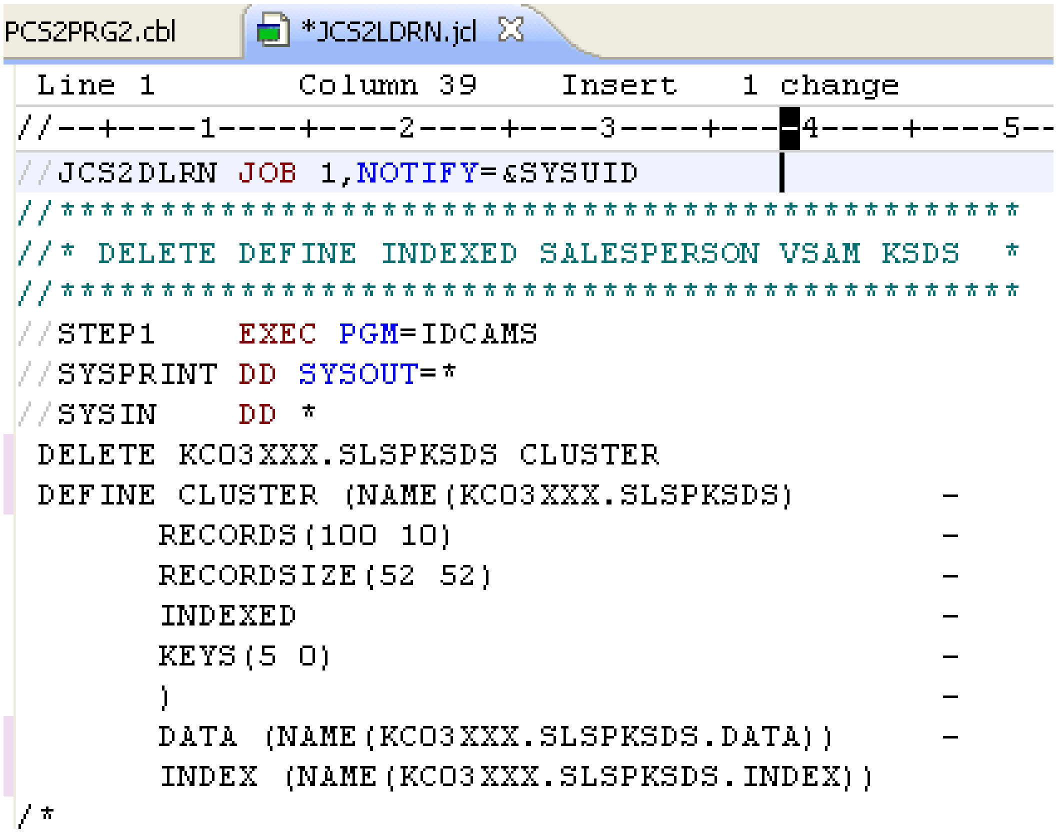
INFO3105 Week 7 Class 2 (Compressed Week 4)

Using VSAM in Case Study #2

Now that we know how to define a KSDS using the ISPF VSAM utility we're going to repeat the process, but this time do it as a **batch job**. The reason to set it up in a job is to allow for repetitive deleting and defining of this file multiple times. It will save work in the long run.

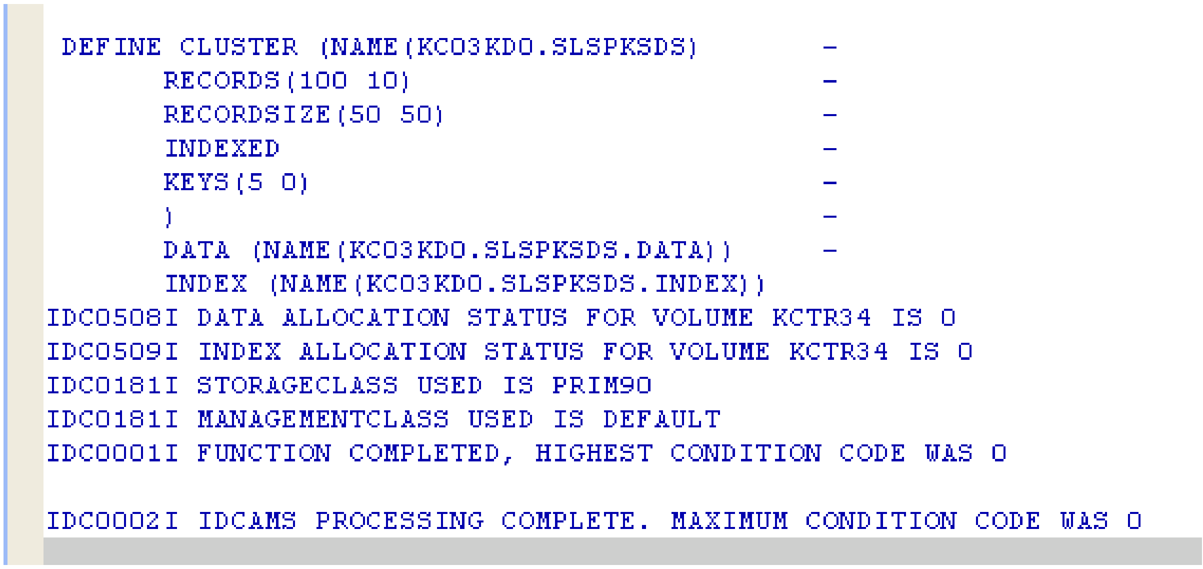
So add a new member to your JCL PDS called **JCS2LDRN** (Job Case2 LoaD RuN report) this eventually will end up as part of a 5 step job stream, today we'll be coding the first 3 steps.

So the starting JCL looks like this, note the KEYS (5 0) line means the salesperson # primary key is length 5, offset 0 columns from the start of the record (Also change xxx to your KC03... ID) :



```
//JCS2DLRN JOB 1,NOTIFY=&SYSUID
//************************************************
//* DELETE DEFINE INDEXED SALESPERSON VSAM KSDS  *
//************************************************
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DELETE KC03XXX.SLSPKSDS CLUSTER
 DEFINE CLUSTER (NAME(KC03XXX.SLSPKSDS)       -
       RECORDS(100 10)                        -
       RECORDSIZE(52 52)                      -
       INDEXED                                -
       KEYS(5 0)                              -
       )                                      -
       DATA (NAME(KC03XXX.SLSPKSDS.DATA))     -
       INDEX (NAME(KC03XXX.SLSPKSDS.INDEX))
/*
```

Note 1 reason you may get a non zero return code is if the Hyphen/Dash – (IDCAMS Sysin continuation character) is missing. When looking at the execution listing you should see something like this (make sure the maximum condition code is 0):

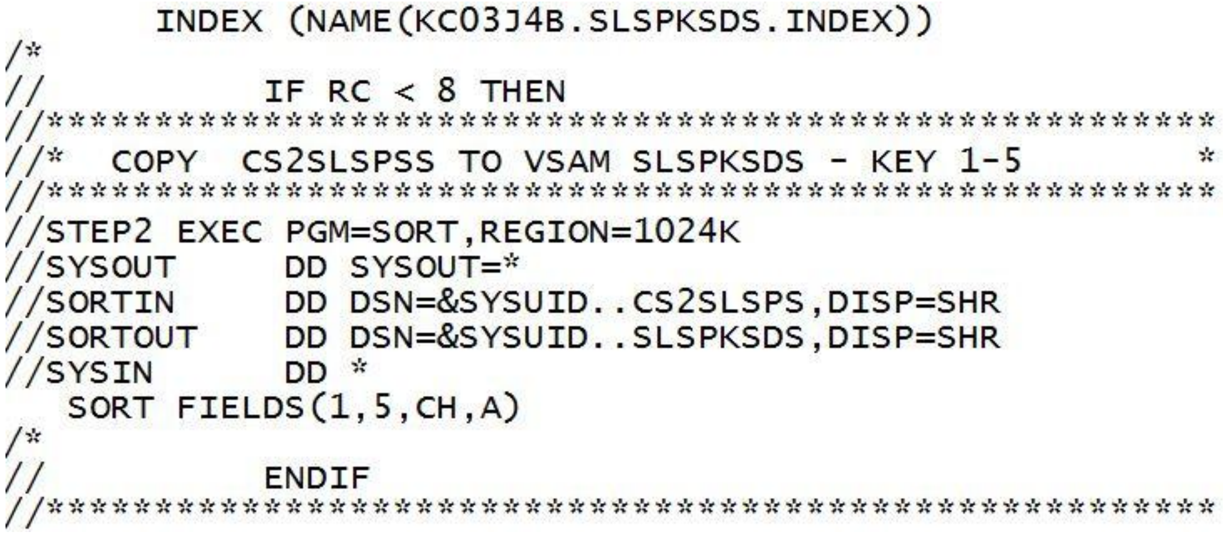


```
    DEFINE CLUSTER (NAME(KC03KD0.SLSPKSDS)        -
           RECORDS(100 10)                        -
           RECORDSIZE(50 50)                      -
           INDEXED                                -
           KEYS(5 0)                              -
           )                                      -
           DATA (NAME(KC03KD0.SLSPKSDS.DATA))     -
           INDEX (NAME(KC03KD0.SLSPKSDS.INDEX))
IDC0508I DATA ALLOCATION STATUS FOR VOLUME KCTR34 IS 0
IDC0509I INDEX ALLOCATION STATUS FOR VOLUME KCTR34 IS 0
IDC0181I STORAGECLASS USED IS PRIM90
IDC0181I MANAGEMENTCLASS USED IS DEFAULT
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

Loading the KSDS Master File

One of the issues that you'll now run into is that IBM Developer/IDZ **can't edit** this new KSDS, which means you can't do a copy/paste to get data into it. We need to come up with another strategy to load the data. Right now your data lies in a sequential file (you can use either of the master files from last class – CS2SLSPS or CS2SLSPU) and we need to get it translated over to the indexed format. Fortunately we can use the **DFSort** utility that we just learned last class to do this for us. So return to your JCS2LDRN member and place some sort jcl after the KSDS creation part and label it as **STEP2** :

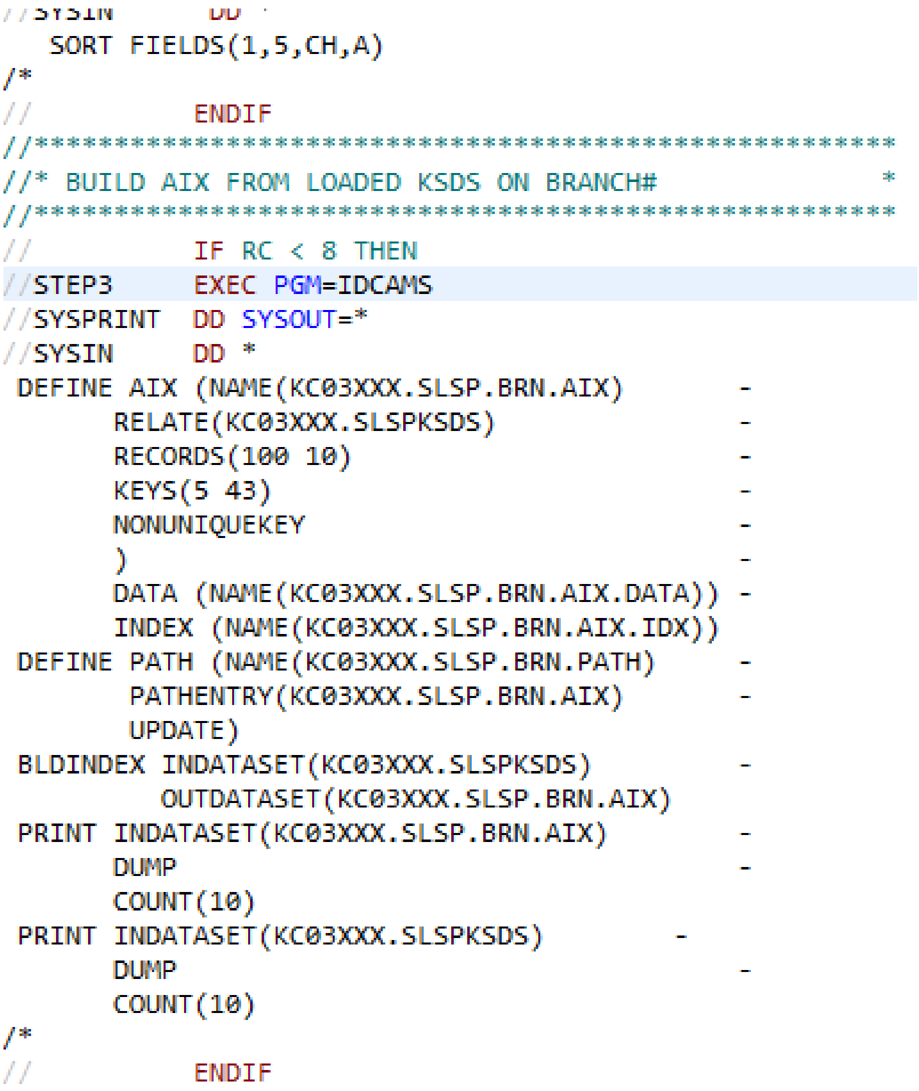


```
       INDEX (NAME(KC03J4B.SLSPKSDS.INDEX))
/*
//          IF RC < 8 THEN
//**********************************************************
//*  COPY  CS2SLSPSS TO VSAM SLSPKSDS - KEY 1-5          *
//**********************************************************
//STEP2 EXEC PGM=SORT,REGION=1024K
//SYSOUT     DD SYSOUT=*
//SORTIN     DD DSN=&SYSUID..CS2SLSPS,DISP=SHR
//SORTOUT    DD DSN=&SYSUID..SLSPKSDS,DISP=SHR
//SYSIN      DD *
   SORT FIELDS(1,5,CH,A)
/*
//          ENDIF
//**********************************************************
```

The sort fields indicate **the primary key** which is the **Salesperson#** in the data. Note the IF RC < 8 THEN ... and ENDIF commands in the JCL (which will control JCL execution & prevent STEPs from executing unless a previous STEP# is successful).

Building an Alternate Index

Similar to database tables KSDS's can have alternate keys. In VSAM we build what is called an alternate index. Right now we are using the **Salesperson number** as the **primary key**. We'll have a new control break program that we'll eventually process against this file but remember that kind of program uses something like the branch no as the control break field. To accommodate this we are going to add an alternate index to this file. The textbook talks about alternate indexes on pages **554-555** - as you read this notice how similar they are to what a foreign key does in the **database world**. The Access Method Services code to do so can be found on pages **580-586**. There are actually 4 parts that get created when building an alternate index. The code below should be added to the **JCS2LDRN** member after the sort step, and will set up these 4 parts and then print out the first 10 entries in the AIX. Again, you will need to **change the XXX** to your credentials. **Note** again the **dashes/hyphens are required** continuation characters in this IDCAMS build.



```
//SYSIN      DD *
   SORT FIELDS(1,5,CH,A)
/*
//          ENDIF
//******************************************************
//* BUILD AIX FROM LOADED KSDS ON BRANCH#              *
//******************************************************
//          IF RC < 8 THEN
//STEP3     EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
 DEFINE AIX (NAME(KC03XXX.SLSP.BRN.AIX)       -
       RELATE(KC03XXX.SLSPKSDS)               -
       RECORDS(100 10)                        -
       KEYS(5 43)                             -
       NONUNIQUEKEY                           -
       )                                      -
       DATA (NAME(KC03XXX.SLSP.BRN.AIX.DATA)) -
       INDEX (NAME(KC03XXX.SLSP.BRN.AIX.IDX))
 DEFINE PATH (NAME(KC03XXX.SLSP.BRN.PATH)     -
        PATHENTRY(KC03XXX.SLSP.BRN.AIX)       -
        UPDATE)
 BLDINDEX INDATASET(KC03XXX.SLSPKSDS)         -
          OUTDATASET(KC03XXX.SLSP.BRN.AIX)
 PRINT INDATASET(KC03XXX.SLSP.BRN.AIX)        -
       DUMP                                   -
       COUNT(10)
 PRINT INDATASET(KC03XXX.SLSPKSDS)        -
       DUMP                                   -
       COUNT(10)
/*
//          ENDIF
```

Note the **KEYS (5 43)** line ... this is for the **key of the alternate index** ... which starts in column 44 (ie. offset 43) for a length of 5 bytes. Check out the output from the PRINT in the spool, it is showing the alternate index data and the salesperson master file data that is stored in the KSDS files. Notice that the salesperson names are only displaying the first letter (since the print program assumes lower case characters are not printable).

Note you may run into CC0016 failures in the DFSORT or in the IDCAMS processing if you include extra spaces, missing Hyphens, etc … - Yes JCL is fairly fussy & error messages are not always intuitive … so best to start early / plan for getting some help in class or extra help sessions…

Another common error now that we are creating more complicated JCL scripts is if you are getting unusual JCL errors – specifically a MISPLACED DD STATEMENT, this is caused by inserting/including a blank line in the JCL

The 4 entities that get created are:

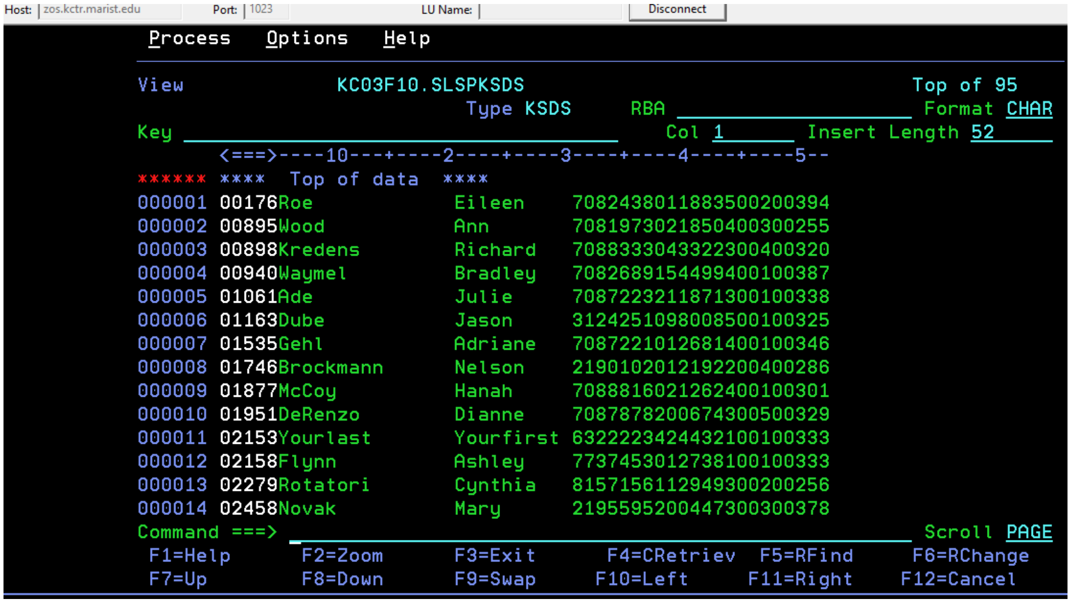
1. AIX
2. Data component
3. Index component
4. Path

Lab 9 - 2%

- Update the JCS2LDRN JCL member to now have:
    - Step 1 – define KSDS
    - Step 2 – execute sort to load the KSDS from one of the 52 byte files from last class
    - Step 3 – build your AIX

Submit:

1) The JES output spool of the entire execution of the JCS2LDRN submission
2) Also submit a screen capture of the newly created **SLSPKSDS** file as shown in the ISPF FM view editor ... similar to this :



```
 Process   Options   Help
View             KC03F10.SLSPKSDS                                 Top of 95
                            Type KSDS     RBA ____________________ Format CHAR
Key _____________________________________    Col 1______ Insert Length 52_____
       <===>----10---+----2----+----3----+----4----+----5--
****** ****  Top of data  ****
000001 00176Roe            Eileen    7082438011883500200394
000002 00895Wood           Ann       7081973021850400300255
000003 00898Kredens        Richard   7088333043322300400320
000004 00940Waymel         Bradley   7082689154499400100387
000005 01061Ade            Julie     7087223211871300100338
000006 01163Dube           Jason     3124251098008500100325
000007 01535Gehl           Adriane   7087221012681400100346
000008 01746Brockmann      Nelson    2190102012192200400286
000009 01877McCoy          Hanah     7088816021262400100301
000010 01951DeRenzo        Dianne    7087878200674300500329
000011 02153Yourlast       Yourfirst 6322223424432100100333
000012 02158Flynn          Ashley    7737453012738100100333
000013 02279Rotatori       Cynthia   8157156112949300200256
000014 02458Novak          Mary      2195595200447300300378
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```