INFO3105 Week 9 Part 2

Recap of Case 2 to Date

1. JCS2LDRN
    a. Defines a KSDS called SLSPKSDS
    b. Executes DFSort to load SLSPKSDS
    c. Defines an AIX based on the branch code
    d. Executes PCS2PRG1 to process transactions against the KSDS
2. Separately you have a JCL stream to compile and link PCS2PRG1

PCS2PRG2

The next program for the second case study is another summary report program. This program builds on the one from the first case study, the difference being this one will do a **double control break vs the single/branch control break**:

```
PAGE   X                                  | ABC CORPORATION                          XX/XX/XXXX
                                          SALESPERSON BY BRANCH

 BRANCH: XXX
 DEPARTMENT   LAST NAME       FIRST NAME     GROSS SALES       RETURNS        NET SALES     COMMISSION   RATIN
    XX         Lauersen          Evan         $XX,XXX.XX      $X,XXX.XX      $XX,XXX.XX     $X,XXX.XX     ***
    XX         Flynn             Ashley       $XX,XXX.XX        $XXX.XX      $XX,XXX.XX     $X,XXX.XX    ****
    XX         Rowan             Eileen       $XX,XXX.XX        $XXX.XX      $XX,XXX.XX     $X,XXX.XX    ****
    XX         DeGaetano         Catherine    $XX,XXX.XX        $XXX.XX      $XX,XXX.XX     $X,XXX.XX      **
    XX         Orlando           Randolph     $XX,XXX.XX        $XXX.XX      $XX,XXX.XX     $X,XXX.XX    ****
    XX         Banasiak          Nancy        $XX,XXX.XX      $X,XXX.XX      $XX,XXX.XX     $X,XXX.XX     ***
    XX         Hau               Jayne        $XX,XXX.XX         $XX.XX      $XX,XXX.XX     $X,XXX.XX    ****

               TOTAL DEPARTMENT XX            $XXX,XXX.XX      $X,XXX.XX     $XXX,XXX.XX    $XX,XXX.XX

    XX         Appel             Anne         $XX,XXX.XX        $XXX.XX      $XX,XXX.XX     $X,XXX.XX    ****
    XX         Baker             Anna         $XX,XXX.XX        $XXX.XX      $XX,XXX.XX     $X,XXX.XX    ****
    XX         Patchik           Joseph       $XX,XXX.XX        $XXX.XX      $XX,XXX.XX     $X,XXX.XX      **
    XX         Steele            Karen        $XX,XXX.XX        $XXX.XX      $XX,XXX.XX     $X,XXX.XX   *****

               TOTAL DEPARTMENT XX            $XXX,XXX.XX        $XXX.XX     $XXX,XXX.XX    $XX,XXX.XX

               TOTAL BRANCH XXX               $XXX,XXX.XX      $X,XXX.XX     $XXX,XXX.XX    $XX,XXX.XX
                                             -------------     ----------    -------------   -----------
  PAGE   X                                      ABC CORPORATION                          XX/XX/XXXX
                                          SALESPERSON BY BRANCH

 BRANCH: XXX
 DEPARTMENT   LAST NAME       FIRST NAME     GROSS SALES       RETURNS        NET SALES     COMMISSION   RATIN
    XX         King              Bryan        $XX,XXX.XX      $X,XXX.XX      $XX,XXX.XX        $XXX.XX      **
    XX         Lee               Christine    $XX,XXX.XX        $XXX.XX      $XX,XXX.XX     $X,XXX.XX    ****
    XX         Lee               Genoa        $XX,XXX.XX        $XXX.XX      $XX,XXX.XX     $X,XXX.XX    ****
```

```
 PAGE   X                                  ABC CORPORATION                        XX/XX/XXXX
                                        SALESPERSON BY BRANCH
BRANCH: XXX
DEPARTMENT   LAST NAME       FIRST NAME     GROSS SALES       RETURNS        NET SALES     COMMISSION   RATING
   XX         Dube             Jason        $XX,XXX.XX        $XXX.XX       $XX,XXX.XX       $XXX.XX      **
   XX         Holtz            Rachel       $XX,XXX.XX        $XXX.XX       $XX,XXX.XX     $X,XXX.XX     ***
   XX         Theis            Sean         $XX,XXX.XX        $XXX.XX       $XX,XXX.XX     $X,XXX.XX      **

              TOTAL DEPARTMENT XX           $XXX,XXX.XX      $X,XXX.XX      $XXX,XXX.XX     $X,XXX.XX

   XX         Roe              Eileen       $XX,XXX.XX        $XXX.XX       $XX,XXX.XX     $X,XXX.XX    ****
   XX         Crean            Kathleen     $XX,XXX.XX        $XXX.XX       $XX,XXX.XX     $X,XXX.XX   *****
   XX         Hazard           Mary         $XX,XXX.XX      $X,XXX.XX       $XX,XXX.XX     $X,XXX.XX    ****

              TOTAL DEPARTMENT XX           $XXX,XXX.XX      $X,XXX.XX      $XXX,XXX.XX    $XX,XXX.XX

   XX         Mohammadi        Mary         $XX,XXX.XX        $XXX.XX       $XX,XXX.XX     $X,XXX.XX    ****
   XX         Reed             Margaret     $XX,XXX.XX        $XXX.XX       $XX,XXX.XX     $X,XXX.XX   *****
   XX         Gibbs            Janelle      $XX,XXX.XX        $XXX.XX       $XX,XXX.XX     $X,XXX.XX   *****
   XX         Stockover        Nancy        $XX,XXX.XX        $XXX.XX       $XX,XXX.XX     $X,XXX.XX    ****
   XX         Ford             Dalia        $XX,XXX.XX      $X,XXX.XX       $XX,XXX.XX     $X,XXX.XX   *****
   XX         Doherty          Elizabeth    $XX,XXX.XX        $XXX.XX       $XX,XXX.XX     $X,XXX.XX   *****

              TOTAL DEPARTMENT XX           $XXX,XXX.XX      $X,XXX.XX      $XXX,XXX.XX    $XX,XXX.XX

              TOTAL BRANCH XXX               $XXX,XXX.XX      $X,XXX.XX      $XXX,XXX.XX    $XX,XXX.XX
                                            -------------    ----------    -------------   -----------
              COMPANY TOTALS                $X,XXX,XXX.XX   $XXX,XXX.XX   $X,XXX,XXX.XX   $XXX,XXX.XX
                                            =============    ==========    =============   ===========
```

... so you know what you are shooting for ... here are the last few lines:

```
30         Doherty          Elizabeth    $81,593.90        $211.72       $81,382.18     $2,465.88

           TOTAL DEPARTMENT 30           $459,512.76      $2,296.68      $457,216.08    $14,255.02

           TOTAL BRANCH 500               $804,061.01      $5,190.57      $798,870.44    $26,649.82
                                         -------------    ----------    -------------   -----------
           COMPANY TOTALS                $5,976,449.07    $95,824.13    $5,880,624.94   $191,495.89
                                         =============    ==========    =============   ===========
```

So we see that we've added one new column (Department #) and one new row (Department Total) to the report. Again, key to this is to be able to process this file in a sorted order. When we defined the KSDS we made the AIX a combination of the BranchNo and DeptNo so the entire key was 5 bytes.

So for this code to work correctly it needs to use the whole 5 byte key ... so ensure your SALESPERSON-MAST file layout copybook (CCS2SLSP) has a 5 byte BRANCH-NO:

```
01 SALESPERSON-MASTER.

...

    05 SALESPERSON-RETURN-SALES   PIC 9(4)V99.

    05 SALESPERSON-BRANCH-NO      PIC X(5).

    05 SALESPERSON-COMM-RATE      PIC V9999.
```

... Then after you read in the SALESPERSON-BRANCH-NO key you can move it to a field like this:

```
05  WS-BRANCH-KEY.
    10 WS-BRANCH-NO        PIC 9(3).
    10 WS-DEPT-NO          PIC 9(2).
```

... so that you can use the WS-BRANCH-NO to check for Branch control breaks and the WS-DEPT-NO to check for Department control breaks...

In PCS2PRG1 we processed the KSDS randomly. KSDS files can also be processed **sequentially** - the difference is to use the **AIX** (Alternate KEY) not the primary cluster, like this:

First we need to define the ACCESS IS SEQUENTIAL in the select:

```
SELECT SALESMAST ASSIGN TO SLSPKS
  ORGANIZATION IS INDEXED
  ACCESS IS SEQUENTIAL
  RECORD KEY IS SALESPERSON-NO
  ALTERNATE KEY IS SALESPERSON-BRANCH-NO WITH DUPLICATES
  FILE STATUS IS WS-IN-STATUS.
```

Then in the Procedure Division we need to **START** the AIX. The start verb is used when accessing a file by alternate index and establishes the key you want to use for retrieval, a complete description and example can be found on pages **432-433** of the text.

```
B000-START-AIX.
    MOVE LOW-VALUE TO SALESPERSON-BRANCH-NO.
    START SALESMAST KEY IS >= SALESPERSON-BRANCH-NO
          INVALID KEY MOVE 'Y' TO WS-SALESMAST-EOF-SWITCH.
```

This START AIX routine should be called directly after opening the files & before the loop processing of the sequential records:

```
PROCEDURE DIVISION.

A000-MAINLINE-PREPARE-SALES-REPORT.
    OPEN INPUT  SALESMAST
         OUTPUT SALESRPT.

    PERFORM B000-START-AIX.

    IF WS-IN-STATUS NOT EQUAL "00"
        DISPLAY "FILE ERROR IN-STATUS = ", WS-IN-STATUS
    ELSE
        PERFORM C200-PROCESS-SALESPEOPLE
           UNTIL WS-SALESMAST-EOF
```

Then it is just a matter of processing the control break logic like we did in case study #1 except we have to do 2 control breaks, one on the department # and the original on the branch #.

Again **you will need a new jcl procedure to compile and link the 2nd program (ie. PCS2PRG2 - remember to change the member names for the .COBOL and .LOAD in the JCL)**. Then you will need to update the JCS2LDRN to execute the new program after the transactions have been processed, something like this:

```
        CHARACTER
        COUNT(5)
/*
//          ENDIF
//*************************************************
//* EXECUTE COBOL PROGRAM MODULE PCS2PRG1         *
//* PROCESS TRANSACTIONS AGAINST KSDS             *
//*************************************************
//          IF RC < 8 THEN
//STEP4     EXEC PGM=PCS2PRG1
//STEPLIB   DD DSN=&SYSUID..LOAD,DISP=SHR
//SLSPKS    DD DSN=&SYSUID..SLSPKSDS,DISP=SHR
//SLSPKS1   DD DSN=&SYSUID..SLSP.BRN.PATH,DISP=SHR
//SLSTRANS  DD DSN=&SYSUID..SLSTRANS,DISP=SHR
//PRNT      DD SYSOUT=*
//          ENDIF
//*************************************************
//* EXECUTE COBOL PROGRAM MODULE PCS2PRG2         *
//* USING AIX FOR SEQUENTIAL ACCESS               *
//* OF KSDS FILE FOR DOUBLE CONTROL BREAK         *
//* THIS PROGRAM CALLS PCS2PRG3                   *
//*************************************************
//          IF RC < 8 THEN
//STEP5     EXEC PGM=PCS2PRG2
//STEPLIB   DD DSN=&SYSUID..LOAD,DISP=SHR
//PRNT      DD SYSOUT=*
//SLSPKS    DD DSN=&SYSUID..SLSPKSDS,DISP=SHR
//SLSPKS1   DD DSN=&SYSUID..SLSP.BRN.PATH,DISP=SHR
//          ENDIF
```

Two-Level Summary Report (Double Control break)

The text book discusses this type of report on pages **180-185**, notice how the author re-writes the logic using an Evaluate statement to give it a cleaner look, and uses the fact that an Evaluate statement checks the conditions in the order they are coded. For this case study I'd like you to use an EVALAUTE similar to the way it is done in the text to handle this double control break scenario (Note although similar in some ways, **the following is code from the textbook** and not the code for the case study) Note as well you will need to deal with the difference in definition between the 5 byte AIX definition and the 3 byte branch and 2 byte department, so you can control break on branch as your main control break, then department as your secondary control break:

```
300-PREPARE-SALES-LINES.
        PERFORM 310-READ-CUSTOMER-RECORD.
        EVALUATE TRUE
                WHEN CUSTMAST-EOF
                        PERFORM 355-PRINT-SALESREP-LINE
                        PERFORM 360-PRINT-BRANCH-LINE
                WHEN FIRST-RECORD
                        PERFORM 320-PRINT-CUSTOMER-LINE
                        MOVE 'N' TO FIRST-RECORD-SWITCH
                        MOVE CM-SALESREP-NUMBER TO OLD-SALESREP-NUMBER
                        MOVE CM-BRANCH-NUMBER TO OLD-CM-BRANCH-NUMBER
                WHEN CM-BRANCH-NUMBER > OLD-BRANCH-NUMBER
                        PERFORM 365-PRINT-SALESREP-LINE
                        PERFORM 360-PRINT-BRANCH-LINE
                        PERFORM 320-PRINT-CUSTOMER-LINE
                        MOVE CM-SALESREP-NUMBER TO OLD-SALESREP-NUMBER
                        MOVE CM-BRANCH-NUMBER TO OLD-CM-BRANCH-NUMBER
                WHEN CM-SALESREP-NUMBER > OLD-SALESREP-NUMBER
                        PERFORM 365-PRINT-SALESREP-LINE
                        PERFORM 320-PRINT-CUSTOMER-LINE
                        MOVE CM-SALESREP-NUMBER TO OLD-SALESREP-NUMBER
                WHEN OTHER
                        PERFORM 320-PRINT-CUSTOMER-LINE
        END-EVALUATE
```

Main Control Break

Secondary Control Break (**Yours** will be Dept-NO ")

Remember that your exact control break logic may be slightly different than the original control break logic from the notes/textbook (for example it is possible to code these report processing/control break programs with READ at the top or the bottom of the EVALUATE - which will change your logic for the 1st and last records of the file.)

Lab 13 - 4%

Due to the amount of work for this lab, there will be some extra time in class to work on this (& extra help – listen carefully in class for this extra help…) . For this lab **submit**:

- The entire JCS2LDRN execution listing, this should now include all of the AMS commands, the transaction processing log program (PCS2PRG1) and now the new double control break listing program (PCS2PRG2).
- The compile and Link output that includes the compile listing for both the PCS2PRG1 and the new PCS2PRG2 double control break program.