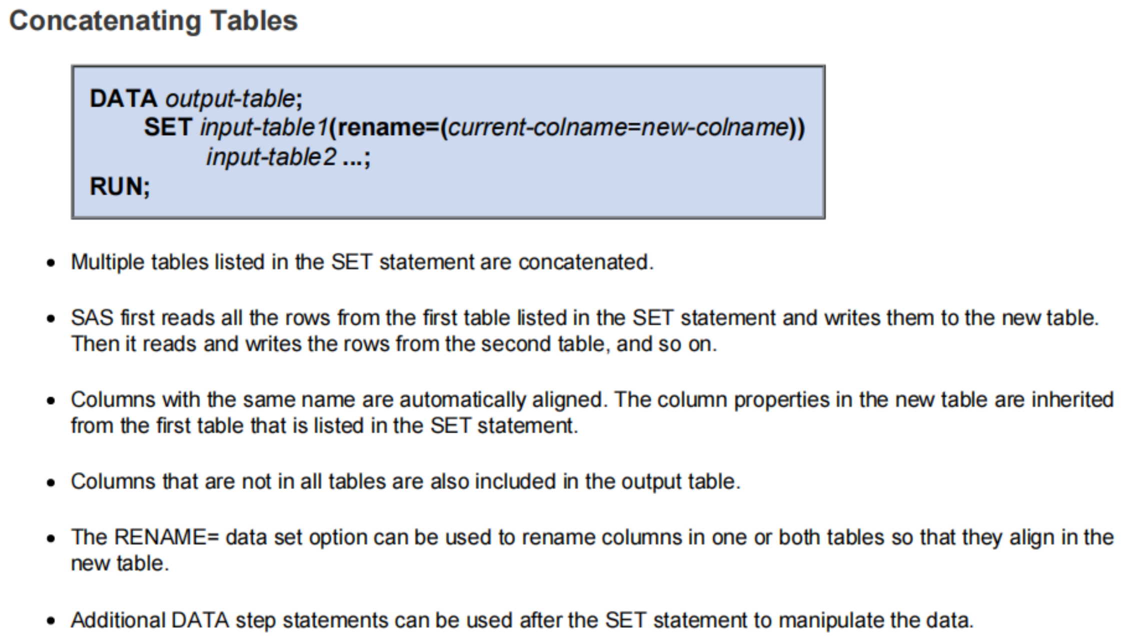
5.combinTables

March 3, 2025

0.0.1 Combining Tables

In Combining Tables we take a comprehensive look at combining tables by using the DATA step. You learn to concatenate tables, merge tables, and identify matching and nonmatching rows.

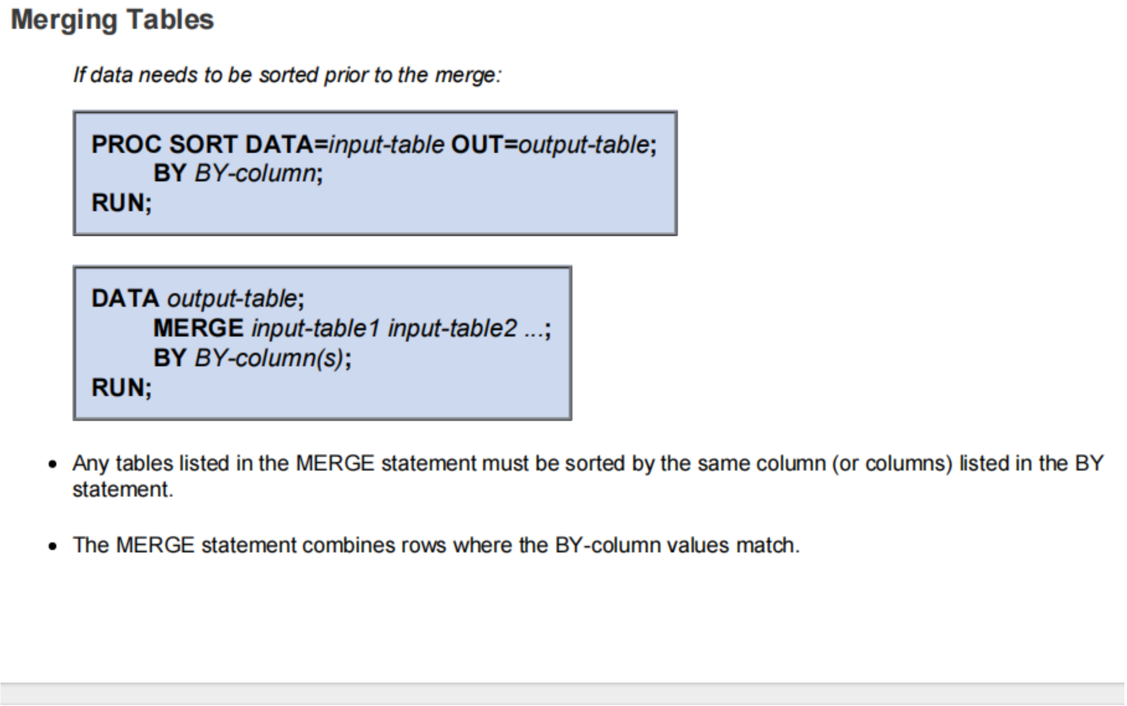


Concatenating Tables

```
DATA output-table;
      SET input-table1(rename=(current-colname=new-colname))
             input-table2 ...;
RUN;
```

- Multiple tables listed in the SET statement are concatenated.
- SAS first reads all the rows from the first table listed in the SET statement and writes them to the new table. Then it reads and writes the rows from the second table, and so on.
- Columns with the same name are automatically aligned. The column properties in the new table are inherited from the first table that is listed in the SET statement.
- Columns that are not in all tables are also included in the output table.
- The RENAME= data set option can be used to rename columns in one or both tables so that they align in the new table.
- Additional DATA step statements can be used after the SET statement to manipulate the data.

0.0.2 Merging Tables



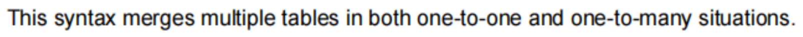
Merging Tables

If data needs to be sorted prior to the merge:

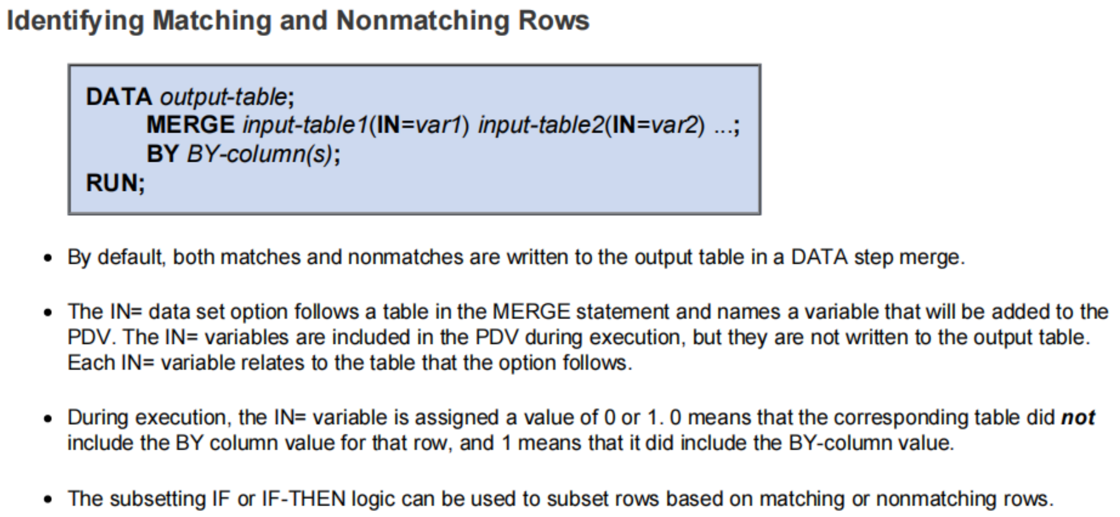
```
PROC SORT DATA=input-table OUT=output-table;
     BY BY-column;
RUN;
```

```
DATA output-table;
     MERGE input-table1 input-table2 ...;
     BY BY-column(s);
RUN;
```

- Any tables listed in the MERGE statement must be sorted by the same column (or columns) listed in the BY statement.
- The MERGE statement combines rows where the BY-column values match.



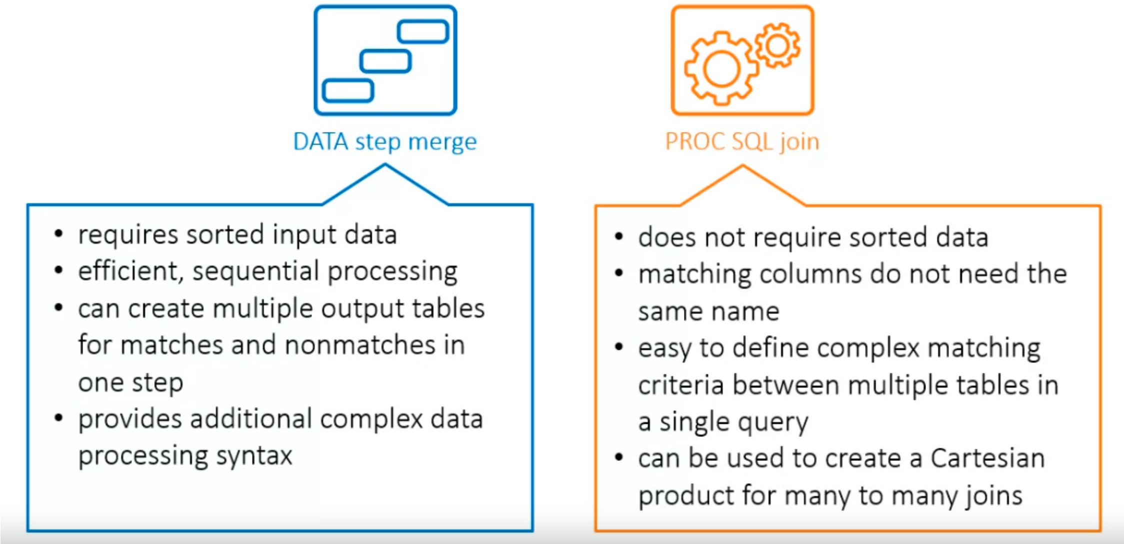
This syntax merges multiple tables in both one-to-one and one-to-many situations.



Identifying Matching and Nonmatching Rows

```
DATA output-table;
     MERGE input-table1(IN=var1) input-table2(IN=var2) ...;
     BY BY-column(s);
RUN;
```

- By default, both matches and nonmatches are written to the output table in a DATA step merge.
- The IN= data set option follows a table in the MERGE statement and names a variable that will be added to the PDV. The IN= variables are included in the PDV during execution, but they are not written to the output table. Each IN= variable relates to the table that the option follows.
- During execution, the IN= variable is assigned a value of 0 or 1. 0 means that the corresponding table did ***not*** include the BY column value for that row, and 1 means that it did include the BY-column value.
- The subsetting IF or IF-THEN logic can be used to subset rows based on matching or nonmatching rows.



DATA step merge

- requires sorted input data
- efficient, sequential processing
- can create multiple output tables for matches and nonmatches in one step
- provides additional complex data processing syntax

PROC SQL join

- does not require sorted data
- matching columns do not need the same name
- easy to define complex matching criteria between multiple tables in a single query
- can be used to create a Cartesian product for many to many joins

[ ]: