MWSUG 2018 – Paper HW-098

The Joinless Join ~ The Impossible Dream Come True
Using SAS® Enterprise Guide® and Base SAS® PROC SQL and DATA Step;

Expand the Power of SAS® Enterprise Guide® and Base SAS® in New Ways

Kent ♥ Ronda Team Phelps ~ The SASketeers ~ *All for SAS & SAS for All!*
Illuminator Coaching, Inc. ~ Des Moines, Iowa

ABSTRACT

SAS Enterprise Guide and Base SAS can easily combine data from tables or data sets by using a PROC SQL Join to match on like columns or by using a DATA Step Merge to match on the same variable name. However, what do you do when tables or data sets do not contain like columns or the same variable name and a Join or Merge cannot be used?

We invite you to attend our exciting Joinless Join Hands-On Workshop where we will empower you to expand the power of SAS Enterprise Guide and Base SAS in new ways by creatively overcoming the limits of a standard Join or Merge.

You will learn how to design a Joinless Join based upon dependencies, indirect relationships, or no relationships at all between the tables or data sets using SAS Enterprise Guide and Base SAS PROC SQL and DATA Step. In addition, we will highlight how to use a Joinless Join to prepare unrelated joinless data to be utilized by ODS and PROC REPORT in creating a PDF. Come experience the power and versatility of the Joinless Join to greatly expand your data transformation and analysis toolkit.

We look forward to introducing you
to the surprising paradox of the
Joinless Join.

INTRODUCTION

The tagline for SAS is *The Power To Know*® and your 'power to know' greatly expands with your ability to access, combine, and analyze important data from tables or data sets (referred to as tables going forward). **The Power To Know** sets off **The Power To Create** which leads to **The Power To Automate ~** much like an intricate and fluid domino design. However, this power will quickly become disjointed if you do not know how to effectively Join or Merge tables of data **~ even when the tables do not have a relationship**.

Here are 2 questions to ask yourself when analyzing 2 or more tables:

- Do the tables contain like columns or the same variable name which can be utilized in a Join or Merge?
- If the tables do not contain like columns or the same variable name and a standard Join or Merge cannot be used, have I reached a *cavernous and insurmountable 'woe is me' research impasse* in my data analysis?

☺ **There is no need to fear, the Joinless Join is here!** ☺

The Joinless Join will bridge your research impasse and empower you to:

- Creatively overcome the limits of a standard Join or Merge using SAS Enterprise Guide and Base SAS
- Access, combine, and analyze tables for the first time based upon dependencies, indirect relationships, or no relationships at all
- Open up new worlds of table creations, calculations, validations, and filtrations
- Prepare unrelated joinless data to be utilized by ODS and PROC REPORT
- Increase your ability to detect and resolve errors including hidden errors
- Prevent validation process failure ~ yea! ~ and completely... yes, completely automate your projects

The SAS project in this presentation demonstrates:

- **The Power To Know** how to design a Joinless Join
- **The Power To Create** tables based upon dependencies, indirect relationships, or no relationships at all
- **The Power To Automate** projects even when tables cannot be directly joined or merged

We invite you to journey with us as we empower you to
E X P A N D
the power of Base SAS and Enterprise Guide in new ways.

Brief Overview of Standard PROC SQL Joins and DATA Step Merges

Just traveling along...
side-by-side.
Harry Macgregor Woods

A standard Join or Merge enables you to combine tables side-by-side horizontally by matching related rows. A like column or the same variable name, with the same attributes and like values, is used to connect the tables and bring together some or all of each table's contents.

An **Inner Join or Merge** is a symmetrical process of matching related rows in tables ~ an Inner Join can match related rows in **2 to 256** tables, and a Merge can match related rows in **2** tables.

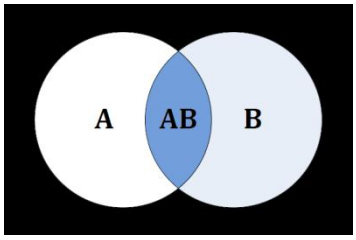
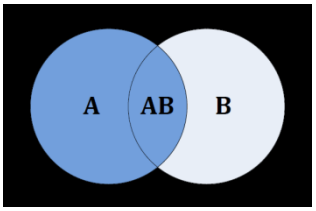


Figure 1. Venn Diagram – Inner Join or Merge

The result of an **Inner Join or Merge** produces only matched rows from the tables. The result is illustrated by the shaded area AB in **Figure 1**.

An **Outer Join or Merge** is an asymmetrical process of matching related rows in **2** tables. The resulting set of data also contains **unmatched** rows from the left, right, or both tables.



The result of a **Left Outer Join or Merge** produces matched rows from both tables while preserving all unmatched rows from the left table. The result is illustrated by the shaded areas A and AB in **Figure 2**.

Figure 2. Venn Diagram – Left Outer Join or Merge

The result of a **Right Outer Join or Merge** produces matched rows from both tables while preserving all unmatched rows from the right table. The result is illustrated by the shaded areas B and AB in **Figure 3**.

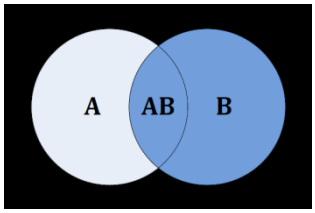
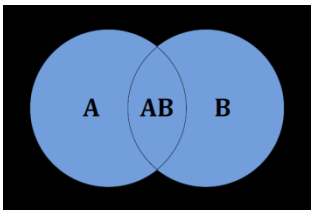


Figure 3. Venn Diagram – Right Outer Join or Merge



The result of a **Full Outer Join or Merge** produces matched rows while preserving all unmatched rows from both tables. The result is illustrated by the shaded areas A, AB, and B in **Figure 4**.

Figure 4. Venn Diagram – Full Outer Join or Merge

All of these Joins and Merges have an important common denominator ~ each of them requires a like column or the same variable name for matching. Thus, we now return to the core focus of this presentation…

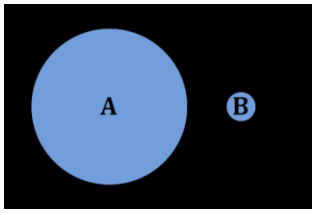


Figure 5. Venn Diagram – Tables Without Like Columns or the Same Variable Name

What do you do when the tables you want to analyze do not contain like columns or the same variable name (**Figure 5**) and a standard Join or Merge cannot be used?

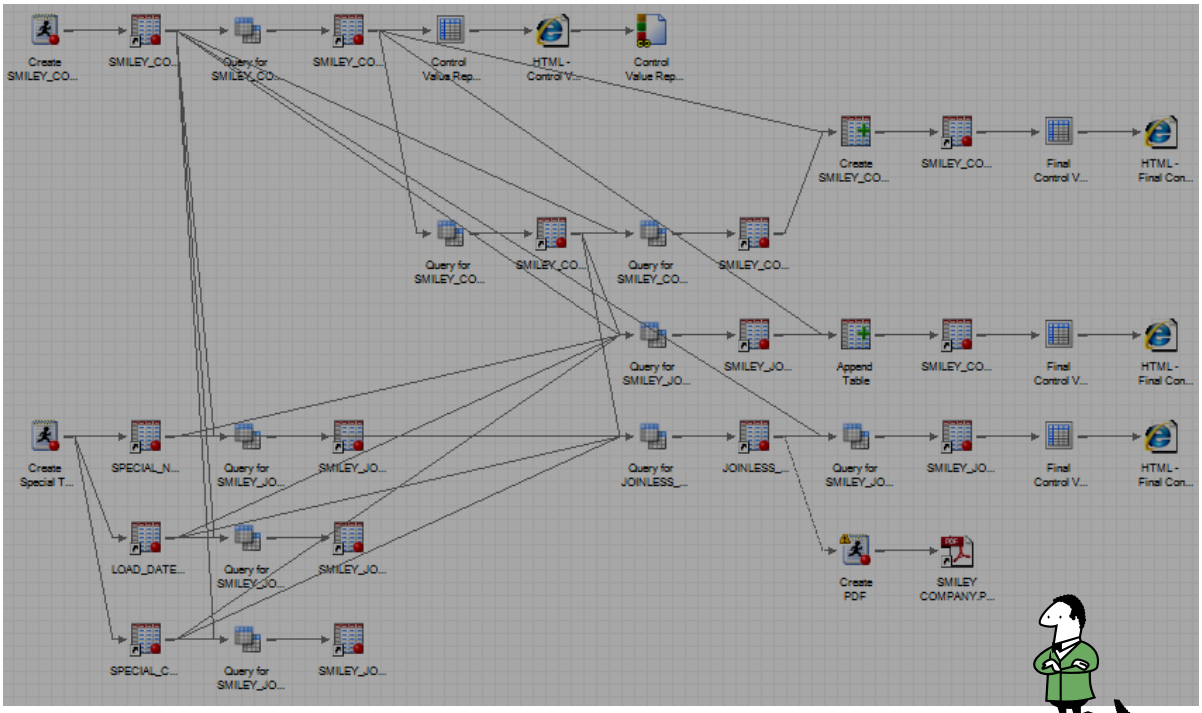
In the next section we will continue to follow The Power To Know dominoes to find the answer.

Professor Domino will be our guide ☺

Illuminating the Paradox of the Joinless Join

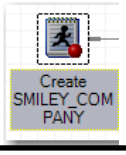
Sometimes success is seeing
what we already have
in a new light.
Dan Miller

Here is our SAS Enterprise Guide project example:



- **Our project example demonstrates some of the ways a Joinless Join can be used.**

First we design a Program Node to create a source table:




```
DATA SMILEY_COMPANY;
    LENGTH Special_Person $20 Special_Number 8 Special_Car $1 Load_Date 8;
    FORMAT Load_Date date9.;
    INFILE DATALINES DELIMITER=',';
    INPUT Special_Person $ Special_Number Special_Car $ Load_Date;
DATALINES;
Smiley,10127911,Honda,'01JAN2018'd
Smiley's Son,10173341,Chevy,'01JAN2018'd
Smiley's Twin,10376606,Buick,'01JAN2018'd
Smiley's Wife,10927911,Saturn,'01JAN2018'd
Smiley's Son,11471884,Dodge,'01JAN2018'd
Smiley's Twin,11573691,GMC,'01JAN2018'd
Smiley's Daughter,11975386,Chevy,'01JAN2018'd
Smiley's Son,12071884,Jeep,'01JAN2018'd
Smiley's Son,12871884,Dodge,'01JAN2018'd
Smiley's Twin,13173691,Jeep,'01JAN2018'd
Smiley's Wife,13771202,Dodge,'01JAN2018'd
Smiley's Daughter,13775498,Honda,'01JAN2018'd
Smiley's Son,14171884,Infinity,'01JAN2018'd
Smiley's Twin,15373691,Ford,'01JAN2018'd
Smiley's Son,15471884,Chevy,'01JAN2018'd
Smiley's Son,16074330,Honda,'01JAN2018'd
Smiley's Daughter,16175498,Buick,'01JAN2018'd
Smiley's Wife,16176964,Honda,'31DEC2017'd
Smiley,16279111,Saturn,'01JAN2018'd
Smiley's Twin,16573691,Chevy,'01JAN2018'd
RUN;
```

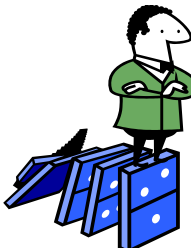
- **This is the code you will need to recreate this table.**

The Program Node creates the SMILEY_COMPANY source table:

	Special_Person	Special_Number	Special_Car	Load_Date
1	Smiley	10127911	Honda	01JAN2018
2	Smiley's Son	10173341	Chevy	01JAN2018
3	Smiley's Twin	10376606	Buick	01JAN2018
4	Smiley's Wife	10927911	Saturn	01JAN2018
5	Smiley's Son	11471884	Dodge	01JAN2018
6	Smiley's Twin	11573691	GMC	01JAN2018
7	Smiley's Daughter	11975386	Chevy	01JAN2018
8	Smiley's Son	12071884	Jeep	01JAN2018
9	Smiley's Son	12871884	Dodge	01JAN2018
10	Smiley's Twin	13173691	Jeep	01JAN2018
11	Smiley's Wife	13771202	Dodge	01JAN2018
12	Smiley's Daughter	13775498	Honda	01JAN2018
13	Smiley's Son	14171884	Infinity	01JAN2018
14	Smiley's Twin	15373691	Ford	01JAN2018
15	Smiley's Son	15471884	Chevy	01JAN2018
16	Smiley's Son	16074330	Honda	01JAN2018
17	Smiley's Daughter	16175498	Buick	01JAN2018
18	Smiley's Wife	16176964	Honda	31DEC2017
19	Smiley	16279111	Saturn	01JAN2018
20	Smiley's Twin	16573691	Chevy	01JAN2018

- **The SMILEY_COMPANY table is used throughout this presentation.**
- **This table contains each Special Person, Special Number, and Special Car of the ☺ Smiley Company ☺ employees.**
- **Load_Date is the date when each row was created.**

Next we design another Program Node to create 3 additional tables:



```
DATA National_Average (KEEP=National_Average)

     Load_Date_Check (KEEP=Load_Date_Check)

     Pace_Car (KEEP=Pace_Car);

    LENGTH Load_Date_Check 8;
    FORMAT Load_Date_Check date9.;

    National_Average = 12000000;
    OUTPUT National_Average;

    Load_Date_Check = '01JAN2018'd;
    OUTPUT Load_Date_Check;

    Pace_Car = 'Saturn';
    OUTPUT Pace_Car;
RUN;
```

- **This is the code you will need to recreate these tables.**

Here are the 3 additional tables the Program Node creates:

National_Average table:

	National_Average
1	12000000

Load_Date_Check table:

	Load_Date_Check
1	01JAN2018

Pace_Car table:

	Pace_Car
1	Saturn

- **The National_Average table contains the average of all the Special_Number columns from each Smiley Company nationwide which we will use in a Joinless Join to calculate a percentage of the Special_Number column in our SMILEY_COMPANY table.**
- **The Load_Date_Check table contains a Load Date which we will use in a Joinless Join to validate that all of our SMILEY_COMPANY table rows were created in 2018.**
- **The Pace_Car table contains a Special Car from the Smiley Company National Headquarters which we will use in a Joinless Join to filter our SMILEY_COMPANY table output.**

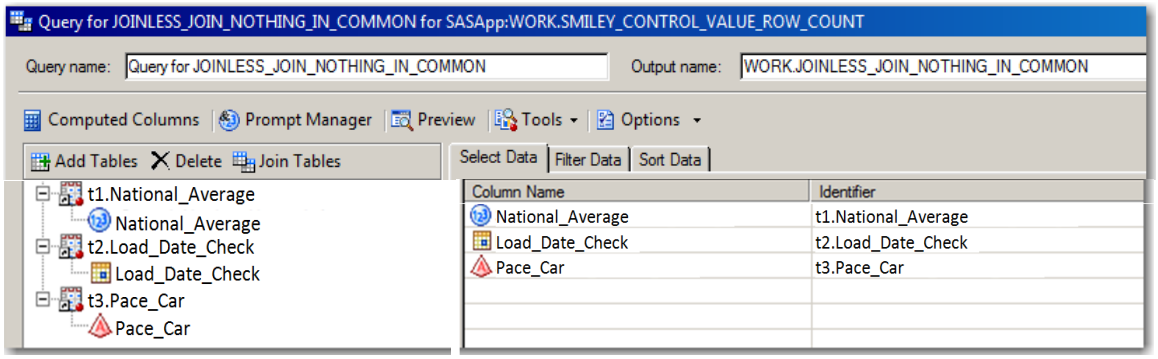
Designing a Joinless Join to combine 3 tables with No Relationships At All using the 3 additional tables that the 2nd Program Node created:

	National_Average
1	12000000

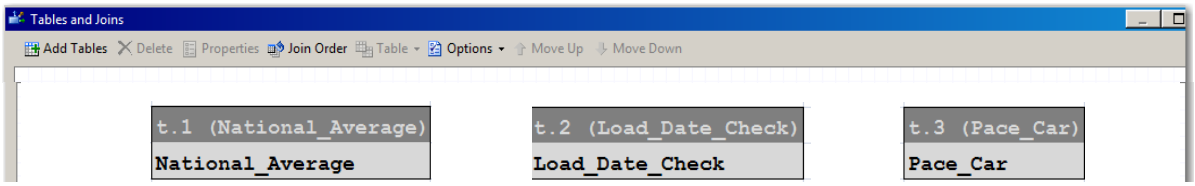
	Load_Date_Check
1	01JAN2018

	Pace_Car
1	Saturn

- **Notice how the 3 columns in the 3 tables have No Relationships At All.**



- **Build a Query with the NATIONAL_AVERAGE, LOAD_DATE_CHECK, and PACE_CAR tables.**



- **This time the Joinless Join is based upon the NATIONAL_AVERAGE, LOAD_DATE_CHECK, and PACE_CAR tables having No Relationships At All.**

No Column to match on? No Problem ~

We will use a Joinless Join based upon No Relationship At All between the tables.

What to do, what to do…

Necessity is the mother of all inventions.

Plato / Einstein

How the Joinless Join works:

The Cartesian product of 3 Tables with 1 Row and 1 Column in each Table

Input Data(3) | Code | Log | Output Data

Modify Task | Filter and Sort | Query Builder | Data | Describe | Graph | Analyze

	National_Average	Load_Date_Check	Pace_Car
1	12000000	01JAN2018	Saturn

- The Joinless Join automatically creates a Cartesian Product which places the 1 row and 1 column of the NATIONAL_AVERAGE, LOAD_DATE_CHECK, and PACE_CAR tables to the right of each other.

JOINLESS_JOIN_NOTHING_IN_COMMON

Filter and Sort | Query Builder | Data | Describe | Graph | Analyze | Export | Send To

	National_Average	Load_Date_Check	Pace_Car
1	12000000	01JAN2018	Saturn

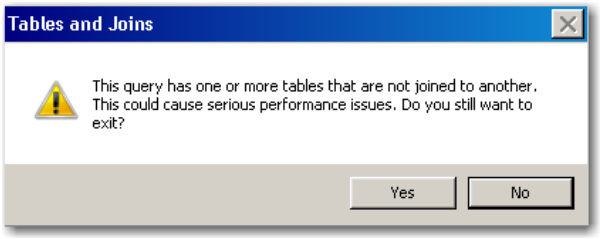
- Here is the final result from selecting all 3 columns which is equal to the Cartesian Product.

SAS Highlight

A Cartesian Product is a result set of all the possible rows and columns contained in 2 or more tables. The resulting set of data can potentially become extremely large and unmanageable. The most noticeable coding characteristic of a PROC SQL join which produces a Cartesian Product is the absence of a WHERE-clause. The most noticeable coding characteristic of a DATA step which produces a Cartesian Product is the careful and strategic use of multiple SET statements.

Always Remember, It's Too Soon To Quit!

Bob Wieland (Mr. Inspiration)



- This **Warning Message** always appears whenever tables are joined with a Joinless Join because SAS knows it will create a **Cartesian Product** which can take a lot of extra resources.

Caution:

When you design your Joinless Join make sure that one of the tables has only ONE row!

This PROC SQL also creates the JOINLESS_JOIN_NOTHING_IN_COMMON table:

```
PROC SQL;
    CREATE TABLE JOINLESS_JOIN_NOTHING_IN_COMMON AS
    SELECT National_Average,
           Load_Date_Check,
           Pace_Car
    FROM NATIONAL_AVERAGE, LOAD_DATE_CHECK, PACE_CAR;
QUIT;
```

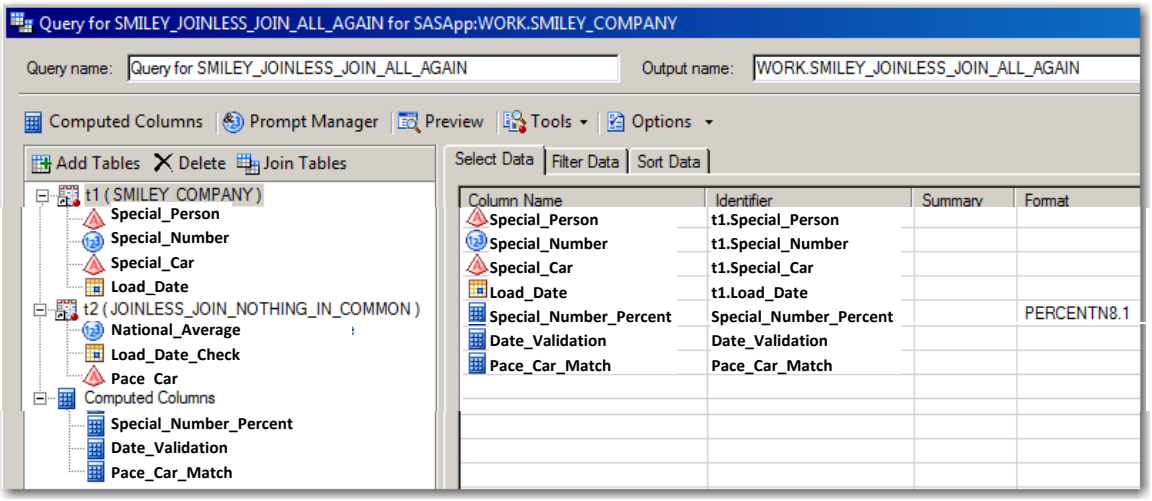
- The `PROC SQL` creates a `TABLE` called `JOINLESS_JOIN_NOTHING_IN_COMMON` by selecting `National_Average` from the `NATIONAL_AVERAGE` table, `Load_Date_Check` from the `LOAD_DATE_CHECK` table, and `Pace_Car` from the `PACE_CAR` table.
- Notice the `FROM` does not contain any type of join between the 3 tables and thus a Joinless Join is once again achieved.
- The result from this `PROC SQL` will be identical to what was achieved by building the GUI Joinless Join above.

This DATA step also creates the JOINLESS_JOIN_NOTHING_IN_COMMON table:

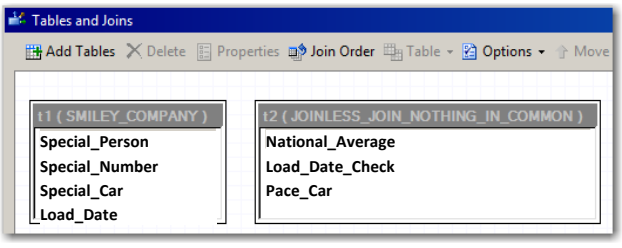
```
DATA JOINLESS_JOIN_NOTHING_IN_COMMON;
     SET National_Average;
     SET Load_Date_Check;
     SET Pace_Car;
RUN;
```

- The **DATA** step creates a data set called **JOINLESS_JOIN_NOTHING_IN_COMMON** by **SET**ing the **NATIONAL_AVERAGE**, the **LOAD_DATE_CHECK** , and the **PACE_CAR** tables using individual **SET** statements.
- Since each of the 3 data sets only contain 1 observation, individually **SET**ing each data set horizontally places each column in the 3 data sets side-by-side and thus is a Joinless Join.
- **Please be aware that this Joinless Join using the DATA step will only work if each of the data sets only contain 1 observation ~ Our next example will demonstrate how to perform a Joinless Join with a data set which contains more than 1 observation.**

Designing a Joinless Join to perform a Calculation, Validation, and Filtration:



- **Build a Query with the SMILEY_COMPANY and the JOINLESS_JOIN_NOTHING_IN_COMMON table.**



- **The Joinless Join is based upon all 3 columns in the JOINLESS_JOIN_NOTHING_IN_COMMON table which indirectly relate to specific columns in the SMILEY_COMPANY table.**

The Cartesian product of SMILEY_COMPANY and 1 Table with 1 Row and 3 Columns

Input Data (2) | Code | Log | Output Data

Modify Task | Filter and Sort | Query Builder | Data | Describe | Graph | Analyze | Export | Send To

	Special_Person	Special_Number	Special_Car	Load_Date	National_Average	Load_Date_Check	Pace_Car
1	Smiley	10127911	Honda	01JAN2018	12000000	01JAN2018	Saturn
2	Smiley's Son	10173341	Chevy	01JAN2018	12000000	01JAN2018	Saturn
3	Smiley's Twin	10376606	Buick	01JAN2018	12000000	01JAN2018	Saturn
4	Smiley's Wife	10927911	Saturn	01JAN2018	12000000	01JAN2018	Saturn
5	Smiley's Son	11471884	Dodge	01JAN2018	12000000	01JAN2018	Saturn
6	Smiley's Twin	11573691	GMC	01JAN2018	12000000	01JAN2018	Saturn
7	Smiley's Daughter	11975386	Chevy	01JAN2018	12000000	01JAN2018	Saturn
8	Smiley's Son	12071884	Jeep	01JAN2018	12000000	01JAN2018	Saturn
9	Smiley's Son	12871884	Dodge	01JAN2018	12000000	01JAN2018	Saturn
10	Smiley's Twin	13173691	Jeep	01JAN2018	12000000	01JAN2018	Saturn
11	Smiley's Wife	13771202	Dodge	01JAN2018	12000000	01JAN2018	Saturn
12	Smiley's Daughter	13775498	Honda	01JAN2018	12000000	01JAN2018	Saturn
13	Smiley's Son	14171884	Infinity	01JAN2018	12000000	01JAN2018	Saturn
14	Smiley's Twin	15373691	Ford	01JAN2018	12000000	01JAN2018	Saturn
15	Smiley's Son	15471884	Chevy	01JAN2018	12000000	01JAN2018	Saturn
16	Smiley's Son	16074330	Honda	01JAN2018	12000000	01JAN2018	Saturn
17	Smiley's Daughter	16175498	Buick	01JAN2018	12000000	01JAN2018	Saturn
18	Smiley's Wife	16176964	Honda	31DEC2017	12000000	01JAN2018	Saturn
19	Smiley	16279111	Saturn	01JAN2018	12000000	01JAN2018	Saturn
20	Smiley's Twin	16573691	Chevy	01JAN2018	12000000	01JAN2018	Saturn

- **The Joinless Join automatically creates a Cartesian Product which places the 1 row and 3 columns of the NATIONAL_AVERAGE, LOAD_DATE_CHECK, and PACE_CAR tables to the right of each of the 20 rows and 4 columns in the SMILEY_COMPANY table.**

Computed Columns

Column	Details
Date_Validation	CASE WHEN t1.Load_Date GE t2.Load_Date_Check THEN 'AOK' ELSE 'NOT AOK' END
Pace_Car_Winner	CASE WHEN t1.Special_Car EQ t2.Pace_Car THEN 'Winner' ELSE 'Try Again' END
Special_Number_Percent	t1.Special_Number/t2.National_Average

- **The Calculation, Validation, and Filtration are represented by Computed Columns that are derived as shown above using the 3 columns in the JOINLESS_JOIN_NOTHING_IN_COMMON table.**

	Special_Person	Special_Number	Special_Car	Load_Date	Special_Number_Percent	Date_Validation	Pace_Car_Winner
1	Smiley	10127911	Honda	01JAN2018	84.4%	AOK	Try Again
2	Smiley's Son	10173341	Chevy	01JAN2018	84.8%	AOK	Try Again
3	Smiley's Twin	10376606	Buick	01JAN2018	86.5%	AOK	Try Again
4	Smiley's Wife	10927911	Saturn	01JAN2018	91.1%	AOK	Winner
5	Smiley's Son	11471884	Dodge	01JAN2018	95.6%	AOK	Try Again
6	Smiley's Twin	11573691	GMC	01JAN2018	96.4%	AOK	Try Again
7	Smiley's Daughter	11975386	Chevy	01JAN2018	99.8%	AOK	Try Again
8	Smiley's Son	12071884	Jeep	01JAN2018	100.6%	AOK	Try Again
9	Smiley's Son	12871884	Dodge	01JAN2018	107.3%	AOK	Try Again
10	Smiley's Twin	13173691	Jeep	01JAN2018	109.8%	AOK	Try Again
11	Smiley's Wife	13771202	Dodge	01JAN2018	114.8%	AOK	Try Again
12	Smiley's Daughter	13775498	Honda	01JAN2018	114.8%	AOK	Try Again
13	Smiley's Son	14171884	Infinity	01JAN2018	118.1%	AOK	Try Again
14	Smiley's Twin	15373691	Ford	01JAN2018	128.1%	AOK	Try Again
15	Smiley's Son	15471884	Chevy	01JAN2018	128.9%	AOK	Try Again
16	Smiley's Son	16074330	Honda	01JAN2018	134.0%	AOK	Try Again
17	Smiley's Daughter	16175498	Buick	01JAN2018	134.8%	AOK	Try Again
18	Smiley's Wife	16176964	Honda	31DEC2017	134.8%	NOT AOK	Try Again
19	Smiley	16279111	Saturn	01JAN2018	135.7%	AOK	Winner
20	Smiley's Twin	16573691	Chevy	01JAN2018	138.1%	AOK	Try Again

- **Here is the final result with the Calculation, Validation, and Filtration Computed Columns to the right of each of the 20 rows and 4 columns of the SMILEY_COMPANY table.**

This PROC SQL also creates the SMILEY_JOINLESS_JOIN_ALL_AGAIN table:

```
PROC SQL;
    CREATE TABLE SMILEY_JOINLESS_JOIN_ALL_AGAIN AS
    SELECT t1.Special_Person,
           t1.Special_Number,
           t1.Special_Car,
           t1.Load_Date,
          /* Special_Number_Percent */
             (t1.Special_Number/t2.National_Average)
              FORMAT=PERCENT8.1 AS Special_Number_Percent,
          /* Date_Validation */
             (CASE
                 WHEN t1.Load_Date GE t2.Load_Date_Check
                     THEN 'AOK'
                 ELSE 'NOT AOK'
              END) AS Date_Validation,
          /* Pace_Car_Winner */
             (CASE
                 WHEN t1.Special_Car = t2.Pace_Car
                     THEN 'Winner'
                 ELSE 'Try Again'
              END) AS Pace_Car_Winner
    FROM SMILEY_COMPANY t1, JOINLESS_JOIN_NOTHING_IN_COMMON t2;
QUIT;
```

- The `PROC SQL` creates a `TABLE` called `SMILEY_JOINLESS_JOIN_ALL_AGAIN` by selecting `Special_Person`, `Special_Number`, `Special_Car`, and `Load_Date` from the `SMILEY_COMPANY` table.
- The `Special_Number_Percent`, `Date_Validation`, and `Pace_Car_Winner` columns are calculated or derived as shown in the GUI example using the 3 columns in the `JOINLESS_JOIN_NOTHING_IN_COMMON` table.
- Notice the `FROM` does not contain any type of join between the 2 tables and thus is a Joinless Join.

This DATA Step also creates the SMILEY_JOINLESS_JOIN_ALL_AGAIN table:

```
DATA SMILEY_JOINLESS_JOIN_ALL_AGAIN;
    IF _N_ = 1 THEN SET JOINLESS_JOIN_NOTHING_IN_COMMON;
    SET SMILEY_COMPANY;
    FORMAT Special_Number_Percent PERCENT8.1;
    /* Special_Number_Percent */
    Special_Number_Percent = Special_Number/National_Average;
    /* Date_Validation */
    IF Load_Date GE Load_Date_Check
        THEN Date_Validation = 'AOK'
    ELSE Date_Validation = 'NOT AOK'
    END-IF;
    /* Pace_Car_Winner */
    IF Special_Car = Pace_Car
        THEN Pace_Car_Winner = 'Winner'
    ELSE Pace_Car_Winner = 'Try Again'
    END-IF;
RUN;
```

- The DATA step creates a data set called SMILEY_JOINLESS_JOIN_ALL_AGAIN by SETing the JOINLESS_JOIN_NOTHING_IN_COMMON data set ~ before the first observation of the SMILEY_COMPANY data set is SET ~ thereby causing the single observation to be merged with all of the observations in the SMILEY_COMPANY data set.

- Please be aware that this Joinless Join using the DATA step will only work if the data set that is SET ~ before the first observation of the second data set is SET ~ only contains 1 observation.

- The Special_Number_Percent, Date_Validation, and Pace_Car_Winner columns are calculated or derived similar to the GUI and PROC SQL examples with the exception of using DATA step syntax when necessary.

Design a Quarterly Report PDF utilizing the results of the Joinless Join of the 3 tables with No Relationships At All:

JOINLESS_JOIN_NOTHING_IN_COMMON

Filter and Sort | Query Builder | Data | Describe | Graph | Analyze | Export | Send To

	National_Average	Load_Date_Check	Pace_Car
1	12000000	01JAN2018	Saturn

- **The Smiley Company has requested a quarterly report of the results of the Joinless Join of the 3 tables with No Relationships At All that we designed earlier in the paper.**

```
ODS PDF FILE='d:\HOW\Phelps\SMILEY COMPANY.PDF' NOTOC;
TITLE 'SMILEY COMPANY - QUARTERLY VALIDATION PARAMETERS';
PROC REPORT DATA=JOINLESS_JOIN_NOTHING_IN_COMMON NOWD;
    COLUMNS National_Average Load_Date_Check Pace_Car;
    DEFINE National_Average / STYLE={WIDTH=25mm JUST=CENTER} "National Average";
    DEFINE Load_Date_Check  / STYLE={WIDTH=25mm JUST=CENTER} "Load Date Check";
    DEFINE Pace_Car         / STYLE={WIDTH=25mm JUST=CENTER} "Pace Car";
RUN;
ODS PDF CLOSE;
```

- **The `ODS PDF FILE` statement opens the `SMILEY COMPANY.PDF` with no table of contents – `NOTOC`.**
- **The `TITLE` statement includes the title shown at the top of the PDF.**
- **`PROC REPORT` is used to report the contents of the `JOINLESS_JOIN_NOTHING_IN_COMMON` table in the PDF with no default report window – `NOWD`.**
- **The `COLUMNS` statement tells `PROC REPORT` which columns to include in the report.**
- **The `DEFINE` statements provide a `WIDTH` and justification along with renaming each column.**
- **The `ODS PDF CLOSE` statement closes the PDF.**

13:30 Tuesday, October 2, 2018 1

SMILEY COMPANY - QUARTERLY VALIDATION PARAMETERS

National Average	Load Date Check	Pace Car
12000000	01JAN2018	Saturn

Yea!!!

Strike up the band,

Toss the confetti,

Release the balloons!

Applause... Applause... Applause...

Bring out the treats for everyone!

☺☺

CONCLUSION

The **Joinless Join** empowers you to creatively overcome the limits of a standard Join or Merge and enables you to expand the power of Base SAS and SAS Enterprise Guide in new ways. **The Power To Know** how to design a Joinless Join sets off **The Power To Create** tables based upon, dependencies, indirect relationships, or no relationships at all which leads to **The Power To Automate** projects even when tables cannot be directly joined or merged **~** ☺ try saying that statement really fast for fun ☺!

The Joinless Join bridges the research impasse you experience when needing to combine data from tables which do not contain like columns or the same variable name. New worlds of table creations, calculations, validations, filtrations, and PROC REPORTing have opened up to greatly expand your data transformation and analysis toolkit. Begin thinking about how you can benefit from the power and versatility of the Joinless Join.

How wonderful it is that we need not wait a single minute before starting to improve ourselves and our world!

Anne Frank

SAS Programming is like a series of intricate and fluid domino designs and you are the **Designer**. Your desire to design a quality program fuels your thoroughness and attention to detail. As a SAS Professional, your inquisitive nature, research oriented mindset, and solution driven focus are among your greatest assets.

Your life is like a campfire at night - You never know how many people will see it and be comforted and guided by your light.

Claire Draper

Rule #6: Study hard and learn all you can.

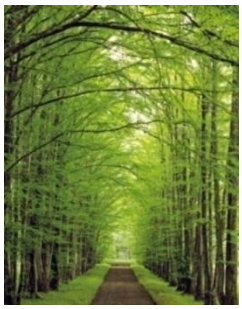
☺ **Roy Rogers Riders Club Rules** ☺

Always remember – *It's not what the SAS World holds for you, it's what **YOU** bring to it!* Continue to develop and build on your many skills and talents. Keep looking for different ways to share your God-given abilities and ideas. You will soon discover new and creative ways to design your SAS programs. Plan on coming back to the MWSUG Regional Conference next year to shed some light on the exciting things you are learning. All of us are on the SAS journey with you and we look forward to your teaching sessions in the future.

As we conclude, we want to introduce you to our **SAS Mascot, Smiley**. Smiley represents the **SAS Joy** which each of us experience as we find better ways to accomplish mighty and worthy deeds using SAS. The three of us, along with Professor Domino, hope we have expanded and enriched your SAS knowledge.

Thank You for sharing part of your SAS journey with us ~

☺ **Happy SAS Trails to you… until we meet again** ☺



ACKNOWLEDGMENTS

We want to thank the 29th Annual MWSUG 2018 ***Hands-On-Workshop*** *Chair,* ***Misty Johnson,*** *for accepting our abstract and paper, and we want to express our appreciation to the Conference Co-Chairs,* ***Dave Foster*** *(Academic Chair) and* ***Cindy (Lee) Wilson*** *(Operations Chair), the Executive Committee and Conference Leaders, and SAS Institute for their diligent efforts in organizing this illuminating conference.*

You inspire us to share what we are learning and we hope to be a light of encouragement to you as well ~ Your friends, Kent ♥ Ronda ***Team*** *Phelps ~ The* ***SAS****keteers ~* ***All for SAS & SAS for All!*** *~ Illuminator Coaching, Inc.*

MEET THE AUTHORS

Writing is a permanent legacy.

John C. Maxwell

Kent ♥ Ronda Team Phelps are the co-founders of Illuminator Coaching, Inc., & The **SAS**keteers: *All for* ***SAS*** *and* ***SAS*** *for All!* **~** they recently co-presented at the SAS Global Forum (SGF) 2018 International Conference in Denver, CO and have co-authored 13 SAS White Papers.

Kent wants to encourage and equip you to fulfill your life, career, and leadership potential as you build an enduring legacy of inspiration, excellence, and honor **~** SAS® Certified Professional Programmer Analyst Consultant **~** B.S. Electrical Engineering **~** serving for over 20 years as an essential bridge builder of consensus and quality programming to connect the needs of Business and IT **~** happily programmed in Base SAS® and SAS® Enterprise Guide® since 2007 with a strong focus on engineering innovative, efficient, and automated solutions **~** and presented and co-presented at the MidWest SAS® Users Group (MWSUG) Regional Conference for the last 6 years including 3 Hands-On Workshops.

Ronda believes that YOU are a gift the world is waiting to receive, and she wants to encourage and equip you to pursue your unique destiny as you navigate your life journey with intentionality, fulfilling purpose, and enduring hope **~** Writer & Coach **~** gifted in helping others to explore and express their hearts and minds through writing **~** served in the Banking and Insurance industries for 19 years **~** and co-presented at 3 MidWest SAS® Users Group (MWSUG) Conferences including a Hands-On Workshop.

We invite you to share your valued comments with us:

Kent ♥ Ronda Team Phelps
The SAS keteers ~ *All for SAS & SAS for All!*
E-mail: SASketeers@IlluminatorCoaching.com

☺ ***We look forward to connecting with you in the future!*** ☺

REFERENCES

Carpenter, Art (2012), *PROC REPORT Basics: Getting Started with the Primary Statements*, Proceedings of the 6th Annual SAS Global Forum (SGF) 2012 International Conference, California Occidental Consultants, Anchorage, AK, USA.
http://support.sas.com/resources/papers/proceedings12/242-2012.pdf

Celko, Joe (2010), *Joe Celko's SQL for Smarties, Fourth Edition: Advanced SQL Programming (The Morgan Kaufmann Series in Data Management Systems)*; November 10, 2010; ISBN-10: 0123820227; ISBN-13: 978-0123820228.
http://www.accuteach.com/book/joe-celkos-sql-for-smarties-fourth-edition-advanced-sql-programming-the-morgan-kaufmann-series-in-data-management-systems-by-joe-celko/#

DelGobbo, Vincent (2017), *New for SAS® 9.4: Including Text and Graphics in Your Microsoft Excel Workbooks, Part 2*, Proceedings of SAS® Global Forum (SGF) 2017 International Conference, Cary, NC: SAS Institute Inc.
https://support.sas.com/resources/papers/proceedings17/SAS0127-2017.pdf

Foley, Malachy J. (2005), *Merging vs. Joining: Comparing the DATA Step with SQL*, Proceedings of the 30th Annual SAS Users Group International (SUGI) 2005 Conference, University of North Carolina, Chapel Hill, NC, USA.
http://www.scsug.org/SCSUGProceedings/2005/Foley_Merging%20vs%20Joining%20-%20184.pdf

Kent, Paul, *SQL Joins -- The Long and The Short of It*; SAS Institute Inc., Cary, NC, USA.
http://support.sas.com/techsup/technote/ts553.html

Kerman, Jonathan (2010), *Ordering PROC FREQ Around*, NorthEast SAS Users Group (NESUG) 2010 Conference, Johns Hopkins University, S Baltimore, MD, USA.
http://www.lexjansen.com/nesug/nesug10/cc/cc16.pdf

Lafler, Kirk Paul (2013), *PROC SQL: Beyond the Basics Using SAS, Second Edition*; SAS Press.
http://support.sas.com/publishing/authors/lafler.html

Lafler, Kirk Paul and Mira Shapiro (2013), *Point-and-Click Programming Using SAS® Enterprise Guide®*, NorthEast SAS Users Group (NESUG) 2013 Regional Conference, Software Intelligence Corporation, Spring Valley, CA, USA.
http://www.lexjansen.com/nesug/nesug13/63_Final_Paper.pdf

Lafler, Kirk Paul (2012), *Exploring DATA Step Merges and PROC SQL Joins*, Proceedings of the 6th Annual SAS Global Forum (SGF) 2012 International Conference, Software Intelligence Corporation, Spring Valley, CA, USA.
http://support.sas.com/resources/papers/proceedings12/251-2012.pdf

Lafler, Kirk Paul (2012), *Exploring DATA Step Merges and PROC SQL Joins*, Proceedings of the 14th Annual Pharmaceutical SAS Users Group (PharmaSUG) 2012 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.
http://pharmasug.org/proceedings/2012/TA/PharmaSUG-2012-TA02.pdf

Lafler, Kirk Paul (2011), *Output Delivery System (ODS)– Simply the Basics*, Proceedings of the 5th Annual SAS Global Forum (SGF) 2011 International Conference, Software Intelligence Corporation, Spring Valley, CA, USA.
http://support.sas.com/resources/papers/proceedings11/273-2011.pdf

Phelps, Kent ♥ Ronda Team (2018), *Base SAS® and SAS® Enterprise Guide®: Automate Your SAS® World with Dynamic Code ~ Your Newest BFF (Best Friend Forever) in SAS*, Proceedings of SAS® Global Forum (SGF) 2018 International Conference, The SASketeers **~** *All for SAS and SAS for All!* **~** Illuminator Coaching, Inc., Des Moines, IA, USA.
https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/2857-2018.pdf

Phelps, Kent ♥ Ronda Team (2017), Hands-On Workshop (HOW): *Base SAS® and SAS® Enterprise Guide® ~ Automate Your SAS World With Dynamic Code; Your Newest BFF (Best Friend Forever) in SAS*, Proceedings of the 28th Annual MidWest SAS Users Group (MWSUG) 2017 Regional Conference, The SASketeers **~** *All for SAS and SAS for All!* **~** Illuminator Coaching, Inc., Des Moines, IA, USA.
https://www.mwsug.org/proceedings/2017/HW/MWSUG-2017-HW03.pdf

Phelps, Kent ♥ Ronda Team (2016), Hands-On Workshop: *The Joinless Join ~ The Impossible Dream Come True; Expand the Power of Base SAS® and SAS® Enterprise Guide® in a New Way*, Proceedings of the 27th Annual MidWest SAS Users Group (MWSUG) 2016 Regional Conference, The SASketeers, Des Moines, IA, USA.
http://www.mwsug.org/proceedings/2016/HW/MWSUG-2016-HW03.pdf

Phelps, Kent ♥ Ronda Team (2016), *Base SAS® and SAS® Enterprise Guide® ~ Automate Your SAS World With Dynamic Code; Your Newest BFF (Best Friend Forever) in SAS*, Proceedings of the 27th Annual MidWest SAS Users Group (MWSUG) 2016 Regional Conference, The SASketeers, Des Moines, IA, USA.
http://www.mwsug.org/proceedings/2016/TT/MWSUG-2016-TT11.pdf

Phelps, Kent ♥ Ronda Team (2015), *The Joinless Join ~ The Impossible Dream Come True; Expanding the Power of SAS® Enterprise Guide® in a New Way*, Proceedings of the 26th Annual MidWest SAS Users Group (MWSUG) 2015 Regional Conference, The SASketeers, Des Moines, IA, USA.
http://www.mwsug.org/proceedings/2015/BI/MWSUG-2015-BI-11.pdf

Phelps, Kent ♥ Ronda Team (2015), *SAS® Enterprise Guide® Base SAS® Program Nodes ~ Automating Your SAS World With a Dynamic FILENAME Statement, Dynamic Code, and the CALL EXECUTE Command; Your Newest BFF (Best Friends Forever) in SAS*, Proceedings of the 26th Annual MidWest SAS Users Group (MWSUG) 2015 Regional Conference, The SASketeers, Des Moines, IA, USA.
http://www.mwsug.org/proceedings/2015/TT/MWSUG-2015-TT-05.pdf

SAS Institute Inc. (2016), *SAS® 9.4 DATA Step Statements: Reference*; Cary, NC; SAS Institute Inc.
https://documentation.sas.com/?docsetId=lestmtsref&docsetTarget=p00hxg3x8lwivcn1f0e9axziw57y.htm&docsetVersion=9.4&locale=en#p02qkin2ubh7tcn1xael6b1o304f

TRADEMARK CITATIONS

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. The symbol, ®, indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.

DISCLAIMER

We have endeavored to provide accurate and helpful information in this SAS White Paper. The information is provided in 'Good Faith' and 'As Is' without any kind of warranty, either expressed or implied. Recipients acknowledge and agree that we and/or our company are/is not, and never will be, liable for any problems and/or damages whatsoever which may arise from the recipient's use of the information in this paper. Please refer to your specific Operating System (e.g. UNIX, Windows, or z/OS) Manual, Installation Configuration, and/or in-house Technical Support for further guidance in how to create the SAS code presented in this paper.