

Q1. WHAT ARE SAS PROGRAMMING STEP?**ANS**

1. DATA STEP -----CREATE SAS DATASET
2. PROC STEP-----PRINT DATASET OUTPUT

Q2. WHHAT IS INFORMAT AND FORMAT?**ANS**

INFORMAT TO READ DATA INTO SAS FORMAT
 FORMAT WRITE DATA INTO SAS FORMAT

Q3. DIFFERENCE BETWEEN IF AND WHERE STATEMEN?**ANS**

1. WHERE STATEMENT CAN BE USED IN BOTH DATA STEP AND PROC STEP
2. IF STATEMENT CAN BE USED IN DATA STEP
3. WHERE PROCESS DATA FASTER THAN IF STTAEMENT
4. IN SAS MERGING WHERE CONDITION APPLIES BEFORE MERGING WHEREAS IF CONDITION APPLIES AFTER MERGING

Q4. HOW DO YOU SOURCE DATA INTO SAS PLATFORM?**ANS**

1. I SOURCE DATA FROM DBMS SYSTEM AND FLAT FILES
 TO GET DATA FROM DBMS SYSTEM LIKE ORACLE OR SQL DATABASE

```

■ PROC SQL;
CONNECT TO ORACLE (USER=DEVD PASSWORD=DV@1234 PATH=DV_DATA);
CREATE TABLE SSN3.MED_APPOLO
AS
SELECT * FROM CONNECT TO ORACLE
(SELECT
CUSTOMER_ID,
COMPANY,
GENDER,
AGE,
STATE_CODE
FROM MED_2020
WHERE COMPANY='APPOLO');
QUIT;

```

```

■ PROC SQL;
CONNECT TO SQL (USER=DEVD PASSWORD=DV@1234 PATH=DV_DATA);
CREATE TABLE SSN3.MED_APPOLO
AS
SELECT * FROM CONNECT TO SQL
(SELECT
CUSTOMER_ID,
COMPANY,
GENDER,
AGE,
STATE_CODE
FROM MED_2020
WHERE COMPANY='APPOLO');
QUIT;

```

2. I SOURCE DATA FROM FLAT FILES LIKE .CSV, .TXT AND .XLSX USING

FROM XLSX FILE

```
PROC IMPORT OUT=SSN3.MED_APPOLO
```

```
DATAFILE='D:\BATCH 202102\MONTH-2\3.SAS BASE AND ADVANCED PROGRAMMING\SESSION-3\2.Raw Data\1.Excel File\2.MED_STORE_BY_COMPANY.XLSX'
```

```
DBMS=XLSX REPLACE;
```

```
SHEET='MED_APPOLO';
```

```
RUN;
```

FROM TXT FILE

```
PROC IMPORT OUT=SSN3.MED_TXT
```

```
DATAFILE='D:\BATCH 202102\MONTH-2\3.SAS BASE AND ADVANCED PROGRAMMING\SESSION-3\2.Raw
```

```
Data\2.Text File\MED_New_2016.TXT'
```

```
DBMS=DLM REPLACE;
```

```
DELIMITER=';';
```

```
RUN;
```

FROM CSV FILE

```
PROC IMPORT OUT=SSN3.MED_CSV
```

```
DATAFILE='D:\BATCH 202102\MONTH-2\3.SAS BASE AND ADVANCED PROGRAMMING\SESSION-3\2.Raw
```

```
Data\2.Text File\MED_New_2016.CSV'
```

```
DBMS=CSV REPLACE;
```

```
RUN;
```

Q5. HOW DO YOU EXPORT DATA FROM SAS PLATFORM?

ANS

EXPORT TO XLSX FILE

```
PROC EXPORT DATA=SSN3.MED_APPOLO
```

```
OUTFILE='D:\BATCH 202102\MONTH-2\3.SAS BASE AND ADVANCED PROGRAMMING\SESSION- 3\5.SAS
```

```
Output\MED_EXPORT_2021.XLSX'
```

```
DBMS=XLSX REPLACE;
```

```
SHEET='MED_APPOLO';
```

```
RUN;
```

EXPORT TO TXT FILE

```
PROC EXPORT DATA=SSN3.MED_APPOLO
```

```
OUTFILE='D:\BATCH 202102\MONTH-2\3.SAS BASE AND ADVANCED PROGRAMMING\SESSION-3\5.SAS
```

```
Output\MED_APPOLO.TXT'
```

```
DBMS=DLM REPLACE;
```

```
DELIMITER=';';
```

```
RUN;
```

EXPORT TO CSV FILE

```
PROC EXPORT DATA=SSN3.MED_GSK
```

```
OUTFILE='D:\BATCH 202102\MONTH-2\3.SAS BASE AND ADVANCED PROGRAMMING\SESSION-3\5.SAS
```

```
Output\MED_GSK.CSV'
```

```
DBMS=CSV REPLACE;
```

```
RUN;
```

Q6. DIFFERENCE BETWEEN INFILE AND FILE STATEMENT?

ANS

INFILE STATEMENT TO READ DATA FROM EXTERNAL PATH

FILE STATEMENT TO WRITE DATA TO EXTERNAL PATH

Q7. OPTIONS THAT CAN BE USED IN INFILE STATEMENT?

ANS

DSD

DLM

MISSEVER

TRUNCOBER

FIRSTOBS=

LRECL=

Q8. WHAT IS THE DIFFERENCE BETWEEN @ AND @@?

ANS

'@' IS A POINTER INPUT WHICH SPECIFIES THE COLUMN NUMBER OR INCASE OF A TEXT POINTER, YOU CAN SPECIFY WHICH COLUMN TO BE READ AND HOW

THE '@@' OPTION IS USED TO HOLD SAS ON THE CURRENT LINE IF THERE ARE MULTIPLE OBSERVATIONS IN A SINGLE LINE

Q9. HOW TO CREATE A NEW DATASET BY SELECTING VARIABLES FROM EXISTING DATASET?

ANS

SET STATEMENT TO READ DATA FROM EXISTING DATASET TO A NEW DATASET
KEEP STATEMENT TO SELECT VARIABLES TO READ TO A NEW DATASET

EX:

```
DATA SSN2.MED_SELECT_V1;
SET SSN2.MED_2021;
KEEP CUSTOMER_ID COMPANY GENDER AGE STATE_CODE SPENT_AMOUNT;
RUN;
```

Q10. HOW TO REPOSITION VARIABLES IN SAS PROGRAMMING?

ANS

USING RETAIN STATEMENT BEFORE SET STATEMENT WITH VARIABLES

EX:

```
DATA STU_SCORE_DV_V1;
RETAIN STU_NAME EXCEL VBA SQL SAS PYTHON R QLIK TABLEAU POWER_BI ML AI;
SET STU_SCORE_DV;
RUN;
```

Q11. CAN WE USE KEEP AS A STATEMENT AND AS AN OPTION?

ANS

YES, WE CAN
HOWEVER, KEEP WITH STATEMENT PROCESS DATA FASTER. PROGRAM DEMAND MORE IN DATA STATEMENT TO USE KEEP AS AN OPTION WITH DATA STATEMENT

Q12. WHAT IS THE DIFFERENCE BETWEEN IF THEN AND THEN DO STATEMENT?

ANS

IF THEN STATEMENT CAN CREATE ONE CONDITIONAL VARIABLES
IF THEN DO STATEMENT CAN CREATE MULTIPLE CONDITIONAL VARIABLES

Q13. WHAT IS THE DIFFERENCE OUT= AND OUTPUT OPTIONS IN SAS PROGRAMMING?

ANS

OUT= TO CREATE NEW DATASET WITHOUT CHANGING TO MASTER DATASET
OUTPUT STATEMENT CREATES DATASET BASED ON CONDITION MEETS

EX:

```
OUT=
PROC SORT DATA=SSN2.STAFF OUT=SSN2.STAFF_V1;
BY JOBTITLE;
RUN;
```

OUTPUT

```
DATA MED_APPOLO MED_CIPLA MED_RELEGARE MED_GSK MED_GENO MED_OTHERs;
SET SSN2.MED_2021;
KEEP CUSTOMER_ID COMPANY GENDER AGE STATE_CODE SPENT_AMOUNT;
IF COMPANY='APPOLO' THEN OUTPUT MED_APPOLO;
ELSE IF COMPANY='CIPLA' THEN OUTPUT MED_CIPLA;
ELSE IF COMPANY='GSK' THEN OUTPUT MED_GSK;
ELSE IF COMPANY='GENO' THEN OUTPUT MED_GENO;
ELSE IF COMPANY='RELEGARE' THEN OUTPUT MED_RELEGARE;
ELSE OUTPUT MED_OTHERs;
RUN;
```

Q14. HOW TO REMOVE DUPLICATES IN SAS DATASET?

ANS

1. PROC SORT WITH NODUP OR NODUPKEY OPTION
2. PROC SQL WITH DISTINCT CLAUSE
3. USING FIRST. AND LAST.
4. PROC SQL AND SELECT ALL FIELDS INTO GROUP BY CLAUSE

Q15. HOW TO APPEND DATASETS IN SAS?

ANS

1. SET STATEMENT
2. PROC SQL UNION ALL
3. PROC APPEND

Q16. HOW TO APPEND DATASETS IN SAS?

ANS

1. SET STATEMENT
2. PROC SQL UNION ALL
3. PROC APPEND

Q17. HOW TO JOIN DATASETS IN SAS?

ANS

1. USING SAS MERGING
2. USING SAS SQL JOIN

Q18. HOW SAS SQL JOIN DIFFER TO SAS MERGING?

ANS

1. SAS MERGING CAN NOT DO MANY TO MANY RELATIONSHIPS. HENCE, WE PREFER SAS SQL JOIN
2. SAS MERGING CAN NOT DO CARTESIAN JOIN IF ALL THE DATASETS HAVING DIFFERENT MATCHING COLUMNS. WE CAN DO THAT EASILY DEFINING THE DATA MODEL IN SAS SQL JOIN
3. SAS MERGING SORT DATASETS BY COMMON VARIABLES BEFORE MERGING. SAS SQL JOIN DOES NOT NEED THAT

Q19. EXPLAIN SAS JOINING/MERGING?

ANS

JOINS ARE 2 TYPES







1. **VERTICAL JOIN** (APPENDING TABLES)
2. **HORIZONTAL JOIN** (MERGING TABLES)

VERTICAL JOIN- WE USE UNION ALL OR UNION QUERY

HORIZONTAL JOIN- WE USE JOININGS LIKE

INNER JOIN- 

OUTER JOIN-

- FULL OUTER JOIN 
 - o UN-MATCHED JOIN FROM FULL JOIN QUERY 
- LEFT OUTER JOIN 
 - o LEFT NULL JOIN FROM LEFT JOIN QUERY 
- RIGHT OUTER JOIN 
 - o RIGHT NULL JOIN FROM RIGHT JOIN QUERY 

Q20. DIFFERENCE BETWEEN UNION VS UNION ALL?

ANS

UNION QUERY APPENDS TABLES WITH UNIQUE VALUES

UNION ALL QUERY APPENDS TABLES WITH IRRESPECTIVE OF DUPLICATE VALUES

Q21. SAS MERGING IN= SIGNIFICANCE OF 1 AND 0?

ANS

1. **IN=** REPRESENT TABLE ALIAS DEFINING LEFT TABLE AND RIGHT TABLE IN SAS MERGING
2. **IN=A** REPRESENTS TABLE-1 **IN=B** REPRESENT TABLE-2
 - a. IF **A=1 AND B=1** REPRESENTS INNER JOIN
 - b. IF **A=1 OR B=1** REPRESENTS FULL JOIN
 - c. IF **A=1 OR B=0** REPRESENTS LEFT JOIN
 - d. IF **A=1 AND B=0** REPRESENTS LEFT NULL JOIN
 - e. IF **A=0 OR B=1** REPRESENTS RIGHT JOIN
 - f. IF **A=0 AND B=1** REPRESENTS RIGHT NULL JOIN
 - g. IF **A=0 OR B=0** REPRESENTS UN-MATCHED JOIN

Q22. IF INNER JOIN IS NOT WORKING IN SQL HOW TO GET INNER JOIN VALUES?**ANS****SOLUTION-1** (FULL JOIN WITH NOT NULL TO COMMON FIELD)

```
PROC SQL;  
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;
```

SOLUTION-2 (LEFT JOIN WITH RIGHT TABLE COMMON FIELD NOT NULL)

```
PROC SQL;  
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;
```

SOLUTION-3 (RIGHT JOIN WITH LEFT TABLE COMMON FIELD NOT NULL)

```
PROC SQL;  
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;
```

SOLUTION-4 (RIGHT JOIN TO LEFT JOIN WITH INTERSECT)

```
PROC SQL;
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;
```

Q23.FIND RELATIONSHIP AND NUMBER OF RECORDS CREATES IN JOINING?

ANS

PROD	YEAR	SALES		PROD	CITY	MS
APPLE	2018	22900		APPLE	BANGALORE	30%
APPLE	2019	54100		APPLE	MUMBAI	30%
APPLE	2020	77100		APPLE	CHENNAI	20%
DEL	2018	76800		DEL	BANGALORE	30%
DEL	2019	74200		DEL	MUMBAI	20%
DEL	2020	66000		DEL	CHENNAI	40%
HP	2018	22800		DEL	DELHI	10%
HP	2019	50500		LENOVO	CHENNAI	50%
HP	2020	53600		LENOVO	DELHI	50%

RELATIONSHIP- MANY TO MANY

INNER JOIN- 21

FULL JOIN- 26

UN-MATCHED-5

LEFT JOIN-24

LEFT NULL-3

RIGHT JOIN-23

RIGHT NULL-2

Q24.FIND RECORDS CREATES IN JOINING TABLE-1 AND TABLE-2?

ANS

TABLE-1	TABLE-2
A1	A1
A1	A1
Z1	A1
X1	Z1
Y1	Z1
Y2	B1
Y1	C1

INNER JOIN-8

FULL JOIN- 14

UN-MATCHED-6

LEFT JOIN-12

LEFT NULL-4

RIGHT JOIN-10

RIGHT NULL-2

Q25. HAVE YOU DESIGNED ANY DATA MODEL? IF YES, WHAT IS YOUR APPROACH?

ANS

BEFORE DESIGNING DATA MODEL, WE MUST IDENTIFY THE FACT AND DIMENSION TABLES THEN UNDERSTAND THE BEST DATA MODEL TO FIT IN LIKE

- STAR SCHEMA
- SNOWFLAKES

Q26. HAVE YOU EVER FACED CHALLENGES DEISGNING DATA MODEL? IF YES, PLEASE EXPLAIN?

ANS

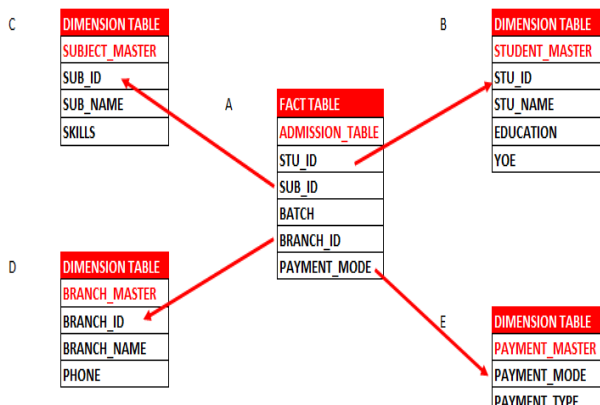
COMPLEXITY IN DATA MODEL DESIGN COMES, WHERE MULIPLE SOURCE FILES TO BE MAPPED AND SOURCE FILES INFORMATION IS NOT IN STRUCTURED FORM. SO SOMETIMES THIS OUTPUT IN MANY TO MANY RELATIONSHIP VALUES AND IMPACT IN DUPLICATE RECORDS

SO BEFORE DESIGNING THE DATA MODEL WE CHECK DATA STRUCTURE FIRST, THEN VALUES AND RELATIONSHIP AND THE TYPE OF DATA MODEL IS GOING DESIGN FOR

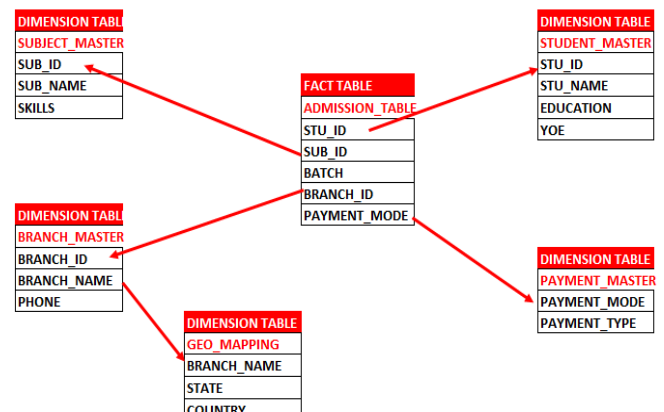
Q27. IDENTIFY THE DATA MODEL?

ANS

STAR SCHEMA



SNOWFLAKES



Q28. HOW TO FIND EMPLOYEE TO EMPLOYEE MANAGER AND MANAGERS MANAGER?**ANS**

	EMP_NAME	EMP_ID	MANAGER_NAME	MANAGER_ID
1	BLAKE	7698	KING	7839
2	CLARK	7782	KING	7839
3	JONES	7566	KING	7839
4	MARTIN	7654	BLAKE	7698
5	ALLEN	7499	BLAKE	7698
6	TURNER	7844	BLAKE	7698
7	JAMES	7900	BLAKE	7698
8	WARD	7521	BLAKE	7698
9	FORD	7902	JONES	7566
10	SMITH	7369	FORD	7902
11	SCOTT	7788	JONES	7566
12	ADAMS	7876	SCOTT	7788
13	MILLER	7934	CLARK	7782

```

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

QUIT;

```

Q29. WHAT IS THE USE OF SUB-QUERIES? CAN WE USE ORDER BY IN SUB-QUERY?**ANS**

A SUBQUERY IS USED TO RETURN DATA THAT WILL BE USED IN THE MAIN QUERY AS A CONDITION TO FURTHER RESTRICT THE DATA TO BE RETRIEVED

WE CAN NOT USE ORDER BY CLAUSE IN SUB-QUERIES

Q30. WHAT IS PDV?

- A.** PDV STANDS FOR PROGRAM DATA VECTOR. IT MAINLY HELPS TO FIND THE BACKEND PROGRAM RUNNING IN SAS SYSTEM
- B.** PDV STANDS FOR PROGRAM DATA VECTOR. IT MAINLY HELPS TO FIND THE BACKEND PROGRAM RUNNING IN SAS SYSTEM AND THROUGHES ERROR INTO LOG WINDOW AS `_ERROR_ = AND _N_`
- C.** PDV STANDS FOR PROGRAM DATA VECTOR. IT MAINLY HELPS TO FIND THE BACKEND PROGRAM RUNNING IN SAS SYSTEM AND THROUGHES ERROR INTO LOG WINDOW AS `_ERROR_ = 1 AND _N_ = 10` AS ERROR IN LINE NO 10
- D.** PDV STANDS FOR PROGRAM DATA VECTOR. IT GRAPHS THE BACKEND PROGRAMMING IN SAS

ANS- C**Q31. WHAT IS THE DIFFERENCE BETWEEN PRIMARY KEY AND UNIQUE KEY?****ANS**

PRIMARY KEY AND UNIQUE KEY BOTH ARE UNIQUE IN NATURE
HOWEVER, UNIQUE CAN HAVE A NULL VALUE

Q32. WHAT IS THE DIFFERENCE BETWEEN PRIMARY KEY AND UNIQUE KEY?**ANS**

PRIMARY KEY AND UNIQUE KEY BOTH ARE UNIQUE IN NATURE
HOWEVER, UNIQUE CAN HAVE A NULL VALUE

Q33. HOW TO GET DATE DIFFERENCE AND DATE ADD IN SAS?**ANS****INTCK** FUNCTION TO GET DATE DIFFERENCE IN DAY, WEEKS, MONTHS, AND YEAR GAP

EX:

```
DATA DATE_DIFFERENCE;
START_DT='01DEC2020'D;
END_DT='31JAN2021'D;
WEEK_GAP=INTCK('WEEK',START_DT,END_DT);
MONTHS_GAP=INTCK('MONTH',START_DT,END_DT);
YEARS_GAP=INTCK('YEAR',START_DT,END_DT);
FORMAT START_DT END_DT MMDDYY10.;
RUN;
```

INTNX FUNCTION TO ADD NUMBER OF DAYS, MONTHS, WEEKS, AND YEARS TO A DATE VALUE

EX:

```
DATA ADD_DATES;
CURRENT_DATE=DATE();
NEW_DATE1=CURRENT_DATE+5;
NEW_DATE2=INTNX('MONTH',CURRENT_DATE,5,'B');
NEW_DATE3=INTNX('MONTH',CURRENT_DATE,5,'S');
NEW_DATE4=INTNX('MONTH',CURRENT_DATE,5,'E');
NEW_DATE5=INTNX('YEAR',CURRENT_DATE,5,'S');
FORMAT CURRENT_DATE NEW_DATE1 NEW_DATE2 NEW_DATE3 NEW_DATE4 NEW_DATE5 MMDDYY10.;
RUN;
```

SIGNIFICANTS OF 'B' AS BIGING OF THE MONTH

SIGNIFICANTS OF 'E' AS END OF THE MONTH

SIGNIFICANTS OF 'S' AS SALES DATE OF THE MONTH

Q34. WHAT WILL BE THE OUTPUT OF BELOW SAS PROGRAM?

```
DATA NEW_DATE;
X='01DEC2020'D;
MONTHS_ADD=INTNX('MONTH',X,3,'S');
DAYS_ADD=INTNX('DAY',X,3);
YEARS_ADD=INTNX('YEAR',X,3,'B');
WEEKS_ADD=INTNX('WEEK',X,3);
FORMAT X MONTHS_ADD DAYS_ADD YEARS_ADD WEEKS_ADD MMDDYY10.;
RUN;
```

ANS

```
MONTHS_ADD=03/01/2021
DAYS_ADD=12/04/2020
YEARS_ADD=01/01/2023
WEEKS_ADD=12/20/2020
```

Q35. WHAT IS THE DIFFERENCE BETWEEN SCAN AND SUBSTR FUNCTION IN SAS?**ANS****SUBSTR** FUNCTION EXTRACTS A PORTION OF THE VALUE BY STATING THE SPECIFIC LOCATION

FIRST_NAME=SUBSTR(STU_NAME,1,INDEX(STU_NAME,' ')-1);

SCAN FUNCTION EXTRACTS WORDS WITHIN A VALUE THAT IS MARKED BY DELIMITERS

```
X='DEBENDRA:DAS>ANALYTICS-16/38:FPM:ODISHA';
FNAME=SCAN(X,1,',');;
```

Q36. WHAT IS THE DIFFERENCE INPUT AND PUT FUNCTION IN SAS?**ANS****INPUT** FUNCTION TO CONVERT CHARACTER INTO NUMERIC

NEW_YEAR=PUT(YEAR,\$10.);

PUT FUNCTION TO CONVERT NUMERIC INTO CHARACTER

NEW_SCORE=INPUT(SCORE,10.);

Q37. EXPLAIN FOLLOWING FUNCTIONS IN SAS?

1. **CATX** - `FULL_NAME2=CATX('_',F_NAME,L_NAME)`; CONCATENATE FIRST AND LAST NAME WITH UNDERSCORE
2. **INDEX**- `ANALYTICS_SEARCH1=INDEX(SEARCH,'ANALYTICS')`; WILL SEARCH ANALYTICS AS CHARACTER IN THE TEXT
3. **INDEXW**- `ANALYTICS_SEARCH2=INDEXW(SEARCH,'ANALYTICS')`; WILL SEARCH THE WORD ANALYTICS IN THE TEXT
4. **COMPRESS**- `ALL_NUM=COMPRESS(PHONE,'-')`; TO COMPRESS DATA WITHOUT KEEPING ANY SPACE IN BETWEEN
5. **COMPBL**- `NEW_NAME1=COMPBL(FULL_NAME)`; TO COMPRESS DATA BY KEEPING ONE SPACE IN BETWEEN

Q38. HOW MANY RECORDS IT WILL CREATE IF WE RUN SAS LOOP CODE BELOW?**QUERY-1**

```
DATA ROW_NUM;
DO I=1 TO 100;
END;
RUN;
```

QUERY-2

```
DATA ROW_NUM;
DO I=1 TO 100;
OUTPUT;
END;
RUN;
```

ANS

QUERY-1 ONE OBSERVATION
QUERY-2 HUNDRED OBSERVATIONS

Q39. WHAT IS THE DIFFERENCE BETWEEN DO UNTIL AND DO WHILE?**ANS**

DO UNTIL LOOPS CONTINUE EXECUTING UNTIL THE CONDITION YOU HAVE SPECIFIED BECOMES TRUE
DO WHILE LOOPS CONTINUE EXECUTING WHILE THE CONDITION YOU HAVE SPECIFIED REMAINS TRUE

```
DATA INVEST;
DO YEAR=1 TO 20 UNTIL (CAPITAL >=20000000);
CAPITAL+500000;
CAPITAL+(CAPITAL*.08);
OUTPUT;
END;
RUN;
```

```
DATA INVEST;
DO WHILE (CAPITAL <= 20000000);
YEAR+1;
CAPITAL+500000;
CAPITAL+(CAPITAL*.08);
OUTPUT;
END;
RUN;
```

Q40. HOW TO FILL ALL NUMERIC VARIABLE MISSING TO 0?**ANS**

	GROUP	VAR1	VAR2	VAR3
1	A	1	2	3
2	A	.	2	5
3	A	3	4	.
4	A	8	.	8
5	B	.	.	1
6	B	.	2	6
7	B	3	1	8
8	C	9	4	.
9	C	.	.	7



	GROUP	VAR1	VAR2	VAR3
1	A	1	2	3
2	A	0	2	5
3	A	3	4	0
4	A	8	0	8
5	B	0	0	1
6	B	0	2	6
7	B	3	1	8
8	C	9	4	0
9	C	0	0	7

SOLUTION-1:

```
DATA WORK.DS_NO_MISSING_VALUES;
SET WORK.DS_MISSING_VALUES;
VAR1 = COALESCE(VAR1,0);
VAR2 = COALESCE(VAR2,0);
VAR3 = COALESCE(VAR3,0);
RUN;
```

SOLUTION-2:

```
DATA WORK.DS_NO_MISSING_VALUES;
SET WORK.DS_MISSING_VALUES;
ARRAY NUM_ARRAY _NUMERIC_;
DO OVER NUM_ARRAY;
IF MISSING(NUM_ARRAY) THEN NUM_ARRAY = 0;
END;
RUN;
```

SOLUTION-3:

```
PROC STDIZE DATA=WORK.DS_MISSING_VALUES
OUT=WORK.DS_NO_MISSING_VALUES
REONLY MISSING=0;
RUN;
```

Q41. HOW TO FILL ALL CHARCATER AND NUMERIC VARIABLE MISSING?**ANS**

```
DATA STU_PERFORMANCE_V1;
SET STU_PERFORMANCE;
ARRAY CHAR _CHARACTER_;
DO OVER CHAR;
IF CHAR=' ' THEN CHAR='NA';
END;
ARRAY NUM _NUMERIC_ ;
DO OVER NUM;
IF NUM=. THEN NUM=0;
END;
RUN;
```

Q42. HOW TO ADD ALL THE NUMERIC VARIABLES WITH 100?**ANS**

```
DATA STU_SCORE2;
SET STU_SCORE;
ARRAY NUM (*) _NUMERIC_;
DO I=1 TO DIM(NUM);
NUM(I)=NUM(I)+100;
END;
DROP I;
RUN;
```

Q43. USE OF FIRST. AND LAST.?**ANS**

FIRST. AND LAST. FIND THE BY VARIABLES FIRST VALUE AND LAST VALUE
 FIRST.variable = 1, when an observation is the first observation in a BY group.
 FIRST.variable = 0, when an observation is not the first observation in a BY group.
 LAST.variable = 1, when an observation is the last observation in a BY group.
 LAST.variable = 0, when an observation is not the last observation in a BY group.

```
DATA PROD_SALES_V1;
SET PROD_SALES;
BY PROD;
IF FIRST.PROD THEN DO;
CNT=0;
CUM_SALES=0;
END;
CNT+1;
CUM_SALES+SALES;
IF LAST.PROD;
DROP CITY SALES;
RUN;
```

Q44. HOW TO GET 3RD HIGHEST SCORE BY STU_NAME?

	STU_NAME	SUBJECT	SCORE
1	ALAM	EXCEL	81
2	ALAM	SQL	42
3	ALAM	PYTHON	84
4	ALAM	SAS	19
5	ALAM	TABLEAU	75
6	ALAM	POWER BI	47
7	KIRAN	EXCEL	13
8	KIRAN	SQL	29
9	KIRAN	PYTHON	69
10	KIRAN	SAS	38
11	KIRAN	TABLEAU	72
12	KIRAN	POWER BI	73
13	JINUS	EXCEL	19
14	JINUS	SQL	13
15	JINUS	PYTHON	71
16	JINUS	SAS	45
17	JINUS	TABLEAU	95
18	JINUS	POWER BI	41

ANS**SOLUTION-1:**

```
PROC RANK DATA=STU_SCORE DESCENDING OUT=STU_SCORE_RANK ;
VAR SCORE ;
RANKS RANKING ;
BY STU_NAME ;
RUN;
```

```
PROC PRINT DATA=STU_SCORE_RANK;
WHERE RANKING=3;
RUN;
```

SOLUTION-2:

```
DATA STU_SCORE_RANKING;
SET STU_SCORE;
BY STU_NAME;
IF FIRST.STU_NAME THEN DO ;
RANK=0;
END;
RANK+1;
IF RANK=3;
RUN;
```

Q45. WHAT IS THE DIFFERENCE BETWEEN SUM AND ADDITION?**ANS**

1. SUM CREATES EXPRESSION VALUE EVEN IF VALUE IS MISSING AND CONSIDER MISSING VALUE AS 0
2. ADDITION CREATES EXPRESSION VALUE TO 0 IF VALUE IS MISSING AND CONSIDER MISSING VALUE AS CHARACTER

```
DATA STU_SCORE_V1;
SET STU_SCORE;
TOTAL_SCORE_1=SAS+R+PYTHON+SQL+EXCEL+TABLEAU+POWER_BI;
TOTAL_SCORE_2=SUM(SAS,R,PYTHON,SQL,EXCEL,TABLEAU,POWER_BI);
RUN;
```

	STU_NAME	SAS	R	PYTHON	SQL	EXCEL	TABLEAU	POWER_BI	TOTAL_SCORE_1	TOTAL_SCORE_2
1	BHAWNA	38	54	97	37	31	21	98	376	376
2	SUBHASHREE	94	21	93	73	69	53	71	474	474
3	KALYAN	49	46	.	86	34	51	81	.	347
4	RAJIV	74	92	16	42	.	.	12	.	236
5	RENUKA	55	.	.	79	44	77	33	.	288
6	MANJULA	48	98	45	87	82	43	18	421	421

Q46. HOW TO SELECT RANDOM OBSERVATIONS FROM A DATASET?**ANS**

```
PROC SURVEYSELECT DATA=MED_SELECT METHOD=SRS N=100 OUT=MED_RANDOM;
RUN;
```

Q47. HOW TO SELECT PRIME NUMBER OBSERVATIONS FROM A DATASET?**ANS**

```
DATA DS;
DO I=1 TO 1000;
OUTPUT;
END;
STOP;
RUN;

DATA WANT (DROP=J);
ARRAY P {1000} _TEMPORARY_ (0 2:1000);
DO I = 2 TO 1000;
DO J = I TO 1000/I;
P [I*J] = 0;
END;
END;
DO UNTIL (LR);
SET DS END=LR;
CALL MISSING (EVEN, ODD, PRIME);
IF MOD(I, 2) = 0 THEN EVEN = I;
ELSE ODD = I;
IF WHICHN(I, OF P[*]) THEN PRIME = I;
OUTPUT;
END;
RUN;
```

Q48. HOW TO SELECT EVEN OBSERVATIONS FROM A DATASET?**ANS**

```
DATA MED_SELECT_V1;
SET MED_SELECT;
IF MOD(_N_,2)=0;
RUN;
```

Q49. HOW TO GET DESCRIBING STATISTICS IN SAS PROGRAMMING?**ANS**

1. PROC MEANS
2. PROC SUMMARY
3. PROC UNIVARIATE

Q50. HOW TO GET MISSING VALUES IN A DATASET?**ANS**

```
PROC MEANS DATA=ALL N NMIS;
VAR JAN FEB MAR;
RUN;

PROC SUMMARY DATA=ALL N NMIS PRINT;
VAR JAN FEB MAR;
RUN;
```

Q51. CAN WE SUMMARISE DATA IN BASE SAS PROGRAMMING WITHOUT USING SAS SQL ?**ANS**

YES, BY USING PROC MEANS OR PROC SUMMARY

```
PROC SQL;
SELECT
LAPTOP,
COUNTRY,
PURCHASE_YEAR,
SUM(UNITS) AS TOTAL_UNITS,
MEAN(SALES) AS AVG_SALES,
SUM(SALES) AS TOTAL_SALES FORMAT=15.
FROM SSN6.LAPTOP_WW_SALES
GROUP BY 1,2,3
ORDER BY 1,2,3;
QUIT;
```

SAME WE CAN GET USING

```
PROC MEANS DATA=SSN6.LAPTOP_WW_SALES MAXDEC=0 NWAY;
CLASS LAPTOP COUNTRY PURCHASE_YEAR;
VAR UNITS SALES;
OUTPUT OUT=SUMMARY_REPORT8 (DROP=_TYPE_ _FREQ_) SUM(UNITS)=TOTAL_UNITS MEAN(SALES)=AVG_SALES
SUM(SALES)=TOTAL_SALES;
RUN;
```

Q52. WHAT IS THE DIFFERENCE BETWEEN PROC MEANS AND PROC SUMMARY?

ANS

PROC MEANS BY DEFAULT PRODUCES PRINTED OUTPUT IN THE LISTING WINDOW OR OTHER OPEN DESTINATION. WHEREAS PROC SUMMARY DOES NOT AND IT NEEDS PRINT AS AN OPTION

Q53. HOW TO CREATE REPORTS USING SAS PROCEDURES?

ANS

1. PROC REPORT
2. PROC TABULATE

Q54. HOW TO SELECT TOP 10 OBSERVATIONS FROM TABLE USING SAS SQL AND BASE SAS PROGRAMMING?

ANS

```
PROC PRINT DATA=MED_SELECT (OBS=10);
RUN;

PROC SQL (OUTOBS=10);
SELECT * FROM MED_SELECT;
QUIT;
```

Q55. HOW TO CREATE A NEW DATASET BY LIMITING NUMBER OF OBSERVATIONS?

ANS

```
DATA SSN7.MED_STORE_V1;
SET SSN7.MED_STORE (OBS=100);
RUN;
```

Q56. WHAT WILL HAPPEN WHILE APPENDING DATASETS WITH MULTIPLE SET STATEMENT?

ANS

EX:

```
DATA ABC;
SET A;
SET B;
SET C;
RUN;
```

IT WILL TAKE OBSERVATIONS AND VALUES FROM DATASET **C** TO DATASET **ABC**

Q57. WHAT WILL HAPPEN IF WE MERGE DATASETS WITHOUT COMMON VARIABLES?

ANS

EX:

```
DATA NEW_DATASET;
MERGE STU_ENQ1 STU_ENQ2;
RUN;
```

IT WILL TAKE OBSERVATIONS AND VALUES FROM DATASET **STU_ENQ2** TO DATASET **NEW_DATASET**

Q58. HOW TO CREATE ROW NUMBER IN PROC SQL?

ANS

USING MONOTONIC() FUNCTION

Q59. WHAT IS COALESCE FUNCTION?**ANS**

COALESCE FUNCTION REPLACE MISSING VALUES IN A COLUMN. IT CAN BE APPLICABLE TO BOTH CHARACTER AND NUMERIC VARIABLES

EX:

1. COALESCE(MAR,0) AS MAR
2. COALESCE(FIRST_NAME,'NA') AS FIRST_NAME

Q60. WHAT IS CALCULATED FIELD IN SAS SQL?**ANS**

WHEN AN EXPRESSION CREATES FROM ANOTHER EXPRESSION, WE USE CALCULATED KEYWORD IN SAS SQL PROGRAMMING

EX:

```
DATA PROD_SALES;
INPUT PROD $ JAN FEB MAR APR MAY JUN;
CARDS;
APPLE 800 900 400 300 200 800
DELL 800 400 600 800 300 800
HP 600 800 400 600 300 300
ACER 200 800 500 700 800 400
SONY 100 700 300 800 900 200
;
RUN;

PROC PRINT DATA=PROD_SALES;
RUN;

PROC SQL;
CREATE TABLE SUMMARY AS
SELECT *,
JAN+FEB+MAR AS Q1_SALES,
APR+MAY+JUN AS Q2_SALES,
(CALCULATED Q1_SALES+ CALCULATED Q2_SALES) AS HAL_YEARLY_SALES,
(CALCULATED Q1_SALES/ CALCULATED HAL_YEARLY_SALES) AS Q1_PERCENTAGE FORMAT=PERCENT5.,
(CALCULATED Q2_SALES/ CALCULATED HAL_YEARLY_SALES) AS Q2_PERCENTAGE FORMAT=PERCENT5.
FROM PROD_SALES;
QUIT;
```

Q61. HOW DO YOU RUN A STORE PROCEDURE USING SAS SQL?**ANS**

WE CONNECT DB AND RUN THE EXISTING STORE PROCEDURE TO PRODUCE OUTPUT INTO SAS DATASET

EX:

```
PROC SQL;
CONNECT TO ODBCOLD(REQUIRED="DSN=&SQLSERVER; DATABASE=_REPOSITORY;");
EXECUTE (SP_META_STATS
@DATABASE = &VINTAGE.,
@TABLE_NAME = &TABLE_NAME.) BY ODBCOLD;
DISCONNECT FROM ODBCOLD;
QUIT;
```

Q62. HOW MANY WAYS WE CAN DEFINE SAS MACROS?**ANS**

1. %LET
2. POSITIONAL PARAMETERS
3. KEYWORD PARAMETERS
4. %GLOBAL
5. %LOCAL
6. %DO LOOP
7. CALL SYMPUT
8. PROC SQL INTO CLAUSE

Q63. WHAT IS THE MACRO DEBUGGING OPTIONS?**ANS**

1. OPTIONS MPRINT
2. OPTIONS MLOGIC
3. OPTIONS SYMBOLGEN
4. %PUT

Q64. TELL ME 5 SCENARIOS WHERE YOU DEVELOPED SAS MACRO PROGRAMMING?**ANS**

1. TO PULL DATA FROM SERVER BY GIVEN AUTOMATED CURRENT DATE AND YEARMONTH USING CALL SYMPUT YYYYMM MACRO VARIABLE AND AFTER PULLING DATA SPECIFY NUMBERS OF MONTHS BACK TO READ DATA FOR AN ANALYSIS USING AUTOMATED MACRO VARIABLES
2. TO READ DATA FROM GIVEN FILE PATH AND WRITE MACRO TO READ AND APPEND DATASETS IRRESPECTIVE OF DIFFERENT FILE NAMES
3. TO READ DATA FROM EXCEL FILE FROM MULTIPLE TABS AND APPEND THEM INTO ONE DATASETS
4. TO CREATE MULTIPLE DATASETS FROM ONE DATASET USING %DO LOOP
5. RUN THE END TO REPORTING PROCESS BY DESIGNING MACRO VARIABLES BASED ON DYNAMICNESS IN THE PROGRAM

EX1: **%MACRO ALL_COMPANY;**

```
%DO I=1 %TO 9;
PROC SQL;
CREATE TABLE MED_COMP&I.
AS
SELECT * FROM MED_SELECT_V1
WHERE COMPANY_ID="COMP&I.";
QUIT;
%END;
%MEND;
```

EX2: **%MACRO MED_DS_ALL;**

```
DATA %DO I=1 %TO 9;
MED_COMP&I.
%END;;
SET MED_SELECT_V1;
%DO I=1 %TO 9;
IF COMPANY_ID="COMP&I." THEN OUTPUT MED_COMP&I.;
%END;
%MEND;
```

EX3: **%MACRO APPEND_ALL;**

```
DATA MED_ALL_DS;
SET %DO I=1 %TO 9;
MED_COMP&I.
%END;;
RUN;
%MEND;
```

EX4: **DATA _NULL_;**

```
CALL SYMPUT ("YYYYMM",PUT(DATE(),YYMMN6.));
CALL SYMPUT ("YYYYMM1",PUT(INTNX('MONTH',DATE(),-1,'S'),YYMMN6.));
RUN;
```

```
DATA PROD_SALES;
INPUT PROD $ YEARMO SALES;
DATA LINES;
APPLE 202103 900
APPLE 202104 300
APPLE 202105 900
DELL 202103 450
DELL 202104 380
DELL 202105 560
HP 202103 950
HP 202104 380
HP 202105 890
;
```

RUN;

```
DATA PROD_SALES_&YYYYMM.;
SET PROD_SALES;
WHERE YEARMO=&YYYYMM.;
RUN;
PROC PRINT DATA=PROD_SALES_&YYYYMM.; RUN;
```


Q65. WHAT IS THE DIFFERENCE BETWEEN %EVAL AND %SYSVALF?**ANS**

1. **%EVAL** USES FOR ARITHMATIC EXPRESSION WITH SAS MACRO PROGRAM WHERE EXPRESSION CREATES FROM AN INTEGER VALUES
2. **%SYSEVALF** USES FOR ARITHMATIC EXPRESSION WITH SAS MACRO PROGRAM WHERE EXPRESSION CREATES FROM FLOATING VALUES

EX:

```
%LET A=100;
%LET B=200;
%LET C=&A.+&B.;

%PUT &C.;

%LET A=100.50;
%LET B=200.50;
%LET C=%SYSEVALF(&A.+&B.);

%PUT &C.;
```

Q66. WHAT IS THE SYMGET AND SYMGETN IN SAS MACRO FUNCTION?**ANS**

1. **SYMGET** READS MACRO VARIABLE VALUE AS CHARACTER
2. **%SYSEVALF** READS MACRO VARIABLE VALUE AS AN INTEGER

EX:

```
DATA CALORIES;
SET EXERCISE;
INTENSITY_CALORIES_CHAR=SYMGET(INTENSITY);
INTENSITY_CALORIES_NUM=SYMGETN(INTENSITY);
RUN;
```

Q67. WHAT IS CALL SYMPUT AND CALL SYMPUTX?**ANS**

BOTH SYMPUT AND SYMPUTX CONVERT THE VALUE TO CHARACTER BEFORE ASSIGNING TO A MACRO VARIABLE. SYMPUT GIVES YOU A MESSAGE ON THE LOG ABOUT THE CONVERSION, WHILE SYMPUTX DOES NOT

INTERVIEW PRACTICALLY ASKED QUESTIONS AND SOLVING APPROACHES

Transforming You

Q1. GIVEN PROD_SALES TABLE FIND THE ANSWER TO QUESTIONS GIVEN BELOW?

1. GET SALES MONTH OVER MONTH
2. GET CUSTOMER LEVEL TOP 3 PRODUCTS SELLING

CUSTOMER_ID	PRODUCT_ID	SALES_DATE	SALES_AMOUNT
CUST4	PROD1	23-09-2020	234
CUST10	PROD7	20-01-2020	422
CUST5	PROD7	14-01-2020	759
CUST1	PROD1	18-03-2020	880
CUST4	PROD5	08-01-2020	386
CUST9	PROD5	23-02-2020	527
CUST8	PROD8	24-11-2020	915
CUST2	PROD6	05-12-2020	270
CUST1	PROD10	05-05-2020	578
CUST1	PROD9	06-10-2020	233
CUST10	PROD10	16-10-2020	872
CUST7	PROD6	16-04-2020	153
CUST2	PROD7	17-03-2020	628
CUST2	PROD7	14-07-2020	789
CUST8	PROD1	10-10-2020	664
CUST10	PROD6	21-12-2020	167
CUST2	PROD2	08-07-2020	888
CUST8	PROD4	26-07-2020	279
CUST3	PROD4	22-09-2020	780
CUST5	PROD10	16-01-2020	178
CUST4	PROD10	10-12-2020	141
CUST5	PROD9	20-03-2020	414
CUST4	PROD5	04-07-2020	419
CUST7	PROD6	08-10-2020	298
CUST6	PROD2	26-10-2020	439
CUST3	PROD2	14-04-2020	326
CUST3	PROD7	11-08-2020	806
CUST2	PROD1	24-08-2020	514
CUST8	PROD9	03-01-2020	412
CUST4	PROD4	01-05-2020	649

ANS-1

GET SALES MONTH OVER MONTH

/--STEP-1 CREATE MONTH FIELD FROM SALES DATE FIRST AND STORE TO A NEW TABLE*/*

```
DATA PROD_SALES;
SET PROD_SALES;
SALES_MONTH=MONTH(SALES_DATE);
RUN;
```

/--STEP-2 SUMMARIZE DATA BY MONTH WISE SALES*/*

```
PROC SQL;
SELECT
SALES_MONTH,
SUM(SALES_AMOUNT) AS TOTAL_SALES
FROM PROD_SALES_V1
GROUP BY SALES_MONTH;
QUIT;
```

ANS-2

GET CUSTOMER LEVEL TOP 3 PRODUCTS SELLING

/*--STEP-1 CREATE MONTH FIELD FROM SALES DATE FIRST AND STORE TO A NEW TABLE*/

```
DATA PROD_SALES;  
SET PROD_SALES;  
SALES_MONTH=MONTH(SALES_DATE);  
RUN;
```

/*--STEP-2 SUMMARIZE DATA BY MONTH WISE SALES*/

```
PROC SQL;  
CREATE TABLE PROD_SALES_V1  
AS  
SELECT  
CUSTOMER_ID,  
PRODUCT_ID,  
SUM(SALES_AMOUNT) AS TOTAL_SALES  
FROM PROD_SALES  
GROUP BY 1,2;  
QUIT;
```

/*--STEP-3 RANK SALES TO GET TOP 3 PRODUCT SALES BY CUSTOMER_ID*/

```
PROC SORT DATA=PROD_SALES_V1;  
BY CUSTOMER_ID DESCENDING TOTAL_SALES;  
RUN;
```

```
PROC RANK DATA=PROD_SALES_V1 DESCENDING OUT=PROD_SALES_V2;  
VAR TOTAL_SALES;  
RANKS RANKING;  
BY CUSTOMER_ID;  
RUN;
```

```
PROC PRINT DATA=PROD_SALES_V2;  
WHERE RANKING <= 3;  
RUN;
```

OR

```
PROC SORT DATA=PROD_SALES_V1;  
BY CUSTOMER_ID DESCENDING TOTAL_SALES;  
RUN;
```

```
DATA PROD_SALES_V2;  
SET PROD_SALES_V1;  
BY CUSTOMER_ID;  
IF FIRST.CUSTOMER_ID THEN DO;  
RANK=0;  
END;  
RANK+1;  
IF RANK <= 3;  
RUN;
```

Q2. GIVEN RESTAURANT_TRANSACTION TABLE FIND THE ANSWER TO QUESTIONS GIVEN BELOW?

1. GET RESTAURANT WISE MONTH ON MONTH SALES
2. GET EACH RESTAURANT WISE TOP 5 CUSTOMERS BY SPENT
3. GET EACH RESTAURANT WISE AVERAGE DAYS GAP BY CUSTOMER VISITS
4. HOW TO GET CUSTOMER INSIGHTS FROM THIS DATA

REST_NAME	CUST_NAME	VISIT_DATE	SPENT_AMOUNT
KFC	Rick Hansen	15-03-2019	2343
MACD	Justin Ritter	10-11-2019	565
GURU	Craig Reiter	01-08-2019	1123
PURVI	Katherine Murray	26-07-2019	2032
DALMA	Jim Mitchum	17-03-2019	4601
MACD	Toby Swindell	20-08-2019	6704
GREEN	Mick Brown	22-08-2019	360
PURVI	Jane Waco	01-02-2019	2304
MALWA	Joseph Holt	20-12-2019	4552
KITTO	Greg Maxwell	08-03-2019	3456
GYMNI	Anthony Jacobs	12-10-2019	3284
JUNGLE	Mick Brown	09-12-2019	6650
TARINI	Magdelene Morse	20-05-2019	5777
DALMA	Craig Reiter	23-03-2019	1109
ODIA	Vicky Freymann	13-08-2019	2624
DINE-IN	Craig Reiter	11-06-2019	6511
LOGGO	Greg Maxwell	26-12-2019	2170
REFORM	Jim Mitchum	24-01-2019	3100
GYMNI	Peter Fuller	14-02-2019	4741
MALWA	Ben Peterman	04-03-2019	3144
KFC	Thomas Boland	07-12-2019	6197
GURU	Rick Hansen	17-03-2019	4996
HOWDI	Magdelene Morse	20-02-2019	3302
LOGGO	Patrick Jones	18-09-2019	2648
DALMA	Jim Sink	09-11-2019	857
ODIA	Patrick Jones	08-02-2019	6721
MACD	Ritsa Hightower	13-08-2019	819
YO CHOW	Ann Blume	26-09-2019	5643
KFC	Rick Hansen	04-03-2019	3870
LEONE	Sue Ann Reed	09-01-2019	4817
MACD	Ann Blume	15-08-2019	2790
YO CHOW	Jason Klamczynski	16-12-2019	3387
MAYURI	Laurel Beltran	01-01-2019	2886

ANS-1

GET RESTAURANT WISE MONTH ON MONTH SALES

/*STEP-1 CREATE MONTH FIELD FIRST AND STORE TO A NEW TABLE*/

```
DATA RESTAURANT_TRANSACTIONS;  
SET RESTAURANT_TRANSACTIONS;  
VISIT_MONTH=MONTH(VISIT_DATE);  
RUN;
```

/*STEP-2 SUMMARIZE DATA BY REST_NAME AND MONTH WISE*/

```
PROC SQL;  
SELECT  
REST_NAME,  
VISIT_MONTH,  
SUM(SPENT_AMOUNT) AS TOTAL_SPENT  
FROM RESTAURANT_TRANSACTIONS  
GROUP BY 1,2;  
QUIT;
```

ANS-2

GET EACH RESTAURANT WISE TOP 5 CUSTOMERS BY SPENT

/*STEP-1 SUMMARIZE DATA BY REST_NAME AND CUST_NAME WISE TOTAL SPENT*/

```
PROC SQL;  
CREATE TABLE RESTAURANT_TRANSACTIONS_V1  
AS  
SELECT  
REST_NAME,  
CUST_NAME,  
SUM(SPENT_AMOUNT) AS TOTAL_SPENT  
FROM RESTAURANT_TRANSACTIONS  
GROUP BY 1,2;  
QUIT
```

/*STEP-1 GET TOP 5 CUSTOMERS BY RESTAURANT WISE*/

```
PROC SORT DATA=RESTAURANT_TRANSACTIONS_V1;  
BY REST_NAME DESCENDING TOTAL_SPENT;  
RUN;
```

```
PROC RANK DATA=RESTAURANT_TRANSACTIONS_V1 DESCENDING OUT=RESTAURANT_TRANSACTIONS_V2;  
VAR TOTAL_SPENT;  
RANKS RANKING;  
BY REST_NAME;  
RUN;
```

```
PROC PRINT DATA=RESTAURANT_TRANSACTIONS_V2;  
WHERE RANKING <= 5;  
RUN;
```

OR

```
PROC SORT DATA=RESTAURANT_TRANSACTIONS_V1;  
BY REST_NAME DESCENDING TOTAL_SPENT;  
RUN;
```

```
DATA RESTAURANT_TRANSACTIONS_V2;  
SET RESTAURANT_TRANSACTIONS_V1;  
BY REST_NAME;  
IF FIRST.REST_NAME THEN DO;  
RANK=0;  
END;  
RANK+1;  
IF RANK <= 5;  
RUN;
```

ANS-3**GET EACH RESTAURANT WISE AVERAGE DAYS GAP BY CUSTOMER VISITS**

/*STEP-1 USE LAG FUNCTION TO GET PREVIOUS_DAY VISIT AND USE DATEDIFF FUNCTION TO GET DAYS GAP BETWEEN PREVIOUS DAY VISIT TO NEXT DAY VISITS DAYS GAP*/

```
DATA RESTAURANT_TRANSACTIONS_V1;  
SET RESTAURANT_TRANSACTIONS;  
BY REST_NAME CUST_NAME;  
PREVIOUS_VISIT_DATE=LAG(VISIT_DATE);  
IF FIRST.CUST_NAME THEN PREVIOUS_VISIT_DATE=.;  
DAYS_GAP=INTCK("DAY",PREVIOUS_VISIT_DATE,VISIT_DATE);  
RUN;
```

/*STEP-2 SUMMARIZE DATA BY REST_NAME, CUST_NAME WISE AVERAGE DAYS_GAP IN VISIT*/

```
PROC SQL;  
SELECT  
REST_NAME,  
CUST_NAME,  
AVG(DAYS_GAP) AS AVG_VISIT_DAYS_GAP  
FROM RESTAURANT_TRANSACTIONS_NEW  
GROUP BY REST_NAME,CUST_NAME  
ORDER BY 1,2;  
QUIT;
```

ANS-4**HOW TO GET CUSTOMER INSIGHTS FROM THIS DATA****APPROACH-1**

GET RESTAURANT WISE CUSTOMERS TOTAL SPENT AND FIND HOW MANY CUSTOMERS SPENT ABOVE AVERAGE AND BELOW AVERAGE TO RESTAURANT AVERAGE SPENT

SPENT ABOVE AVERAGE CALL THEM PREMIER

SPENT BELOW AVERAGE CALL THEM NON-PREMIER

APPROACH-2

GET RESTAURANT WISE CUSTOMERS AVG_DAYS_GAP IN VISITS AND THEIR TOTAL_SPENT

GET RESTAURANT LEVEL AVG VISITS DAYS GAP AND AVG SPENT

SEGMENT CUSTOMER WHO ARE BELOW AVG DAYS GAP AND ABOVE AVG SPENT AT RESTAURANT LEVEL AS

PREMIER

SEGMENT CUSTOMER WHO ARE ABOVE AVG DAYS GAP AND BELOW AVG SPENT AT RESTAURANT LEVEL AS **NON-**

PREMIER

THIS HELPS IN CRACKING SAS INTERVIEW QUESTIONS.

MAKE USE OF IT