```
LIBNAME SSN4 '/home/debendra330/BATCH_202404/SESSION_4/A3.SAS_DATASET';
RUN;

/* JOINING AND RELATIONSHIPS IN SAS */
================================================


/* JOINING ARE OF 2 TYPES */
/* 1. VERTICAL JOIN */
/* 2. HORIZONTAL JOIN */

/* 1. VERTICAL JOIN - APPENDING OF TABLES */
==============================================
```

```
DATA MED_APPOLO;
INFILE CARDS DSD DLM='09'X;
INPUT CUSTOMER_ID Company $ GENDER $ Age STATE_CODE $ POST_CODE NO_OF_TRIPS;
CARDS;
1000032552  APPOLO  Female  26  NSW 2558    6
1000032554  APPOLO  Male    26  QLD 4575    8
1000032555  APPOLO  Male    55  NSW 2322    1
1000032584  APPOLO  Female  37  QLD 4065    1
1000032588  APPOLO  Female  25  NSW 2099    4
1000032609  APPOLO  Male    81  WA  6232    2
1000032626  APPOLO  Male    20  VIC 3169    3
1000032629  APPOLO  Male    49  VIC 3199    15
1000032666  APPOLO  Female  40  NT  870 1
;
RUN;
```

```
PROC PRINT DATA=MED_APPOLO;
RUN;
```

```
DATA MED_CIPLA;
INFILE CARDS DSD DLM='09'X;
INPUT CUSTOMER_ID Company $ GENDER $ Age STATE_CODE $ POST_CODE NO_OF_TRIPS;
CARDS;
1000032553  CIPLA   Female  26  VIC 3136    6
1000032576  CIPLA   Female  46  QLD 4551    1
1000032592  CIPLA   Female  44  QLD 4519    1
1000032595  CIPLA   Female  27  NSW 2646    1
1000032618  CIPLA   Female  39  NSW 2871    6
1000032622  CIPLA   Female  26  NSW 2820    10
1000032624  CIPLA   Male    36  NSW 2800    2
1000032628  CIPLA   Female  39  NSW 2148    1
1000032631  CIPLA   Male    61  QLD 4301    2
;
RUN;
```

```
PROC PRINT DATA=MED_CIPLA;
RUN;

/* APPENDING SCENARIO-1 */
============================
```

```
DATA MED_ALL;
SET MED_APPOLO MED_CIPLA;
RUN;
```

```
PROC PRINT DATA=MED_ALL;
RUN;

/* SET STATEMENT IS USED TO APPRND DATA TOGETHER IN SAS */
```

```
DATA MED_GENO;
INFILE CARDS DSD DLM='09'X;
INPUT CUSTOMER_ID Company $ GENDER $ Age STATE_CODE $ POST_CODE NO_OF_TRIPS;
CARDS;
1000032559  GENO    Female  34  VIC 3875    1
1000032565  GENO    Female  63  NSW 2750    2
1000032571  GENO    Female  60  VIC 3351    16
1000032573  GENO    Female  35  NSW 2069    1
1000032578  GENO    Male    44  QLD 4511    5
1000032583  GENO    Female  26  NSW 2502    4
1000032587  GENO    Female  56  QLD 4300    1
1000032605  GENO    Male    65  QLD 4305    2
1000032608  GENO    Female  33  NSW 2795    7
1000032613  GENO    Female  23  NSW 2283    1
1000032621  GENO    Male    21  NSW 2794    15
;
RUN;
```

```
PROC PRINT DATA=MED_GENO;
```

```
RUN;

/* APPENIDNG SCENARIO-2 */
============================

DATA MED_ALL_V1;
SET MED_APPOLO MED_CIPLA MED_GENO;
RUN;

PROC PRINT DATA=MED_ALL_V1;
RUN;

/* APPENDING DATA SCENARIO-3 */
================================

DATA MED_RELEGARE;
INFILE CARDS DSD DLM='09'X;
INPUT CUSTOMER_ID Company $ GENDER $ Age STATE_CODE $ POST_CODE VISITS;
CARDS;
1000032549  RELEGARE    Female  54  NSW 2580    1
1000032561  RELEGARE    Female  51  QLD 4118    3
1000032563  RELEGARE    Female  48  QLD 4504    2
1000032568  RELEGARE    Female  24  VIC 3995    1
1000032577  RELEGARE    Female  56  VIC 3214    1
1000032580  RELEGARE    Female  64  SA  5066    7
1000032597  RELEGARE    Female  43  QLD 4074    1
1000032602  RELEGARE    Male    42  NSW 2480    1
1000032606  RELEGARE    Male    43  NSW 2650    4
1000032612  RELEGARE    Male    24  QLD 4818    11
1000032616  RELEGARE    Male    25  VIC 3152    1
;
RUN;

PROC PRINT DATA=MED_RELEGARE;
RUN;

PROC PRINT DATA=MED_APPOLO;
RUN;

DATA MED_ALL_V2;
SET MED_APPOLO MED_CIPLA MED_GENO MED_RELEGARE;
RUN;

PROC PRINT DATA=MED_ALL_V2;
RUN;

DATA MED_ALL_V2;
SET MED_APPOLO MED_CIPLA MED_GENO MED_RELEGARE (RENAME=(VISITS=NO_OF_TRIPS));
RUN;

PROC PRINT DATA=MED_ALL_V2;
RUN;

/* APPENDING SCENARIO-4 */
=======================================

DATA MED_GSK;
INFILE CARDS DSD DLM='09'X;
INPUT CUSTOMER_ID Company $ GENDER $ Age STATE_CODE $ POST_CODE NO_OF_TRIPS TOWN $ SPENT_AMOUNT;
CARDS;
1000032550  GSK Male    35  VIC 3043    2   Willow Grove    160.98
1000032551  GSK Male    34  VIC 3190    12  mona vale   289.1
1000032557  GSK Female  28  QLD 4814    1   Rowville    274.66
1000032572  GSK Female  42  NSW 2533    18  warilla 666.1
1000032574  GSK Female  25  NSW 2077    7   Buderim 739.84
1000032585  GSK Female  51  NT  850 3   cobram  153.75
1000032586  GSK Male    27  NSW 2680    1   Boat Harbour    57.46
1000032589  GSK Male    27  SA  5118    3   darnum  89.72
1000032590  GSK Male    22  NSW 2650    1   Darnum  8.49
1000032610  GSK Male    73  NSW 2075    10  Delacombe   755.93
1000032615  GSK Female  45  SA  5095    3   Greensborough   198.36

;
RUN;

PROC PRINT DATA=MED_GSK;
RUN;

PROC PRINT DATA=MED_APPOLO;
RUN;

DATA MED_ALL_V3;
SET MED_APPOLO MED_CIPLA MED_GENO MED_RELEGARE (RENAME=(VISITS=NO_OF_TRIPS)) MED_GSK;
RUN;
```

```
PROC PRINT DATA=MED_ALL_V3;
RUN;

/* APPENDING SCENARIO-5 */
====================================

DATA MED_ALL;
SET MED_APPOLO;
SET MED_CIPLA;
RUN;

PROC PRINT DATA=MED_ALL;
RUN;

/* IF YOU HAVE MULTIPLE SET STATEMENTS, ONLY THE LAST SET STATEMENT WILL BE IN YOUR FINAL OUTPUT */

DATA MED_ALL;
SET MED_APPOLO;
SET MED_CIPLA;
SET MED_GSK;
RUN;

PROC PRINT DATA=MED_ALL;
RUN;

/* APPENDING SCENARIO-6 */
======================================

DATA MED_APPOLO;
INFILE CARDS DSD DLM='09'X;
INPUT CUSTOMER_ID Company $ GENDER $ Age STATE_CODE $ POST_CODE NO_OF_TRIPS;
CARDS;
1000032552  APPOLO  Female  26  NSW 2558    6
1000032554  APPOLO  Male    26  QLD 4575    8
1000032555  APPOLO  Male    55  NSW 2322    1
1000032584  APPOLO  Female  37  QLD 4065    1
1000032588  APPOLO  Female  25  NSW 2099    4
1000032609  APPOLO  Male    81  WA  6232    2
1000032626  APPOLO  Male    20  VIC 3169    3
1000032629  APPOLO  Male    49  VIC 3199    15
1000032666  APPOLO  Female  40  NT  870 1
;
RUN;

PROC PRINT DATA=MED_APPOLO;
RUN;


DATA MED_CIPLA;
INFILE CARDS DSD DLM='09'X;
INPUT CUSTOMER_ID : $11. Company $ GENDER $ Age STATE_CODE $ POST_CODE  NO_OF_TRIPS;
CARDS;
1000032553  CIPLA   Female  26  VIC 3136    6
1000032576  CIPLA   Female  46  QLD 4551    1
1000032592  CIPLA   Female  44  QLD 4519    1
1000032595  CIPLA   Female  27  NSW 2646    1
1000032618  CIPLA   Female  39  NSW 2871    6
1000032622  CIPLA   Female  26  NSW 2820    10
1000032624  CIPLA   Male    36  NSW 2800    2
1000032628  CIPLA   Female  39  NSW 2148    1
1000032631  CIPLA   Male    61  QLD 4301    2
;
RUN;

PROC PRINT DATA=MED_CIPLA;
RUN;

/* THERE IS A DATA TYPE MISMATCH IN BOTH THE TABLES */

DATA MED_ALL;
SET MED_APPOLO MED_CIPLA;
RUN;

PROC PRINT;
RUN;

/* IF THERE IS A DATA TYPE MISMATCH, WE CAN SEE THAT SAS DOES NOT APPEND THE TABLES */

/* APPENDING SCENARIO-7 */
==================================

DATA MED_APPOLO;
INFILE CARDS DSD DLM='09'X;
INPUT CUSTOMER_ID Company $ GENDER $ Age STATE_CODE $ POST_CODE NO_OF_TRIPS;
CARDS;
```

```
1000032552  APPOLO  Female  26  NSW 2558    6
1000032554  APPOLO  Male    26  QLD 4575    8
1000032555  APPOLO  Male    55  NSW 2322    1
1000032584  APPOLO  Female  37  QLD 4065    1
1000032588  APPOLO  Female  25  NSW 2099    4
1000032609  APPOLO  Male    81  WA  6232    2
1000032626  APPOLO  Male    20  VIC 3169    3
1000032629  APPOLO  Male    49  VIC 3199    15
1000032666  APPOLO  Female  40  NT  870 1
;
RUN;

PROC PRINT DATA=MED_APPOLO;
RUN;


DATA MED_CIPLA;
INFILE CARDS DSD DLM='09'X;
INPUT CUSTOMER_ID  Company $ GENDER $ Age STATE_CODE $ POST_CODE    NO_OF_TRIPS;
CARDS;
1000032553  CIPLA   Female  26  VIC 3136    6
1000032576  CIPLA   Female  46  QLD 4551    1
1000032592  CIPLA   Female  44  QLD 4519    1
1000032595  CIPLA   Female  27  NSW 2646    1
1000032618  CIPLA   Female  39  NSW 2871    6
1000032622  CIPLA   Female  26  NSW 2820    10
1000032624  CIPLA   Male    36  NSW 2800    2
1000032628  CIPLA   Female  39  NSW 2148    1
1000032631  CIPLA   Male    61  QLD 4301    2
;
RUN;

PROC PRINT DATA=MED_CIPLA;
RUN;

PROC SQL;
CREATE TABLE MED_ALL AS
SELECT * FROM MED_APPOLO
UNION ALL
SELECT * FROM MED_CIPLA;
QUIT;

PROC PRINT DATA=MED_ALL;
RUN;

/* YOU CAN USE PROC SQL TO APPEND DATASETS  */

/* APPENDING DATASET SCENARIO-8 */
=====================================

DATA MED_ALL;
SET MED_APPOLO;
RUN;

PROC APPEND BASE=MED_ALL DATA=MED_CIPLA;
RUN;

PROC PRINT DATA=MED_ALL;
RUN;

PROC APPEND BASE=MED_ALL DATA=MED_GENO;
RUN;

PROC PRINT;
RUN;

PROC PRINT DATA=MED_RELEGARE;
RUN;

PROC APPEND BASE=MED_ALL DATA=MED_RELEGARE;
RUN;

PROC PRINT;
RUN;

PROC APPEND BASE=MED_ALL DATA=MED_RELEGARE FORCE;
RUN;

PROC PRINT;
RUN;

/* WE SHOULD NOT USE FORCE UNLESS VERY NECESSARY */

/* INTERVIEW QUESTIONS */
-----------------------------------
```

```
/* 1. HOW MANY WAYS IN WHICH WE CAN APPEND DATASETS IN SAS */
/* ANS. SET STATEMENT, PROC SQL, PROC APPEND */

/* 2. WHAT WILL HAPPEN IF YOU WRITE MULTIPLE SET STATEMNETS */
/* ANS. ONLY THE LAST SET STATEMENT IS EXECUTED */

/* 3. WHY SHOULD YOU NOT USE FORCE WITH PROC APPEND */
/* ANS. WE WILL LOSE DATA WHEN FORCEFULLY APPENDING TABLES */


/* UNION VS UNION ALL */
====================================
```

```
DATA STU_SCORE1;
INFILE CARDS DSD DLM='09'X;
INPUT STU_NAME $ SUBJECT $ SCORE;
CARDS;
SIMI    SAS 453
SIMI    R   743
SIMI    PYTHON  282
SIMI    EXCEL   250
SUDIP   R   598
SUDIP   TABLEAU 459
SUDIP   PYTHON  776
SUDIP   VBA 266
SASHI   EXCEL   282
SASHI   VBA 590
SASHI   SQL 456
;
RUN;
```

```
PROC PRINT DATA=STU_SCORE1;
RUN;
```

```
DATA STU_SCORE2;
INFILE CARDS DSD DLM='09'X;
INPUT STU_NAME $ SUBJECT $ SCORE;
CARDS;
SIMI    SAS 453
SIMI    R   743
SIMI    PYTHON  282
SIMI    TABLEAU 450
RUCHIKA R   708
RUCHIKA TABLEAU 704
RUCHIKA EXCEL   562
RUCHIKA QLIKVIEW    438
PRABHAT EXCEL   751
PRABHAT VBA 749
PRABHAT SQL 880
;
RUN;
```

```
PROC PRINT DATA=STU_SCORE2;
RUN;
```

```
PROC PRINT DATA=STU_SCORE1;
RUN;

/* APPENDING DATASET USING UNION ALL */
```

```
PROC SQL NUMBER;
SELECT * FROM STU_SCORE1
UNION ALL
SELECT * FROM STU_SCORE2;
QUIT;
```

```
PROC SQL NUMBER;
SELECT * FROM STU_SCORE1
UNION
SELECT * FROM STU_SCORE2;
QUIT;

/* UNION ALL APPENDS ALL RECORDS WITHOUT CHECKING DUPLICATES */
/* UNION APPENDS RECORDS AFTER CHECKING DUPLICATES AND REMOVING THEM  FROM FINAL DATASET */

/* 2. HORIZONTAL JOIN - MERGING OF TABLES */
=======================================================

/* HORIZONTAL JOIN ARE OF 2 TYPES */
/* 1. INNER JOIN - IT TAKES COMMON RECORDS FROM BOTH LEFT AND RIGHT TABLES */
/* 2. OUTER JOIN */
/*                  A. FULL OUTER JOIN/ FULL JOIN - IT TAKES ALL RECORDS FROM BOTH TABLES */
```

```
/*                  B. UNMATCHED JOIN - IT TAKES ALL UNCOMMON RECORDS FROM BOTH TABLES */
/*                  C. LEFT OUTER JOIN/ LEFT JOIN - IT TAKES ALL RECORDS FROM LEFT TABLE AND COMMON FROM RIGHT */
/*                  D. LEFT NULL JOIN =- IT TAKES ALL RECORDS FROM LEFT TABLE AND UNCOMMON FROM RIGHT */
/*                  E. RIGHT OUTER JOIN/ RIGHT JOIN - IT TAKES ALL RECORDS FROM RIGHT TABLE AND COMMON FROM LEFT */
/*                  F. RIGHT NULL - IT TAKES ALL RECORDS FROM RIGHT WHIHC ARE NOT PRESENT IN LEFT */

DATA STU_EDUCATION;
INFILE CARDS DSD DLM='09'X;
INPUT STU_ID $ STU_NAME : $15. GENDER $ EDUCATION $;
CARDS;
S1  GAURABH MALE    BTECH
S2  NIRUPA  FEMALE  BCOM
S3  LAXMI   FEMALE  BSC
S4  BIJAY   MALE    MCA
S5  KARAN   MALE    MTECH
S6  ABHISHEK    MALE    MPHARMA
S7  DEBASISH    MALE    MBA
;
RUN;

PROC PRINT DATA=STU_EDUCATION;
RUN;

DATA STU_EXPERIENCE;
INFILE CARDS DSD DLM='09'X;
INPUT STU_ID $ YOE  COMPANY : $15. SALARY;
CARDS;
S1  4   JP MORGAN   126000
S2  5   MORGAN STANLEY  152000
S3  3   BCG 145000
S4  2   MCKENSEY    140000
S5  6   HUL 168000
S8  7   ITC 100000
S9  1   HSBC    172000
;
RUN;

PROC PRINT DATA=STU_EXPERIENCE;
RUN;

/* RULES FOR MERGING DATASETS */
/* 1. IDENTIFY LEFT AND RIGHT TABLES */
/* 2. IDENTIFY COMMON FIELDS */
/* 3. SORT THE DATASETS BEFORE MERGING BY COMMON COLUMN */

PROC SORT DATA=STU_EDUCATION;
BY STU_ID;
RUN;

PROC SORT DATA=STU_EXPERIENCE;
BY STU_ID;
RUN;

/* MERGE */

DATA STU_ALL_DETAILS;
MERGE STU_EDUCATION STU_EXPERIENCE;
BY STU_ID;
RUN;

PROC PRINT DATA=STU_ALL_DETAILS;
RUN;

/* BY DEFAULT MERGE ALWAYS GIVES FULL JOIN OUTPUT */

/* INNER JOIN */

DATA STU_ALL_DETAILS_INNER;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF A=1 AND B=1;
RUN;

PROC PRINT DATA=STU_ALL_DETAILS_INNER;
RUN;

/* IN SAS SQL */

PROC SQL NUMBER;
SELECT A.STU_ID, A.STU_NAME, A.GENDER, A.EDUCATION,
B.STU_ID, B.YOE, B.COMPANY, B.SALARY
FROM STU_EDUCATION AS A
INNER JOIN
STU_EXPERIENCE AS B
ON A.STU_ID = B.STU_ID;
```

```
QUIT;

/* FULL JOIN */

DATA STU_ALL_DETAILS;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF A=1 OR B=1;
RUN;

PROC PRINT DATA=STU_ALL_DETAILS;
RUN;

/* IN SAS SQL */

PROC SQL NUMBER;
SELECT A.STU_ID, A.STU_NAME, A.GENDER, A.EDUCATION,
B.STU_ID, B.YOE, B.COMPANY, B.SALARY
FROM STU_EDUCATION AS A
FULL JOIN
STU_EXPERIENCE AS B
ON A.STU_ID = B.STU_ID;
QUIT;

/* UNMATCHED JOIN */
DATA STU_ALL_DETAILS_UNMATCHED;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF A=0 OR B=0;
RUN;

PROC PRINT DATA=STU_ALL_DETAILS_UNMATCHED;
RUN;

/* IN SAS SQL */

PROC SQL NUMBER;
SELECT A.STU_ID, A.STU_NAME, A.GENDER, A.EDUCATION,
B.STU_ID, B.YOE, B.COMPANY, B.SALARY
FROM STU_EDUCATION AS A
FULL JOIN
STU_EXPERIENCE AS B
ON A.STU_ID = B.STU_ID
WHERE A.STU_ID IS NULL OR B.STU_ID IS NULL;
QUIT;

/* LEFT JOIN */
DATA STU_ALL_DETAILS_LEFT;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF A=1;
RUN;

PROC PRINT DATA=STU_ALL_DETAILS_LEFT;
RUN;

/* IN SAS SQL */

PROC SQL NUMBER;
SELECT A.STU_ID, A.STU_NAME, A.GENDER, A.EDUCATION,
B.STU_ID, B.YOE, B.COMPANY, B.SALARY
FROM STU_EDUCATION AS A
LEFT JOIN
STU_EXPERIENCE AS B
ON A.STU_ID = B.STU_ID;
QUIT;

/* LEFT NULL */

DATA STU_ALL_DETAILS_LEFT_NULL;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF A=1 AND B=0;
RUN;

PROC PRINT DATA=STU_ALL_DETAILS_LEFT_NULL;
RUN;

/* RIGHT JOIN */

DATA STU_ALL_DETAILS_RIGHT;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF B=1;
RUN;
```

```
PROC PRINT DATA=STU_ALL_DETAILS_RIGHT;
RUN;

/* IN SAS SQL */

PROC SQL NUMBER;
SELECT A.STU_ID, A.STU_NAME, A.GENDER, A.EDUCATION,
B.STU_ID, B.YOE, B.COMPANY, B.SALARY
FROM STU_EDUCATION AS A
RIGHT JOIN
STU_EXPERIENCE AS B
ON A.STU_ID = B.STU_ID;
QUIT;

/* RIGHT NULL */

DATA STU_ALL_DETAILS_RIGHT_NULL;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF B=1 AND A=0;
RUN;

PROC PRINT DATA=STU_ALL_DETAILS_RIGHT_NULL;
RUN;

/* IN SAS SQL */

PROC SQL NUMBER;
SELECT A.STU_ID, A.STU_NAME, A.GENDER, A.EDUCATION,
B.STU_ID, B.YOE, B.COMPANY, B.SALARY
FROM STU_EDUCATION AS A
RIGHT JOIN
STU_EXPERIENCE AS B
ON A.STU_ID = B.STU_ID
WHERE A.STU_ID IS NULL;
QUIT;

/* SIGNIFICANCE OF 1 AND 0 */
==================================

PROC SORT DATA=STU_EDUCATION;
BY STU_ID;
RUN;

PROC SORT DATA=STU_EXPERIENCE;
BY STU_ID;
RUN;

/* FULL JOIN */

DATA STU_ALL_DETAILS_FULL;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
RUN;

PROC PRINT DATA=STU_ALL_DETAILS_FULL;
RUN;

/* OR */

DATA STU_ALL_DETAILS_FULL;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF A=1 OR B=1;
RUN;

PROC PRINT DATA=STU_ALL_DETAILS_FULL;
RUN;


/* INNER JOIN */

DATA STU_ALL_DETAILS_INNER;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF A=1 AND B=1;
RUN;

PROC PRINT;
RUN;

/* UNMATCHED JOIN */

DATA STU_ALL_DETAILS_UNMATCHED;
```

```
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF A=0 OR B=0;
RUN;

PROC PRINT;
RUN;

/* LEFT JOIN */

DATA STU_ALL_DETAILS_LEFT;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF A=1;
RUN;

PROC PRINT;
RUN;

/* OR */

DATA STU_ALL_DETAILS_LEFT;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF A=1 OR B=0;
RUN;

PROC PRINT;
RUN;

/* LEFT NULL JOIN */

DATA STU_ALL_DETAILS_LEFT_NULL;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF A=1 AND B=0;
RUN;

PROC PRINT;
RUN;

/* RIGHT JOIN */
DATA STU_ALL_DETAILS_RIGHT;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF B=1;
RUN;

PROC PRINT;
RUN;

/* OR */

DATA STU_ALL_DETAILS_RIGHT;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF B=1 OR A=0;
RUN;

PROC PRINT;
RUN;

/* RIGHT NULL JOIN */

DATA STU_ALL_DETAILS_RIGHT_NULL;
MERGE STU_EDUCATION (IN=A) STU_EXPERIENCE (IN=B);
BY STU_ID;
IF B=1 AND A=0;
RUN;

PROC PRINT;
RUN;

/* HOW TO USE INTERSECT AND EXCEPT IN SAS SQL */
=========================================================

/* 1. INTERSECT */

PROC SQL NUMBER;
SELECT A.STU_ID, A.STU_NAME, A.GENDER, A.EDUCATION,
B.STU_ID, B.YOE, B.COMPANY, B.SALARY
FROM STU_EDUCATION AS A
LEFT JOIN
STU_EXPERIENCE AS B
ON A.STU_ID = B.STU_ID
```

```
INTERSECT

SELECT A.STU_ID, A.STU_NAME, A.GENDER, A.EDUCATION,
B.STU_ID, B.YOE, B.COMPANY, B.SALARY
FROM STU_EDUCATION AS A
RIGHT JOIN
STU_EXPERIENCE AS B
ON A.STU_ID = B.STU_ID;
QUIT;

/* 2. EXCEPT */

PROC SQL NUMBER;
SELECT A.STU_ID, A.STU_NAME, A.GENDER, A.EDUCATION,
B.STU_ID, B.YOE, B.COMPANY, B.SALARY
FROM STU_EDUCATION AS A
LEFT JOIN
STU_EXPERIENCE AS B
ON A.STU_ID = B.STU_ID

EXCEPT

SELECT A.STU_ID, A.STU_NAME, A.GENDER, A.EDUCATION,
B.STU_ID, B.YOE, B.COMPANY, B.SALARY
FROM STU_EDUCATION AS A
RIGHT JOIN
STU_EXPERIENCE AS B
ON A.STU_ID = B.STU_ID;
QUIT;


/* QUESTION */
/* 1. HOW TO GET INNER JOIN OUTPUT WHEN INNER JOIN IS NOT WORKING */

/* ANS. USE OF INTERSECT WITH LEFT AND RIGHT JOIN */
/* WE CAN ALSO USE FULL JOIN, LEFT JOIN AND RIGHT JOIN */

PROC SQL NUMBER;
SELECT A.STU_ID, A.STU_NAME, A.GENDER, A.EDUCATION,
B.STU_ID, B.YOE, B.COMPANY, B.SALARY
FROM STU_EDUCATION AS A
FULL JOIN
STU_EXPERIENCE AS B
ON A.STU_ID = B.STU_ID
WHERE A.STU_ID IS NOT NULL AND B.STU_ID IS NOT NULL;
QUIT;

PROC SQL NUMBER;
SELECT A.STU_ID, A.STU_NAME, A.GENDER, A.EDUCATION,
B.STU_ID, B.YOE, B.COMPANY, B.SALARY
FROM STU_EDUCATION AS A
LEFT JOIN
STU_EXPERIENCE AS B
ON A.STU_ID = B.STU_ID
WHERE B.STU_ID IS NOT NULL;
QUIT;

PROC SQL NUMBER;
SELECT A.STU_ID, A.STU_NAME, A.GENDER, A.EDUCATION,
B.STU_ID, B.YOE, B.COMPANY, B.SALARY
FROM STU_EDUCATION AS A
RIGHT JOIN
STU_EXPERIENCE AS B
ON A.STU_ID = B.STU_ID
WHERE A.STU_ID IS NOT NULL;
QUIT;

/* PRACTICAL EXAMPLES OF VERTICAL JOIN*/
=======================================================

OPTIONS VALIDVARNAME=V7;
PROC IMPORT OUT=SSN4.YATRA_TRANSACTION_Q1
            DATAFILE='/home/debendra330/BATCH_202404/SESSION_4/A1.RAW_DATA/YATRA_DATA/E BOOKING TRANSACTION DATA Q1.txt'
            DBMS=DLM REPLACE;
            DELIMITER=',';
RUN;

OPTIONS VALIDVARNAME=V7;
PROC IMPORT OUT=SSN4.YATRA_TRANSACTION_Q2
            DATAFILE='/home/debendra330/BATCH_202404/SESSION_4/A1.RAW_DATA/YATRA_DATA/E BOOKING TRANSACTION DATA Q2.txt'
            DBMS=DLM REPLACE;
            DELIMITER=',';
RUN;
```

```
OPTIONS VALIDVARNAME=V7;
PROC IMPORT OUT=SSN4.YATRA_TRANSACTION_Q3
            DATAFILE='/home/debendra330/BATCH_202404/SESSION_4/A1.RAW_DATA/YATRA_DATA/E BOOKING TRANSACTION DATA Q3.txt'
            DBMS=DLM REPLACE;
            DELIMITER=',';
RUN;

OPTIONS VALIDVARNAME=V7;
PROC IMPORT OUT=SSN4.YATRA_TRANSACTION_Q4
            DATAFILE='/home/debendra330/BATCH_202404/SESSION_4/A1.RAW_DATA/YATRA_DATA/E BOOKING TRANSACTION DATA Q4.txt'
            DBMS=DLM REPLACE;
            DELIMITER=',';
RUN;

/* NOW WE COMBINE THESE DATASETS INTO 1 MASTER DATASET */

/* SAS STATEMENT */

DATA SSN4.YATRA_MASTER;
SET SSN4.YATRA_TRANSACTION_Q1 SSN4.YATRA_TRANSACTION_Q2 SSN4.YATRA_TRANSACTION_Q3 SSN4.YATRA_TRANSACTION_Q4;
RUN;

/* PROC SQL */

PROC SQL;
CREATE TABLE SSN4.YATRA_MASTER_SQL AS
SELECT * FROM SSN4.YATRA_TRANSACTION_Q1
UNION
SELECT * FROM SSN4.YATRA_TRANSACTION_Q2
UNION
SELECT * FROM SSN4.YATRA_TRANSACTION_Q3
UNION
SELECT * FROM SSN4.YATRA_TRANSACTION_Q4;
QUIT;

/* PRACTICAL EXAMPLE OF HORIZONTAL JOIN */
==================================================

OPTIONS VALIDVARNAME=V7;
PROC IMPORT OUT=SSN4.BOOKING_MASTER
            DATAFILE='/home/debendra330/BATCH_202404/SESSION_4/A1.RAW_DATA/PRACTICAL_EXAMPLE1.xlsx'
            DBMS=XLSX REPLACE;
            SHEET='BOOKING_DETAILS';
RUN;

OPTIONS VALIDVARNAME=V7;
PROC IMPORT OUT=SSN4.CUSTOMER_MASTER
            DATAFILE='/home/debendra330/BATCH_202404/SESSION_4/A1.RAW_DATA/PRACTICAL_EXAMPLE1.xlsx'
            DBMS=XLSX REPLACE;
            SHEET='CUSTOMER_MASTER';
RUN;

/* MERGE BOOKING DETAILS AND CUSTOMER MASTER FILE */

PROC SQL NUMBER;
CREATE TABLE SSN4.CUSTOMER_FLIGHT_DETAILS AS
SELECT A.ORDER_ID, A.CUSTOMER_ID,
B.CUSTOMER_NAME, B.CUSTOMER_TYPE, B.DOB, B.EMAIL_ID,
A.BOOKING_DATE, A.NO_OF_TRAVELLERS, A.FLIGHT_NAME, A.FLIGHT_FARE, A.PAYMENT_MODE, A.PAYMENT_STATUS
FROM SSN4.BOOKING_MASTER AS A
LEFT JOIN
SSN4.CUSTOMER_MASTER AS B
ON A.CUSTOMER_ID = B.CUSTOMER_ID;
QUIT;


/* FIND CUSTOMER WHOSE PERSONAL INFORMATION IS MISSING */

PROC SQL NUMBER;
CREATE TABLE SSN4.CUSTOMER_FLIGHT_DETAILS AS
SELECT A.ORDER_ID, A.CUSTOMER_ID,
B.CUSTOMER_NAME, B.CUSTOMER_TYPE, B.DOB, B.EMAIL_ID,
A.BOOKING_DATE, A.NO_OF_TRAVELLERS, A.FLIGHT_NAME, A.FLIGHT_FARE, A.PAYMENT_MODE, A.PAYMENT_STATUS
FROM SSN4.BOOKING_MASTER AS A
LEFT JOIN
SSN4.CUSTOMER_MASTER AS B
ON A.CUSTOMER_ID = B.CUSTOMER_ID
WHERE B.CUSTOMER_ID IS NULL;
QUIT;

/* GET CUSTOMER DETAILS WHOSE PAYMENT_STATUS IS NO AND THEY ARE FROM HSBC CC HOLDER.GET THEIR NAME,DOB AND EMAIL ADDRESS */


PROC SQL NUMBER;
CREATE TABLE SSN4.CUSTOMER_FLIGHT_DETAILS AS
```

```
SELECT
B.CUSTOMER_NAME, B.CUSTOMER_TYPE, B.DOB, B.EMAIL_ID
FROM SSN4.BOOKING_MASTER AS A
LEFT JOIN
SSN4.CUSTOMER_MASTER AS B
ON A.CUSTOMER_ID = B.CUSTOMER_ID
WHERE A.PAYMENT_STATUS='NO' AND A.PAYMENT_MODE='HSBC CC';
QUIT;

/* FIND CUSTOMER WHO TRAVELS IN AIR INDIA AND THEIR FULL DETAILS */

PROC SQL NUMBER;
CREATE TABLE SSN4.CUSTOMER_FLIGHT_DETAILS AS
SELECT A.ORDER_ID, A.CUSTOMER_ID,
B.CUSTOMER_NAME, B.CUSTOMER_TYPE, B.DOB, B.EMAIL_ID,
A.BOOKING_DATE, A.NO_OF_TRAVELLERS, A.FLIGHT_NAME, A.FLIGHT_FARE, A.PAYMENT_MODE, A.PAYMENT_STATUS
FROM SSN4.BOOKING_MASTER AS A
LEFT JOIN
SSN4.CUSTOMER_MASTER AS B
ON A.CUSTOMER_ID = B.CUSTOMER_ID
WHERE A.FLIGHT_NAME = 'Air India';
QUIT;

/* ONE MORE EXAMPLE */

DATA ACCOUNT_PERFORMANCE_202001;
INFILE CARDS DSD DLM='09'X;
INPUT ACCOUNT_NUMBER STATUS $ YEARMO ;
CARDS;
1000032547  D   202001
1000032548  D   202001
1000032549  ND  202001
1000032550  ND  202001
1000032551  ND  202001
1000032552  ND  202001
1000032553  D   202001
1000032554  D   202001
1000032555  D   202001
1000032556  ND  202001
1000032557  ND  202001
1000032558  ND  202001
1000032559  D   202001
1000032560  D   202001
1000032561  D   202001
1000032562  D   202001
1000032563  ND  202001
1000032564  ND  202001
1000032565  ND  202001
1000032566  ND  202001
1000032567  ND  202001
1000032568  ND  202001
;
RUN;


DATA ACCOUNT_PERFORMANCE_202002;
INFILE CARDS DSD DLM='09'X;
INPUT ACCOUNT_NUMBER STATUS $ YEARMO ;
CARDS;
1000032547  ND  202002
1000032548  D   202002
1000032549  ND  202002
1000032550  D   202002
1000032551  D   202002
1000032552  ND  202002
1000032553  D   202002
1000032554  ND  202002
1000032555  ND  202002
1000032556  ND  202002
1000032557  D   202002
1000032558  ND  202002
1000032559  D   202002
1000032560  D   202002
1000032561  D   202002
1000032562  D   202002
1000032563  D   202002
1000032569  ND  202002
1000032570  ND  202002
1000032571  ND  202002
1000032572  ND  202002
1000032573  ND  202002
1000032574  ND  202002
;
RUN;
```

```
PROC SORT DATA=account_performance_202001;
BY ACCOUNT_NUMBER;
RUN;

PROC SORT DATA=account_performance_202002;
BY ACCOUNT_NUMBER;
RUN;

/* FULL JOIN */
DATA DEFAULT_ANALYSIS;
MERGE account_performance_202001 (IN=A) account_performance_202002 (IN=B);
BY ACCOUNT_NUMBER;
RUN;

PROC PRINT;
RUN;

/* WE SHOULD NOT GO FOR SAS MERGING IF WE HAVE THE SAME COLUMN NAMES IN BOTH DATASET */

PROC SQL NUMBER;
SELECT A.ACCOUNT_NUMBER FORMAT=10.,
A.STATUS AS JAN_STATUS,
A.YEARMO AS JAN,
B.ACCOUNT_NUMBER FORMAT=10.,
B.STATUS AS FEB_STATUS,
B.YEARMO AS FEB
FROM ACCOUNT_PERFORMANCE_202001 AS A
FULL JOIN
ACCOUNT_PERFORMANCE_202002 AS B
ON A.ACCOUNT_NUMBER = B.ACCOUNT_NUMBER;
QUIT;

/* MULTI-COLUMN JOIN */
========================================

DATA PROD_UNITS;
INFILE CARDS DSD DLM='09'X;
INPUT PRODUCT $ CITY : $15. UNITS;
CARDS;
APPLE   BANGALORE   16100
APPLE   CHENNAI 89100
APPLE   NEW DELHI   72700
APPLE   MUMBAI  77700
APPLE   HYDERABAD   12500
DELL    BANGALORE   47600
DELL    CHENNAI 84200
DELL    NEW DELHI   44400
DELL    MUMBAI  33800
DELL    HYDERABAD   63000
HP  BANGALORE   28800
HP  CHENNAI 55300
HP  NEW DELHI   86900
HP  MUMBAI  28700
HP  HYDERABAD   44900
;
RUN;

DATA PROD_PRICE;
INFILE CARDS DSD DLM='09'X;
INPUT PRODUCT $ CITY : $15. PRICE;
CARDS;
APPLE   BANGALORE   90000
APPLE   CHENNAI 92000
APPLE   NEW DELHI   88000
APPLE   MUMBAI  89000
APPLE   HYDERABAD   95000
DELL    BANGALORE   56000
DELL    CHENNAI 50000
DELL    NEW DELHI   49000
DELL    MUMBAI  50000
DELL    HYDERABAD   67000
ASUS    BANGALORE   69000
ASUS    CHENNAI 45000
ASUS    NEW DELHI   78000
ASUS    MUMBAI  89000
ASUS    HYDERABAD   67000
;
RUN;

/* SORT */

PROC SORT DATA=PROD_UNITS;
BY PRODUCT CITY;
RUN;
```

```
PROC SORT DATA=PROD_PRICE;
BY PRODUCT CITY;
RUN;

/* MERGING BOTH DATASETS */

DATA PROD_ALL_DETAILS;
MERGE PROD_UNITS (IN=A) PROD_PRICE (IN=B);
BY PRODUCT CITY;
SALES = UNITS*PRICE;
RUN;

PROC PRINT;
RUN;

/* IN SAS SQL */

PROC SQL NUMBER;
SELECT A.PRODUCT, A.CITY, A.UNITS,
B.PRODUCT, B.CITY, B.PRICE,
(A.UNITS*B.PRICE) AS SALES FORMAT=DOLLAR20.
FROM PROD_UNITS AS A
FULL JOIN
PROD_PRICE AS B
ON A.PRODUCT=B.PRODUCT
AND
A.CITY = B.CITY;
QUIT;

/* SELF- JOIN */
============================

DATA EMP_MANAGER;
INFILE CARDS DSD DLM='09'X;
INPUT EMP_NAME $ EMP_ID MANAGER_NAME $ MANAGER_ID ;
CARDS;
BLAKE   7698    KING    7839
CLARK   7782    KING    7839
JONES   7566    KING    7839
MARTIN  7654    BLAKE   7698
ALLEN   7499    BLAKE   7698
TURNER  7844    BLAKE   7698
JAMES   7900    BLAKE   7698
WARD    7521    BLAKE   7698
FORD    7902    JONES   7566
SMITH   7369    FORD    7902
SCOTT   7788    JONES   7566
ADAMS   7876    SCOTT   7788
MILLER  7934    CLARK   7782
;
RUN;

PROC PRINT DATA=EMP_MANAGER;
RUN;

PROC SQL NUMBER;
SELECT A.EMP_NAME,
A.EMP_ID,
A.MANAGER_NAME,
A.MANAGER_ID,
B.MANAGER_NAME AS MANAGERS_MANAGER_NAME,
B.MANAGER_ID AS MANAGERS_MANAGER_ID
FROM EMP_MANAGER AS A
LEFT JOIN
EMP_MANAGER AS B
ON A.MANAGER_ID = B.EMP_ID;
QUIT;

/* CROSS JOIN */
================================

DATA PROD_DETAILS;
INFILE CARDS DSD DLM='09'X;
INPUT PROD_ID $ PROD_NAME $;
CARDS;
P1  APPLE
P2  BANANA
P3  MANGO
P4  GRAPES
;
RUN;

DATA STU_DETAILS;
INFILE CARDS DSD DLM='09'X;
INPUT STU_ID $ STU_NAME $;
```

```
CARDS;
S1  RAJ
S2  GAUTAM
S3  KOMAL
S4  KAVYA
;
RUN;

PROC PRINT DATA=PROD_DETAILS;
RUN;

PROC PRINT DATA=STU_DETAILS;
RUN;

DATA ALL_DETAILS;
MERGE PROD_DETAILS STU_DETAILS;
RUN;

/* SAS MERGE STATEMENT CANNOT PRODUCE CROSS JOIN */

PROC SQL NUMBER;
SELECT PROD_ID, PROD_NAME, STU_ID, STU_NAME
FROM PROD_DETAILS
CROSS JOIN
STU_DETAILS;
QUIT;

/* NULL JOIN */
================================

DATA CUSTOMER_INFORMATION_P1;
INFILE CARDS DSD DLM='09'X;
INPUT CUSTOMER_ID COMPANY $ GENDER $ AGE;
DATALINES;
1000032547  MED +   Female  29
1000032548  DR.REDDYS   Female  39
    RELEGARE    Female  54
1000032550  GSK Male    35
1000032551  GSK Male    34
1000032552  APPOLO  Female  26
1000032553  CIPLA   Female  26
    APPOLO  Male    26
1000032555  APPOLO  Male    55
1000032556  MED +   Female  61
1000032557  GSK Female  28
1000032558  MED +   Male    57
    GENO    Female  34
1000032560  DR.REDDYS   Female  28
;
RUN;

PROC PRINT DATA=CUSTOMER_INFORMATION_P1;
RUN;

DATA CUSTOMER_INFORMATION_P2;
INFILE CARDS DSD DLM='09'X;
INPUT CUSTOMER_ID TOWN $ NO_OF_TRIPS SPENT_AMOUNT;
CARDS;
1000032547  Kyabram 4   69.37
1000032548  Hawthorn East   17  498.5
1000032549  Revesby 1   1.56
1000032550  Willow Grove    2   160.98
1000032551  mona vale   12  289.1
1000032552  Rowville    6   690.45
1000032553  MINNAMURRA  6   160.89
    Lake Illawarra  8   253.23
    Vermont South   1   69.77
1000032556  Blackburn Nth   2   232.31
1000032557  Rowville    1   274.66
    Box Hill North  1   42.83
1000032559  FOREST HILL 1   79.48
1000032560  BERWICK 1   26.88
;
RUN;

PROC PRINT DATA=CUSTOMER_INFORMATION_P2;
RUN;

PROC SORT DATA=CUSTOMER_INFORMATION_P1;
BY CUSTOMER_ID;
RUN;

PROC SORT DATA=CUSTOMER_INFORMATION_P2;
BY CUSTOMER_ID;
RUN;
```

```sas
DATA CUSTOMER_DETAILS;
MERGE CUSTOMER_INFORMATION_P1 CUSTOMER_INFORMATION_P2;
BY CUSTOMER_ID;
RUN;

/* EVEN IN PROC SQL, NULL JOIN DOES NOT WORK IN SAS */
/* WE PREFER SQL OR PYTHON FOR NULL JOIN */
```