REVATURE – 18-09-2023 COHORT

L2-ASSIGNMENT – **JCL-VSAM-COBOL**

Prepared by : Alwyn Micaiah T

NOMINEE – 3A

Naming Convention:

PDS name: <**HLQ>.NOMINEE.PDS**

LOAD LIBRARY name: <**HLQ>.NOMINEE.LOADLIB**

HLQ → Mainframe Userid.(OZAGS1)

Member names:

JCL(VSAM) : JA01YYZZ

COMPILE JOB : COBCOMP

RUNJOB :JA03YYZZZ

COBOL MAIN PROGRAM MEM NAME: CA01YYZZ

COBOL SUB PROGRAM MEM NAME: CA02YYZZ

XX→ SEQUENCE NUMBER eg: 01,02,etc

YY→ QUESTION NUMBER . 3A as mentioned in line 4.

ZZ → The last 2 chars of your userid. Eg:"S1" if user id is OZAGS1.

STEP1: ISPF

- Allocate a PS dataset with the record length of 80 using the naming conventions given below PS1 - <HLQ>.NOMINEE.PS1
- Using the details from below file layout, enter records into the PS files as per the instructions given below,
  - 1st row contains the header details, which has to be entered in both the PS file PS1.
  - Do not enter the 2nd row in any of the PS files. 2nd row contains layout details for your reference.
  - One space filler had to be inserted between each field .
  - Insert 6 more records of your choice .

NOM_ID	**DOB**	**ANN_INC**	**NOM_REL**
X(5)	**X(10)**	**9(6).9(2)**	**X(4)**
N0001	02-03-1966	500000.23	Self

N0002	03-07-1989	600000.52	Self
	09-09-1992	569880.32	Hus
N0004	17-10-1984	597563.21	Hus
N0003	08-08-1991	400000.23	Self
N0006	04-05-1985	765632.00	Wife
N0002	03-07-1989	600000.52	Self
N0004	17-10-1984	597563.21	Hus

STEP2: JCL

- Write a IDCAMS job and allocate a VSAM KSDS datasets with RECORDSIZE (80, 80) and KEYS (5, 0), with naming convention
  - <HLQ>.NOMINEE.KSDS

STEP3: JCL

Write a job for the below

Step005: PRE- DELETE step for PS2 file

Step010: Using SORT utility perform the below operations:

Input : ps1
Output: ps2

- Remove the header record from file PS1 .
- Sort the records in ascending order based on Nom_Id
- Remove duplicate records and omit those records which have spaces in the Nom-Id
- Reformat the records to the given format
-

NOM_ID	DOB	NOM_REL	ANN_INC
X(5)	X(10)	X(4)	9(6).9(2)

Step 4:COBOL

Input file : <HLQ>.NOMINEE.PS2

DD Name: INDD1

Output file : <HLQ>.NOMINEE.KSDS, DD Name: OUTKSDS

Error file: <HLQ>.NOMINEE.ERR, DD Name: OUTERRPS

Eligible file : <HLQ>.NOMINEE.ECL, DD Name: OUTECLPS

Non-eligiblefile : <HLQ>.NOMINEE.NCL, DD Name: OUTNCLPS

Write a COBOL program to perform the following,

- Read records from input PS2dataset(<HLQ>.NOMINEE.PS2) and validate input values for each field in the input file.
  - Check whether Date, Month and Year of the DOB field is numeric.
  - Check whether NOM_REL is not blank.
  - Check whether ANN_INC is **numeric before and after decimal point**.
- If the input record does not pass through the validations specified above successfully, then write that input record into error PS file <HLQ>.NOMINEE.ERR
  Layout of ERROR file is as below.

NOM_ID	DOB	NOM_REL	ANN_INC
X(5)	X(10)	X(4)	9(6).9(2)

Note : One space filler to be inserted between each field.

- Do the following processing for **every record which has passed through the validations successfully.**

1. If Year(DOB) < = 1985
   Claim-Elig = N
   Else
   Claim-Elig = Y

2. Using Subprogram

- Calculate Disc-Amt and Amt-Paid

  Amt-Paid = Ann-Inc – (0.45 * Ann-Inc)

  Disc-Amt = Amt-Paid / 2

3. In the main program

- If Nom-Rel = self
  Increase Ann-inc by 40% of amt-paid
- *If Nom-Rel = Hus*
  Increase Ann-inc by 30% of amt-paid
- For others
  Increase Ann-inc by 30% of amt-paid

4.Write all the records into the output KSDS file which has the following format

NOM_ID	DOB	NOM_REL	ANN_INC	DISC-AMT	AMT-PAID	CLAIM-ELIG
X(5)	X(10)	X(4)	9(6).9(2)	9(6)	9(6).9(2)	X

5.Also move all the records into a Cobol 1 dimensional table(Array) having the following format

NOM_ID	DOB	NOM_REL	ANN_INC	CLAIM-ELIG
X(5)	X(10)	X(4)	9(6).9(2)	X

From the Cobol 1 dimensional table check the following

- If CLAIM_ELIG = Y
  - Write the records to the Eligible file. **<HLQ>.NOMINEE.ECL**
- If CLAIM_ELIG = N
  - Write the records to the Non-eligible file **<HLQ>.NOMINEE.NCL**

The Layout of both Eligible and Non-eligible files will be same as that of the Cobol 1 dimensional table

6.Compile and run the above COBOL program to achieve the results.