

```
/* JOINING USING SAS:--> */

/* PART-1.0:---> */

/* IMPORT FILES */
PROC IMPORT DATAFILE='/home/u63730693/my_sas/JOIN.xlsx'
    OUT=CUST_6
    DBMS=xlsx REPLACE;
    SHEET='CUST_6';
RUN;

/* COMBINING TWO DATASETS: */

/* a. TWO DATASETS HAVE COMMON COLUMNS. */

/* EG-1: */

DATA COMBINED;
SET WORK.CUST_1;
SET WORK.CUST_2;
RUN;

PROC PRINT DATA=combined;

/* b. TWO DATASETS HAVE COMMON COLUMNS WITH DIFFERENT VALUES. */

/* CRITERIA-1: */

/* IT WILL GIVE THE VALUES OF THE SECOND DATASET'S COMMON COLUMNS */

DATA COMBINED1;
SET WORK.CUST_1;
SET WORK.CUST_3;
RUN;

PROC PRINT DATA=COMBINED1;

/* CRITERIA-1.1: */

DATA COMBINED11;
SET WORK.CUST_3;
SET WORK.CUST_1;
RUN;

PROC PRINT DATA=COMBINED11;

/* HOW THIS HAPPENED? */
/* SAS HAS BUFFER AREA THAT IS PDV(PROGRAM DATA VECTOR).
SAS READ FIRST OBSERVATION FROM FIRST DATASET,
THEN SAS READ FIRST OBSERVATION FROM SECOND DATASET
AND WHATEVER VALUE EXISTS IN THE COLUMN IT REPLACE THAT VALUE. */

/* CRITERIAL-2: */

/* IT WILL APPEND THE TWO DATASET WITH NULL VALUES. */

DATA COMBINED2;
SET WORK.CUST_1
    WORK.CUST_3;
RUN;

PROC PRINT DATA=COMBINED2;
```

```
/* PART-1.1:---> */

/* PDV: */
/* IT STANDS FOR PROGRAM DATA VECTOR. */
/* IT IS A LOGICAL AREA IN THE MEMORY WHERE SAS BUILDS THE DATASETS. */
```

```
/* PART-2:---> */
```

```
/* MERGING THE TABLES: */
```

```
/* PREREQUISITES: */
/* 1. THERE SHOULD BE AT LEAST TWO TABLES. */
/* 2. BOTH/ALL THE TABLES MUST HAVE AT LEAST ONE COMMON VARIABLE. */
/* 3. BOTH/ALL OF THE TABLES MUST BE SORTED BY COMMON VARIABLE. */
```

```
/* FULL JOIN: */
```

```
/* STEP-1: SORT THE DATASET */
```

```
PROC SORT DATA=WORK.cust_1; BY CONT_NAME; RUN;
PROC SORT DATA=WORK.cust_4; BY CONT_NAME; RUN;
```

```
/* STEP-2: MERGING BY COMMON COLUMNS */
```

```
DATA COMBINS;
MERGE WORK.cust_1
      WORK.cust_4;
BY CONT_NAME;
RUN;
```

```
PROC PRINT DATA=COMBINS;
```

```
/* LEFT NULL JOIN: */
```

```
DATA COMBINS1;
MERGE WORK.cust_1(IN=A)
      WORK.cust_4(IN=B);
BY CONT_NAME;
IF A;
RUN;
```

```
PROC PRINT DATA=COMBINS1;
```

```
/* RIGHT NULL JOIN: */
```

```
DATA COMBINS2;
MERGE WORK.cust_1(IN=A)
      WORK.cust_4(IN=B);
BY CONT_NAME;
IF B;
RUN;
```

```
PROC PRINT DATA=COMBINS2;
```

```
/* INNER JOIN: */
```

```
DATA COMBINS3;
MERGE WORK.cust_1(IN=A)
      WORK.cust_4(IN=B);
BY CONT_NAME;
IF A AND B;
RUN;
```

```
PROC PRINT DATA=COMBINS3;
```

```
/* LEFT JOIN: */
```

```
DATA COMBINS4;  
MERGE WORK.cust_1(IN=A)  
      WORK.cust_4(IN=B);  
BY CONT_NAME;  
IF A OR B;  
RUN;
```

```
PROC PRINT DATA=COMBINS4;
```

```
/* PART-3:---> */
```

```
/* APPEND TWO TABLES: */
```

```
/* PROC APPEND: */  
/* THE APPEND PROCEDURE ADDS THE OBSERVATIONS FROM ONE SAS DATASET TO THE END OF ANOTHER SAS DATASET. */  
/* SYNTAX--> PROC APPEND <BASE = 'BASE-SAS-DATASET'> <DATA = SAS-DATASET-TO-APPEND> <FORCE> */  
/* BASE--> TABLE IN WHICH THE DATA HAS TO BE APPENDED. */  
/* DATA--> TABLE WHICH HAS TO BE APPENDED. */  
/* FORCE--> IF TWO TABLES DO HAVE DIFFERENT COLUMNS THEN USE 'FORCE' */
```

```
/* PROC APPEND CHANGE THE ORIGINAL DATASET. */
```

```
PROC APPEND BASE=WORK.cust_1 DATA=WORK.cust_5;  
RUN;
```

```
PROC PRINT DATA=WORK.CUST_1;  
RUN;
```

```
/* PROC APPEND WITH 'FORCE' STATEMENT. */
```

```
PROC APPEND BASE=WORK.cust_2 DATA=WORK.cust_6 FORCE;  
RUN;
```

```
PROC PRINT DATA=WORK.CUST_2;  
RUN;
```

```
/* CRITERIA-3: */
```

```
/* COMBINING ALL THE TABLES(HAVING SIMILAR INITIAL IN THEIR NAMES) IN A LIBRARY INTO ONE. */
```

```
DATA COMBINED_ALL;  
SET WORK.CUST;;  
RUN;
```

```
PROC PRINT DATA = COMBINED_ALL;
```