" username=sas password=sas database=sas);
  /* Run the ARIMA function to create the ARIMA Model & Residual tables
     These will be used by the ARIMAPredict function to predict future data.
  */
  execute (CREATE TABLE arimatest AS (SELECT * FROM ARIMA (ON inventory AS InputTable
```
OUT TABLE ModelTable (arimamodel)
OUT TABLE ResidualTable (arimaresidual)
USING
  TimeColumns ('d')
  TargetColumn ('data')
  Orders ('1,1,1')
  SeasonalOrders('1,1,1')
  Period(12)
  AS dt
) AS DATA
  PRIMARY INDEX (td_arima_partition_id);
) by teradata;
execute (COMMIT) by teradata;

/* Use the ARIMA model & residual tables in the ARIMAPredict function to forecast/predict future data */
Execute (CREATE TABLE arimamodelpredict AS (
  SELECT * FROM ArimaPredict (ON arimamodel AS ModelTable PARTITION BY td_arima_partition_id
  ON arimaresidual AS ResidualTable PARTITION BY td_arima_partition_id ORDER BY d
  USING
    PartitionColumns ('td_arima_partition_id')
    StepAhead ('48')
  ) dt
) WITH DATA
  UNIQUE PRIMARY INDEX (stepahead);
) by teradata;
execute (COMMIT) by teradata;
disconnect from teradata;
quit;

...<Additional Data Transformation occurs here, to get in the proper format for graphing>...

**SAS® Code to graph forecast:** Using PROC SGPLOT to graph historical & predicted inventory.

/* Copy table into SAS dataset & define Historical/Forecasted data */
libname td teradata
  server=TDProd
  username=sas
  password=sas
  database=sas;

data arimatransform;
  set td.arimatransform;
  if DateData <= '01DEC2018'd then BlockID = "Historical";
  else BlockID = "Forecast";
run;

/* Sort SAS Dataset by Date */
proc sort
  data=arimatransform
  out=forecast;
  by DateData;
run;

ods graphics on / attrpriority=none;
titel1 COLOR=bib "Predict Water Bottle Inventory" ;
titel2 "ARIMA Model and Forecast";
proc sgplot data=forecast noautolegend;
  styleattrs DATACOLORS=(verylightgrey verylightred) /* region */
    DATALINEPATTERNS=(solid dot) /* line patterns */;
  block x=DateData block=BlockID / transparency=0.75;
  scatter x=DateData y=Co2Level;
**Observations:** There is more demand for bottled water in the summer months. The peak occurs in August. Also, there is an overall upward trend in the demand for bottled water. The company will need to increase/decrease their inventory taking both trends into consideration.

**CONCLUSION**

SAS® and Teradata® have been partners for over a decade. They continually capitalize on the strengths of both technologies to provide cutting-edge analytic solutions. Teradata Vantage is another example of advances in In-Database analytics that are available to SAS® and Teradata® customers.

**LINKS & REFERENCES**

- Teradata Vantage 1.0 User Guide
- Teradata Database Analytic Functions
- Teradata Vantage Machine Learning Engine Analytic Function Reference
- SAS® In-Database Product Documentation
RECOMMENDED READING
Explicit SQL Pass-Through

ACKNOWLEDGEMENTS
We would like to thank the SAS® Global Alliances, R&D and Project Management groups for their support. This paper would not have been possible without their assistance.

CONTACT INFORMATION
Heather Burnette
Heather.Burnette@Teradata.com
(919) 531-2059
https://www.teradata.com

Greg Otto
Greg.Otto@Teradata.com
(919) 531-2661
https://www.teradata.com

Salman Maher
Salman.Maher@sas.com
(919) 531-5847
https://www.sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.
Other brand and product names are trademarks of their respective companies.