ICETOOL – An Advance sorting functionality with Examples

31/12/2010

<Manufacturing Domain>/ < Cummins project>

<SINDUJA S>

sinduja2.s@tcs.com

ICETOOL - An Advance Sorting Utility

ICETOOL is a versatile DFSORT utility that allows you to perform multiple operations on one or more data sets in a single job step.

The ICETOOL has various operators, each of which can be used one or more times in a single run, allow you to perform a variety of functions.

It is used in displaying statistical information, creating output data sets with information spliced together from two or more input records with duplicate values, sorting records between headers and trailers and many more.

Sample Job Format

An ICETOOL job consists of :

- JCL Statements which are required for creating any job.
- The OPERATOR Statements(ICETOOL Statements) indicating the operations performed.
- The Additional JCL statement as a result of specified operator statements.
- The comment statement can be given which asterisk(*) in the column 1.
- Job Format: The general job format is given below,

```
//EXAMP JOB ...
//TOOL EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=A
//DFSMSG DD SYSOUT=A
//TOOLIN DD *
<ICETOOL statements go here>
/*
<Additional JCL statements go here>
```

The general format for all ICETOOL statements is:

Operator Operand-1<Parameters> ... Operand-n<Parameters>

JOB - Signifies the beginning of the job.

EXEC - Signifies the beginning of the job step and executes the ICETOOL program with the region of 1024K recommended.

TOOLMSG - Defines the output dataset of the ICETOOL messages.

DFSMSG - Defines the output dataset of the DFSORT messages.

TOOLIN- The area where the ICETOOL statements are to be written . The ICETOOL statements in TOOLIN ends with a comment in the end..

Operators

There are around 15 Operators as below,

COPY	*COUNT*	*DATASORT*	*DEFAULTS*	
DISPLAY	*MODE*			
OCCUR	*RANGE*	*SELECT*	*SORT*	*SPLICE*
STATS				
SUBSET	*UNIQUE*	*VERIFY*		

Functions Performed

The major operators among them are in the section below.

- **COPY -** Copies a data set to one or more output data sets. Multiple output is handled using a single pass over the input.

  Syntax:

  COPY FROM(indd) TO(outdd,...) USING(xxxx) VSAMTYPE(x)

  Example: JCL:

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.INPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPUT,
//   DISP=(NEW,CATLG,DELETE),
//   UNIT=WORK,VOL=(,,,48),
//   SPACE=(TRK,(1500,500),RLSE),
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//TOOLIN    DD   *
  COPY FROM(IN) TO(OUT) USING(CTL1)
//CTL1CNTL DD   *
  OMIT COND=(28,5,CH,NE,C'NUM')
/*
```

  Input:

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
--------------------------------------------------------------------------------
VIEW       CEC.IW588.COPY.INPUT                          Columns 00001 00072
Command ===>                                               Scroll ===> CSR
****** ******************************** Top of Data ********************************
000100 NIL                          NUM
000200 BABU                         NU1
000300 CHRISTOPHER                  NU2
000400 CHARLES                      NUM
000500 DANIEL                       NUM2
****** ***************************** Bottom of Data ******************************
```

Output:

```
File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
VIEW       CEC.IW588.COPY.OUTPUT                        Columns 00001 00072
Command ===>                                              Scroll ===> CSR
****** ***************************** Top of Data ******************************
000100 ANIL                          NUM
000400 CHARLES                       NUM
****** **************************** Bottom of Data ****************************
```

- **COUNT -** Prints a message in TOOLMSG containing the count of records in a data set. Can also be used to create an output data set containing text and the count, or to set RC=12 or RC=0 based on the count of records in a data set.

  Syntax:

  COUNT FROM(indd) USING(xxxx)

  Count operator can also be used for many other operations and the syntax is given below:

  EMPTY ,NOTEMPTY,HIGHER(x),LOWER(y),EQUAL(v),NOTEQUAL(w)

  Examples:

  - Simple Count Operator - To count the number of records in the input dataset.

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.INPUT,DISP=SHR
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//TOOLIN    DD   *
  COUNT FROM(IN)
/*
****************************** Bottom of Data
```

Input:

```
File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
VIEW       CEC.IW588.COPY.INPUT                         Columns 00001 00072
Command ===>                                              Scroll ===> CSR
****** ***************************** Top of Data ******************************
000100 ANIL                          NUM
000200 BABU                          NU1
000300 CHRISTOPHER                   NU2
000400 CHARLES                       NUM
000500 DANIEL                        NUM2
****** **************************** Bottom of Data ****************************
```

TOOLMSG:

```
 SDSF OUTPUT DISPLAY IW588SRT JOB01029  DSID   102 LINE 0       COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> CSR
******************************** TOP OF DATA ********************************
SYT000I  SYNCTOOL RELEASE 1.5.3 - COPYRIGHT 2004  SYNCSORT INC.
SYT001I  INITIAL PROCESSING MODE IS "STOP"
SYT002I  "TOOLIN" INTERFACE BEING USED

           COUNT FROM(IN)
SYT020I  SYNCSORT CALLED WITH IDENTIFIER "0001"
SYT031I  NUMBER OF RECORDS PROCESSED: 00000000000005
SYT030I  OPERATION COMPLETED WITH RETURN CODE 0

SYT004I  SYNCTOOL PROCESSING COMPLETED WITH RETURN CODE 0
******************************** BOTTOM OF DATA ********************************
```

- Count Operator with EMPTY - To check whether the dataset is empty.

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.INPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPU2,DISP=SHR
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//TOOLIN    DD   *
  COUNT FROM(IN) EMPTY
/*
```

Input dataset:

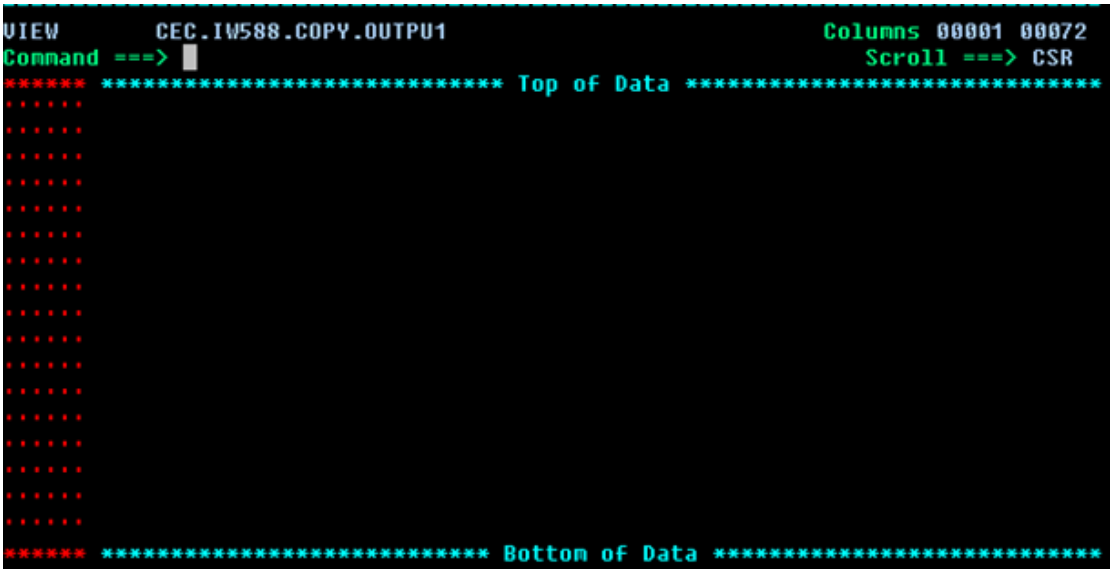
```
VIEW       CEC.IW588.COPY.INPUT                          Columns 00001 00072
Command ===>                                               Scroll ===> CSR
****** ***************************** Top of Data ******************************
000100 ANIL                          NUM
000200 BABU                          NU1
000300 CHRISTOPHER                   NU2
000400 CHARLES                       NUM
000500 DANIEL                        NUM2
000600
000700
****** **************************** Bottom of Data ****************************
```

If the condition is not met, i.e., the input datasets contains records, then RC=00 is returned.

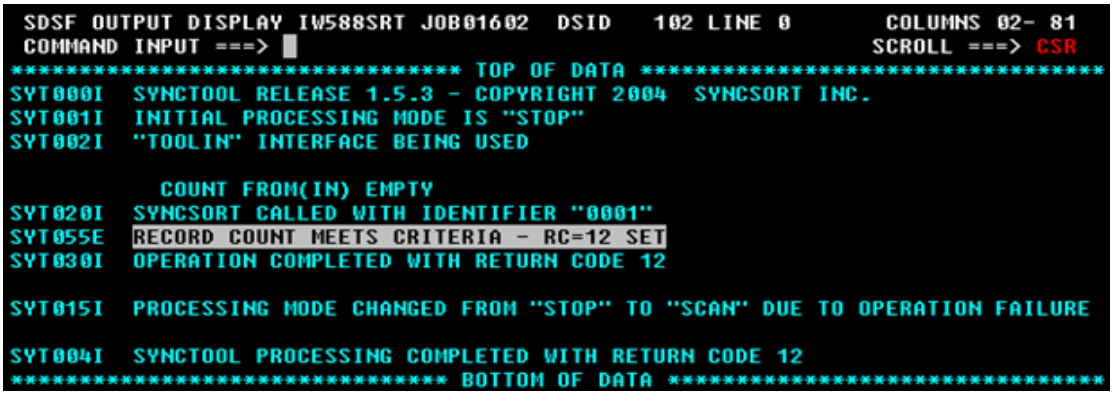
TOOLMSG:

```
 SDSF OUTPUT DISPLAY IW588SRT JOB01181  DSID   102 LINE 0
 COMMAND INPUT ===>
******************************** TOP OF DATA ********************************
SYT000I  SYNCTOOL RELEASE 1.5.3 - COPYRIGHT 2004  SYNCSORT INC
SYT001I  INITIAL PROCESSING MODE IS "STOP"
SYT002I  "TOOLIN" INTERFACE BEING USED

           COUNT FROM(IN) EMPTY
SYT020I  SYNCSORT CALLED WITH IDENTIFIER "0001"
SYT056I  RECORD COUNT DOES NOT MEET CRITERIA - RC=0 SET
SYT030I  OPERATION COMPLETED WITH RETURN CODE 0

SYT004I  SYNCTOOL PROCESSING COMPLETED WITH RETURN CODE 0
******************************** BOTTOM OF DATA ********************************
```

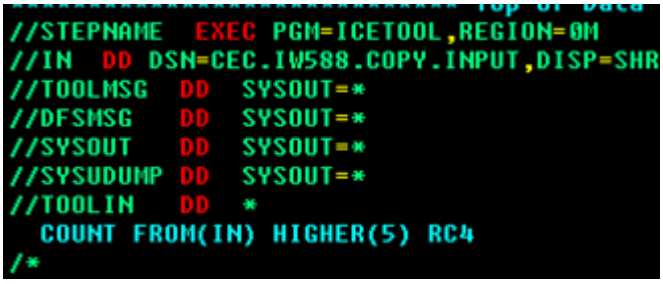
Suppose consider the input dataset is empty,



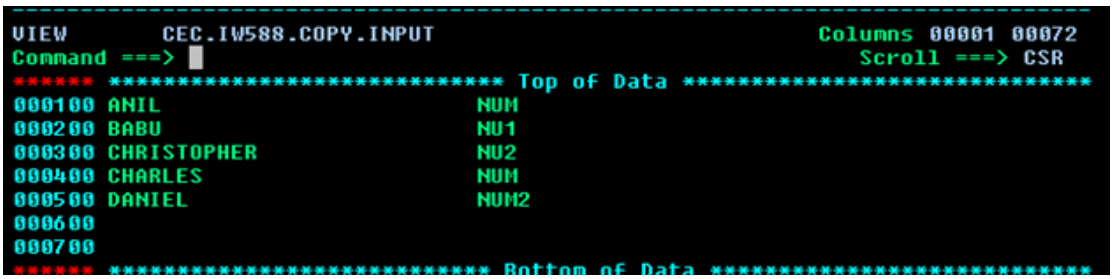
TOOLMSG: If the condition is met, i.e., the input dataset DOES NOT contains records, then RC=12 is returned



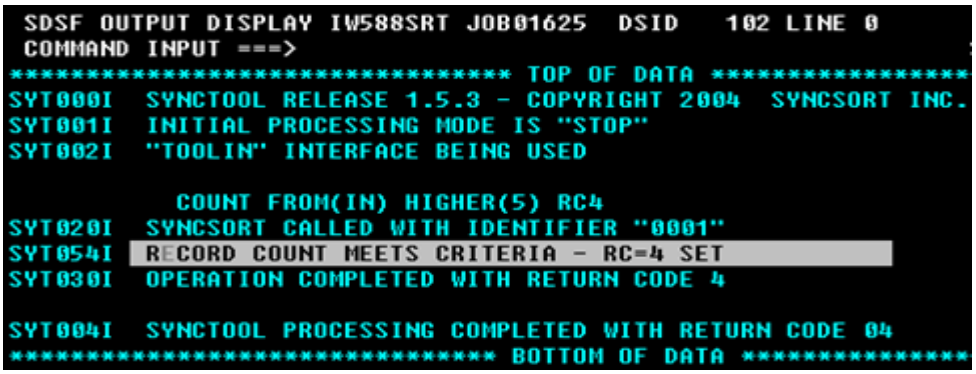
- Count Operator with HIGHER and RC - To Check for a condition say, whether the input dataset contains records more than the specified value and to return with a RC also with specified value.



Input dataset:



If criteria is met, i.e., if the count of records is greater than 5 , then the return code will be 4.



- **SELECT -** Selects records from the indd data set for inclusion in the outdd data set based on meeting criteria for the number of times specified numeric and/or character field values occur.

  Syntax:

  SELECT FROM(indd) TO(outdd) ON(p,m,f)

  Select operator can also be used for many other operations and the syntax is given below:

  ALLDUPS,NODUPS,HIGHER(x),LOWER(y),EQUAL(v),FIRST,FIRST(u),

  LAST,FIRSTDUP,FIRSTDUP(w),LASTDUP,LASTDUP(w).

Examples: Input dataset

```
VIEW       CEC.IW588.COPY.INPUT                              Columns 00001 00072
Command ===>                                                   Scroll ===> CSR
****** ***************************** Top of Data ******************************
000100 ANIL                          NUM
000200 BABU                          NU1
000300 CHRISTOPHER                   NU2
000400 CHARLES                       NUM
000500 DANIEL                        NUM2
000600
000700
****** **************************** Bottom of Data ****************************
```

- Select Operator with ALLDUPS - To fetch all records including duplicate record based on the key value.

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.INPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPU2,DISP=SHR
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//TOOLIN    DD   *
  SELECT FROM(IN) TO(OUT) ON(28,2,CH) ALLDUPS
/*
```

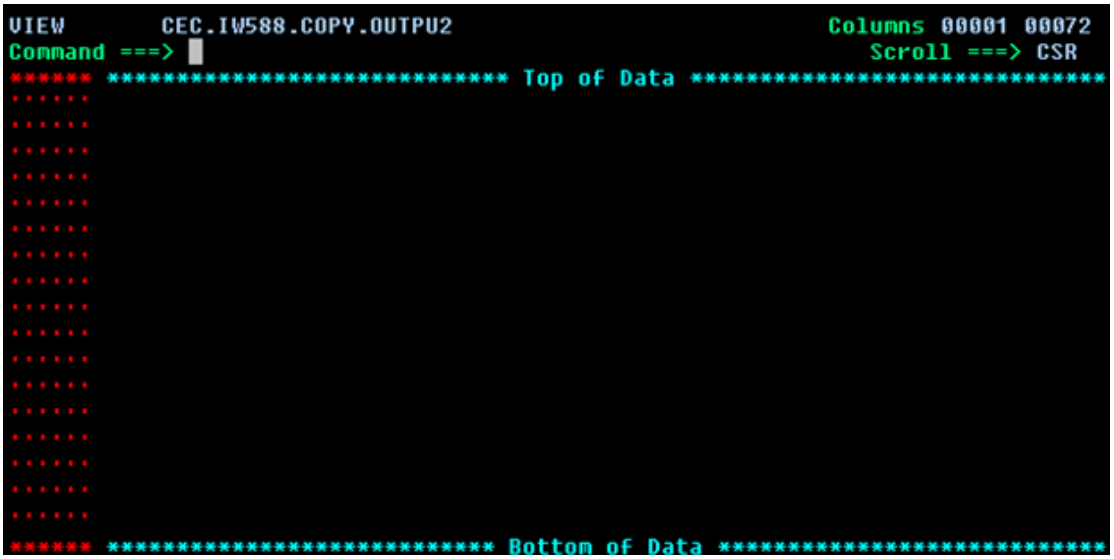
Output:

```
VIEW       CEC.IW588.COPY.OUTPU2                             Columns 00001 00072
Command ===>                                                   Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001
000002
000003 ANIL                          NUM
000004 BABU                          NU1
000005 CHRISTOPHER                   NU2
000006 CHARLES                       NUM
000007 DANIEL                        NUM2
****** **************************** Bottom of Data ****************************
```

- Select Operator with NODUPS - To fetch only unique records based on the key value.

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.INPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPU2,DISP=SHR
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//TOOLIN    DD   *
  SELECT FROM(IN) TO(OUT) ON(28,2,CH) NODUPS
/*
```

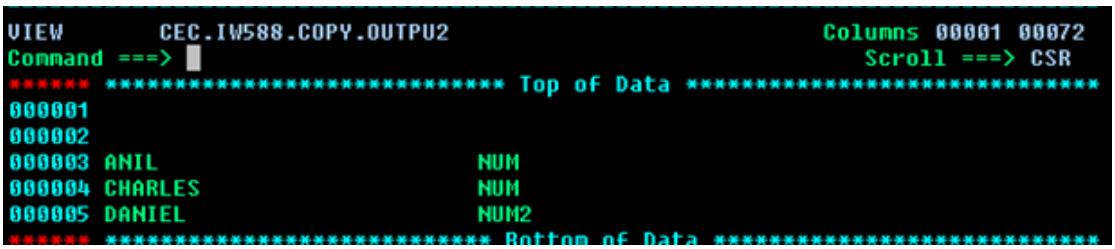
Output:There is no records which is unique in the input.So the output is mpty.



- Select Operator with HIGHER - To fetch record which are more than one based on key value.

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.INPUT,DISP=SHR
//OUT    DD DSN=CEC.IW588.COPY.OUTPU2,DISP=SHR
//TOOLMSG  DD  SYSOUT=*
//DFSMSG   DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//TOOLIN   DD  *
  SELECT FROM(IN) TO(OUT) ON(28,3,CH) HIGHER(1)
/*
```

Output: There are 3 records of 'NUM' and 2 empty record which met the criteria HIGHER(1).



- Select Operator with FIRSTDUP - To fetch the first duplicate record based on key value.

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.INPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPU2,DISP=SHR
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//TOOLIN    DD   *
  SELECT FROM(IN) TO(OUT) ON(28,3,CH) FIRSTDUP
/*
```

Output:

```
VIEW       CEC.IW588.COPY.OUTPU2                          Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001
000002 ANIL                          NUM
****** **************************** Bottom of Data ****************************
```

- Select Operator with FIRST - To fetch the first record based on key value.

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.INPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPU2,DISP=SHR
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//TOOLIN    DD   *
  SELECT FROM(IN) TO(OUT) ON(28,3,CH) FIRST
/*
```

Output:

```
VIEW       CEC.IW588.COPY.OUTPU2                          Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001
000002 ANIL                          NUM
000003 BABU                          NU1
000004 CHRISTOPHER                   NU2
****** **************************** Bottom of Data ****************************
```

- Select Operator with EQUAL - To fetch the record based on key value which have occurred only once.

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.INPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPU2,DISP=SHR
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//TOOLIN    DD   *
  SELECT FROM(IN) TO(OUT) ON(28,3,CH) EQUAL(1)
/*
```

Output:

```
VIEW       CEC.IW588.COPY.OUTPU2                          Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000200 BABU                        NU1
000300 CHRISTOPHER                 NU2
****** **************************** Bottom of Data ****************************
```

- Select Operator with CNTL - To fetch the record based on key value using the control card.

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.INPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPUT,DISP=SHR
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//TOOLIN    DD   *
  SELECT FROM(IN) TO(OUT) ON(28,3,CH) LASTDUP USING(CTL1)
//CTL1CNTL DD   *
  OMIT COND=(1,1,CH,EQ,C'C')
/*
```

Output:

```
VIEW       CEC.IW588.COPY.OUTPUT                          Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001
000002 DANIEL                      NUM2
****** **************************** Bottom of Data ****************************
```

- **OCCUR -** Prints each unique value for specified numeric and character fields and how many times it occurs in a separate list data set. Simple or tailored reports can be produced. The values printed can be limited to those for which the value meets specified criteria (e.g. only duplicate values).

Syntax:

OCCUR FROM(indd) LIST(listdd) ON(p,m,f)

Occur operator can also be used for many other operations and the syntax is given below:

ALLDUPS,NODUPS,HIGHER(x),LOWER(y),EQUAL(v)

Example:

Input Dataset:

```
VIEW       CEC.IW588.COPY.INPUT                              Columns 00001 00072
Command ===>                                                   Scroll ===> CSR
****** ***************************** Top of Data ********************************
000100 ANIL                          NUM
000200 BABU                          NU1
000300 CHRISTOPHER                   NU2
000400 CHARLES                       NUM
000500 DANIEL                        NUM2
000600
000700
****** **************************** Bottom of Data ******************************
```

- Occur Operator to count the number of records based on key value.

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.INPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPUT,DISP=SHR
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//TOOLIN    DD   *
  OCCUR FROM(IN) LIST(OUT) ON(28,3,CH) ON(VALCNT)
/*
```

Output:

```
VIEW       CEC.IW588.COPY.OUTPUT                             Columns 00001 00072
Command ===>                                                   Scroll ===> CSR
****** ***************************** Top of Data ********************************
000001 1(28,3,CH)                   VALUE COUNT
000002                              0000000000000002
000003  NUM                         0000000000000003
000004  NU1                         0000000000000001
000005  NU2                         0000000000000001
****** **************************** Bottom of Data ******************************
```

- Occur Operator to count the number of records based on key value removing the header .

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.INPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPUT,DISP=SHR
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//TOOLIN    DD   *
  OCCUR FROM(IN) LIST(OUT) NOHEADER BLANK ON(28,3,CH) ON(VALCNT)
/*
```

Output:

```
VIEW       CEC.IW588.COPY.OUTPUT                          Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001                        2
000002  NUM                   3
000003  NU1                   1
000004  NU2                   1
****** **************************** Bottom of Data ****************************
```

- Occur Operator to count the number of records based on key value with specific check condition.

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.INPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPUT,DISP=SHR
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//TOOLIN    DD   *
  OCCUR FROM(IN) LIST(OUT) NOHEADER BLANK -
    ON(28,3,CH) ON(VALCNT) HIGHER(1)
/*
```

Output:

```
VIEW       CEC.IW588.COPY.OUTPUT                          Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001                        2
000002  NUM                   3
****** **************************** Bottom of Data ****************************
```

- **SPLICE –** Splices together specified fields from records with matching numeric or character field values (that is, duplicate values), but different information. This makes it possible to join fields from different types of input records to create an output record with information from two or more records.

  Syntax:

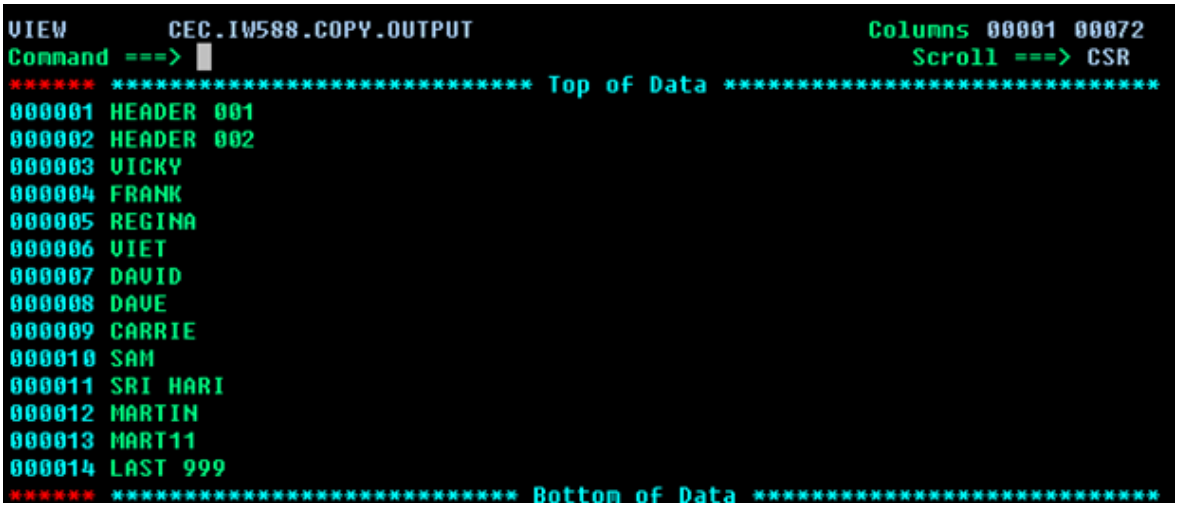
  SPLICE FROM(indd) TO(outdd) ON(p,m,f) ... WITH(p,m) ...

  Splice operator can also be used for many other operations and the syntax is given below:

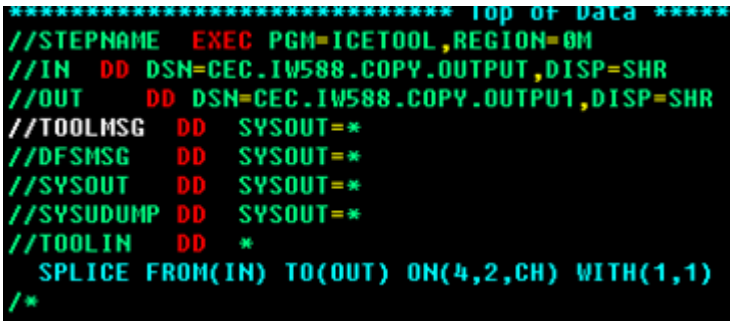
  KEEPNODUPS,KEEPBASE,WITHALL,WITHANY,WITHEACH

  Examples:

  Input Dataset:



  - Splice Operator - To fetch the value based on the Key value. And splice them based on the data in the WITH field.By default, the first duplicate is spliced with all of the WITH fields from the last duplicate.



Output:

```
VIEW       CEC.IW588.COPY.OUTPU1                          Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 HEADER 001
000002 LIET
****** **************************** Bottom of Data ****************************
```

Input Dataset(The dataset has been slightly modified)

```
VIEW       CEC.IW588.COPY.OUTPUT                          Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 HEADER 001
000002 HEADER 002
000003 VICKY
000004 FRANN
000005 REANNA
000006 VIET
000007 DAVID
000008 DAVE
000009 CARRIE
000010 SARR
000011 XXRR
000012 YYRR
000013 SRI HARI
000014 MARTIN
000015 MARTI1
000016 LAST 999
****** **************************** Bottom of Data ****************************
```

- Splice Operator(Another example using WITH)

Output: The first character of SARR is replaced with Y from the third and last duplicate word YYRR.

```
VIEW       CEC.IW588.COPY.OUTPU1                          Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 HEADER 001
000002 RRANN
000003 YARR
000004 LIET
000005 MARTIN
****** **************************** Bottom of Data ****************************
```

- Splice Operator using WITHEACH - the first duplicate is spliced with one specified WITH field from each subsequent duplicate (overriding the default of splicing the first duplicate with all of the specified WITH fields from the last duplicate).

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.OUTPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPU1,DISP=SHR
//TOOLMSG  DD  SYSOUT=*
//DFSMSG   DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//TOOLIN   DD  *
  SPLICE FROM(IN) TO(OUT) ON(4,2,CH) -
  WITHEACH WITH(1,1)
/*
```

Output:

```
VIEW       CEC.IW588.COPY.OUTPU1                          Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 HEADER 001
000002 RRANN
000003 XARR
000004 LIET
000005 MARTIN
****** **************************** Bottom of Data ****************************
```

- Splice Operator using WITHALL - WITHALL can be used to splice the first duplicate with all of the WITH fields from the second and subsequent duplicates.With all with splice all fisrt duplicates first values, with the all duplicates first value.

```
//IW588SRT  JOB B02645,
//             IW588,         **JOB STATEMENT GENERATED
//             NOTIFY=IW588,CLASS=F,
//             MSGLEVEL=(1,1)
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.OUTPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPU1,DISP=SHR
//TOOLMSG  DD  SYSOUT=*
//DFSMSG   DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//TOOLIN   DD  *
  SPLICE FROM(IN) TO(OUT) ON(4,2,CH) WITHALL WITH(1,1)
/*
```

Output:

```
VIEW       CEC.IW588.COPY.OUTPU1                          Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 HEADER 001
000002 RRANN
000003 XARR
000004 YARR
000005 LIET
000006 MARTIN
****** **************************** Bottom of Data ****************************
```

- Splice Operator using KEEPNODUPS - non-duplicates are kept (unchanged) along with the spliced records (overriding the default of deleting non-duplicates).

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.OUTPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPU1,DISP=SHR
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//TOOLIN    DD   *
  SPLICE FROM(IN) TO(OUT) ON(4,2,CH) -
  WITHALL WITH(1,1) KEEPNODUPS
/*
```

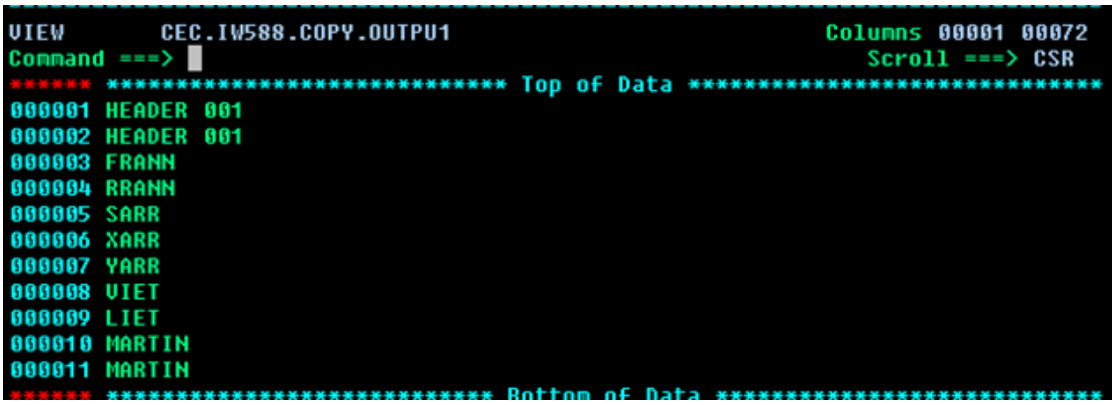
Output:

```
VIEW       CEC.IW588.COPY.OUTPU1                          Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ********************************
000001 SRI HARI
000002 HEADER 001
000003 DAVE
000004 DAVID
000005 VICKY
000006 RRANN
000007 XARR
000008 YARR
000009 CARRIE
000010 LIET
000011 MARTIN
****** **************************** Bottom of Data ****************************
```

- Splice Operator using KEEPBASE - the base records are kept (unchanged) along with the spliced records (overriding the default of deleting the base records).

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN   DD DSN=CEC.IW588.COPY.OUTPUT,DISP=SHR
//OUT     DD DSN=CEC.IW588.COPY.OUTPU1,DISP=SHR
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//TOOLIN    DD   *
  SPLICE FROM(IN) TO(OUT) ON(4,2,CH) -
  WITHALL WITH(1,1) KEEPBASE
/*
```

Output:



Things to Remember

- For more than 1 input datasets, should be of same record length.
- The intermediate file should of disposition (MOD,PASS).
- Each set of control statements should be separate control cards in production job.
- Maximum number of ON operand can be 10.
- Maximum number of WITH operand can be 50.
- Try using BUILD rather than OUTREC in ICETOOL.

Example

Using more than one operator in the single job

- With **COPY** and **SELECT** Operators:

  //S2 EXEC PGM=ICETOOL

  //TOOLMSG DD SYSOUT=

  //DFSMSG DD SYSOUT=

  //IN1 DD DSN=... input file1 (FB/25)

  //IN2 DD DSN=... input file2 (FB/15)

  //T1 DD DSN=&&T1,UNIT=SYSDA,SPACE=(CYL,(5,5)),

  // USE MOD FOR T1

  // DISP=(MOD,PASS)

  //OUT DD DSN=... output file (FB/25)

  //TOOLIN DD

  COPY FROM(IN2) TO(T1)

  COPY FROM(IN1) TO(T1) USING(CTL2)

```
SELECT FROM(T1) TO(OUT) ON(1,9,CH) FIRSTDUP
/*
//CTL2CNTL DD
INREC BUILD=(1:7,9,15:X)
/*
```

- With **COPY** and **SELECT** operators(Similar example)

Input1:

```
VIEW       CEC.IW588.COPY.INPUT1                              Columns 00001 00072
Command ===>                                                    Scroll ===> CSR
=COLS> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
****** ***************************** Top of Data ******************************
000001 12345  NIRMAL              NUM
000002 12314  SANDHYA             NU1
000003 12345  SUKHESH             NU2
000004 11231  SATISH              NUM
000005 11223  KIRAN               NUM2
****** **************************** Bottom of Data ****************************
```

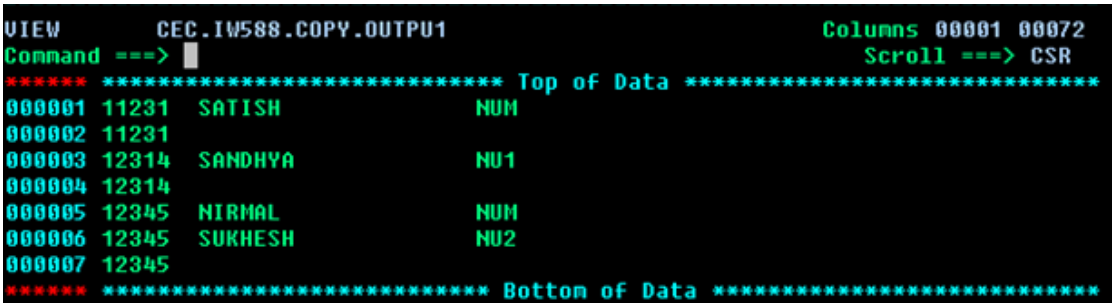
Input2:

```
VIEW       CEC.IW588.COPY.INPUT2                              Columns 00001 00072
Command ===>                                                    Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001          100 BRIGHT          12314
000002          20  POOR            12345
000003          80  ABOVE AVERAGE   11231
****** **************************** Bottom of Data ****************************
```

JCL:

```
//STEPNAME  EXEC PGM=ICETOOL,REGION=0M
//IN1 DD DSN=CEC.IW588.COPY.INPUT1,DISP=SHR
//IN2 DD DSN=CEC.IW588.COPY.INPUT2,DISP=SHR
//TEMP   DD DSN=CEC.IW588.TEMP.DATASET,
// DISP=(MOD,PASS),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
// UNIT=WORK,SPACE=(CYL,(5,1),RLSE)
//OUT    DD DSN=CEC.IW588.COPY.OUTPU1,DISP=SHR
//TOOLMSG  DD  SYSOUT=*
//DFSMSG   DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//TOOLIN   DD  *
  COPY FROM(IN1) TO(TEMP)
  COPY FROM(IN2) TO(TEMP) USING(CTL2)
  SELECT FROM(TEMP) TO(OUT) ON(1,5,CH) ALLDUPS
//CTL2CNTL DD *
  INREC BUILD=(1:30,5,6:X)
/*
```

Output:



- With **COPY** and **SPLICE** operators:

```
//SORTITEM EXEC PGM=ICETOOL
//TOOLMSG  DD SYSOUT=*
//DFSMSG   DD SYSOUT=*
//IN1      DD DSN=BEX.HX014.EXPLRQMT.G4847V00.FEB19.B01,DISP=SHR
//IN2      DD DSN=BEX.EE631.DATAPLAT.ORIGINAL.GV2515.FEB19.B01,DISP=SHR
//OUT1     DD DSN=CEC.TEST.ITMEXT.ECONULL(+1),
// *DCB Parameters
//OUT2     DD DSN=BEX.HX014.EXPLRQMT.VALID,
// *DCB Parameters
//TOOLIN DD *
 COPY FROM(IN1) TO(OUT1) USING(CTL1)
 COPY FROM(IN2) TO(OUT1) USING(CTL2)
 SPLICE FROM(OUT1) TO(OUT2) ON(1,7,CH)  -
 WITHALL  WITH(702,1) USING(CTL3)
/*
```