

UNIT 7

CHARACTER HANDLING

STRING

Two or more fields can be concatenated or ‘string’ in to single-field.

STRING	(id- 1 literal), (id-2 literal)
DELIMITED BY	(id- 3 literal size space),.....
	(id- 4 literal)
DELIMITED BY	(id- 4 literal), (id 5 literal).....
INTO id – 7	(with pointer id- 6).
	(ON OVERFLOW statement)

Figure 7-1 STRING

Notes:

‘Delimited by’ clause specifies how the fields are concatenated, Its usage is described in following foils.

‘With pointer’ option if used gives the total no. of characters in the concatenated field (id-7)

If the length of id-7 is not enough to hold the transferred characters, the statement after ‘on overflow’ option is executed.

One STRING statement can be written instead of a series of MOVE statements.

The following rules should be followed when this verb is used.

- (i). This statement is used to concatenate one or more strings into one by placing them side by side.
- (ii). Sending strings may be alphanumeric literals, figurative constants or identifiers with usage DISPLAY.
- (iii). The receiving string, i.e., identifier- 7 must also be with usage DISPLAY.

STRING example :

```

77  id-1  PIC  X(7)  VALUE 'Con, Con'.
77  id-2  PIC  X(7)  VALUE 'Cat, Cat'.
77  id-3  PIC  X(6)  VALUE 'enated'.
77  id-4  PIC  X(12) VALUE Spaces.
77  counter PIC 9 (3)  VALUE zeros.

```

STRING id-1, id-2, id-3.

DELIMITED BY id-2, id-3, into id-4.

WITH POINTER Counter.

Figure 7-2 STRING

RESULT

Content of id-4: CONCATENATED.

The following example shows usage of 'size' option.

```

77 F – 1            PIC  X(7) VALUE 'HIGHTEC'
77 F – 2            PIC  X(7) VALUE SPACES.
STRING 'MAIN' DELIMITED BY SIZE F – 1 INTO F – 2.

```

Guess the result.

ILLUSTRATES STRING VERB

DATA DIVISION.

WORKING STORAGE-SECTION.

```
01  FULL-NAME      PIC  X(30).  
01  FIRST-NAME     PIC  X(10)    VALUE SPACES.  
01  MIDDLE-NAME    PIC  X (10)   VALUE SPACES.  
01  LAST-NAME      PIC  X (10)   VALUE SPACES.
```

PROCEDURE DIVISION.

100-MAIN-PARA.

 MOVE 'RAJA' TO FIRST-NAME.

 MOVE 'ROMOHAN' TO MIDDLE-NAME.

 MOVE 'ROY' TO LAST-NAME.

 STRING FIRST-NAME,MIDDLE-NAME,LAST-NAME DELIMITTED BY
 SPACE INTO FULL-NAME.

UNSTRING Statement

The **UNSTRING** statement is used to split a single data item into several data items

Format:

```
>>__UNSTRING__identifier-1____>
>__
|_DELIMITED__ _____ identifier-2__ _____|
|_____ |_BY_| |_ALL_| |_literal-1_____| |<_____ |
|_____ |_OR_ _____ identifier-3_ _| |
|_____ |_ALL_| |_literal-2_____| |
>__INTO____>
<_____
>__identifier-4__ _____|
|_DELIMITER__ _____ identifier-5 _| |_COUNT__ _____ identifier-6_____|
|_____ |_IN_| |_____ |_IN_|
>__
|_ _____ POINTER__ identifier-7_| |_TALLYING__ _____ identifier-8_|
|_WITH_| |_____ |_IN_|
>__
|_ _____ OVERFLOW__ imperative-statement-1 ____|
|_WITH_|
>__
|_ NOT__ _____ OVERFLOW__ imperative-statement_2_| |_END-UNSTRING_|
|_ON_|
```

Figure 7-3 UNSTRING Statement

Notes:

The UNSTRING statement causes contiguous data in a sending field to be separated and placed into multiple receiving fields.

One UNSTRING statement can take the place of a series of MOVE statements, except that evaluation or calculation of certain elements is performed only once, at the beginning of the execution of the UNSTRING statement.

When the TALLYING phrase is specified, the field-count field contains a value equal to the initial value, plus the number of data receiving areas acted upon.

ILLUSTRATES UNSTRING VERB

DATA DIVISION.

WORKING STORAGE-SECTION.

01	FULL-NAME	PIC	X(30).	
01	FIRST-NAME	PIC	X(10)	VALUE SPACES.
01	MIDDLE-NAME	PIC	X (10)	VALUE SPACES.
01	LAST-NAME	PIC	X (10)	VALUE SPACES.

PROCEDURE DIVISION.

100-MAIN-PARA.

 MOVE 'RAJA ROMOHAN ROY' TO FULL-NAME.

 UNSTRING FULL-NAME DELIMITTED BY SPACE INTO FIRST-NAME,
 MIDDLE-NAME, LAST-NAME.

EXAMINE Statement

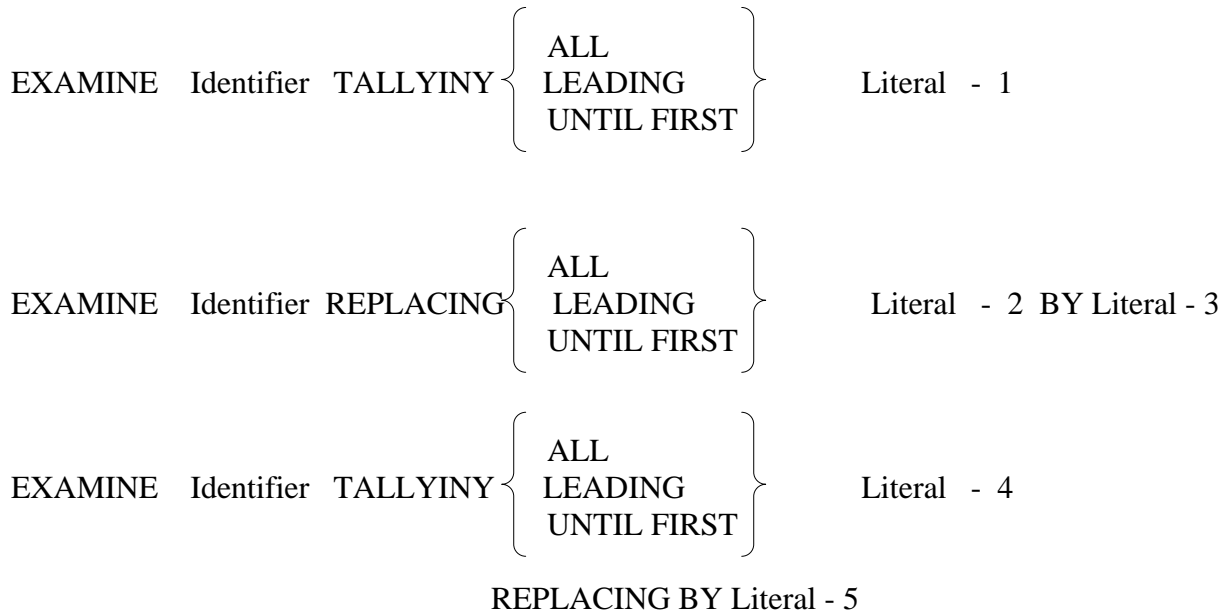


Figure 7-6 : EXAMINE

Notes:

This verb is used to scan a string to find the number of occurrences of a given character in it. In addition, the Verb can also be used to replace some or all occurrences of the said character by another character.

Eg :

Let us consider the following DATA DIVISION entry

77 A PIC X(5) VALUE IS "PPRIP".

Now the statement

EXAMINE A TALLYING ALL "P".

Will store 3 in the TALLY register as there are altogether three P's in the string. However the statement

EXAMINE A TALLYING LEADING "P"

Will store 2 in the TALLY , Since there are only Two leading P's.

The statement

EXAMINE A TALLYING UNTIL FIRST "I".

Will store 3 in the TALLY as there are only Three characters before the character I. It may be noted here that if a particular character is not found, TALLY is set to Zero, When the ALL or LEADING phrase is used. For eg. In the statement

EXAMINE A TALLYING LEADING "R"

Will set TALLY to zero, Since the leading character is not R. If the UNTIL FIRST phrase is used and the specified character is not found, the TALLY will contain the size of the string

EXAMINE---REPLACING.

EXAMINE A TALLYING ALL "P" REPLACING BY "Q"

Will store 3 in the TALLY register and will change the content of A to "QQRIQ".

EXAMINE A REPLACING FIRST "I" BY "M"

Will change the content of A to "PPRMP".

EXAMINE A REPLACING UNTIL FIRST "I" BY "Y"

Will change the content of A to "YYYIP".

In each of the cases A is assumed to be defined as before.

- It will count the occurrence of a specific character (alphabetic, numeric, or special character) in a data item.
- It will fill all or portions of a data item with specified characters, such as spaces or zeros.
- It will convert all occurrences of specific characters in a data item to user-supplied replacement characters.

```

>> __INSPECT__ identifier-1 ____ TALLYING _____>
<_____
|
> __identifier-2__ FOR ____ CHARACTERS ____ |_____|
|                                     |phrase 1 |
|                                     <_____
|                                     |
| ALL ____ identifier-3 ____
| LEADING ____ | literal ____ | phrase 1 |
|
> ____ REPLACING _____>
<_____
> ____ CHARACTER BY ____ identifier-5 ____ |_____|
|         |literal-3|      | phrase   |
|         <_____
| ALL ____ identifier-3 ____ BY ____ identifier-5 ____ |
| LEADING _ | literal-1 ____ | literal-3 ____ | phrase 1 |
|
phrase 1:
| BEFORE ____ identifier-4 ____
| AFTER ____ INITIAL ____ literal-2 ____

```

Unit 1. CHARACTER HANDLING AND

Notes:

TALLING Phrase counts the occurrence of a specific character (alphabetic, numeric, or special character) in a data item.

Identifier-1:

Identifier-1 is the inspected item and can be any of the following:

- An alphanumeric data item
- An numeric data item with USAGE DISPLAY
- An external floating point item

Identifier-2

Is the count field, and must be an elementary integer item defined without the symbol P in its PICTURE character-string

You must initialize identifier-2 before execution of the INSPECT statement begins.

Identifier-3 or literal-1

Is the tallying field (the item whose occurrences will be tallied).

Eg:

PROCEDURE DIVISION.

INSPECT HELLO TALLYING TALLY-COUNT FOR ALL "A"

Let the picture of HELLO be X(20) and suppose its content before the execution of the above statement is as follows :

APARNA*b***KUMARI***b***AMMA***b*****

If picture of TALLY-COUNT is 9(2) and originally contains 08, then after the execution of the statement, TALLY-COUNT will contain 14, as there are a total of 6 A's in HELLO.

If ALL in the statement is changed to LEADING , TALLY-COUNT will be increased to 9, as there is only one leading A. if CHARACTERS is specified instead of ALL "A", TALLY-COUNT will be increased to 28 as there are a total of 20 characters in HELLO.

When REPLACING CHARACTERS is used the identifier-5 must be 1 character in length

```
>> __INSPECT__identifier-1__REPLACING_____>
<_____
```

```

> _____ _CHARACTERS BY_ _____ _identifier-5_ _____ | _____ | _____>
|                                     |_____||_ phrase 2 |__|
|                                     <_____>
|                                     <_____> |
|_ ALL _____ identifier-3_ _BY_ _____ identifier-5_ _|_ |_
|_ LEADING_ |_ literal-1_ |_ literal-3_ |_ |_ phrase 1 |_

```

BEFORE	<i>identifier-4</i>	
AFTER	INITIAL	literal-2

```
INSPECT DATA1 REPLACING ALL " " BY "0"
INSPECT DATA2 REPLACING FIRST ZERO BY SPACE
INSPECT DATA3 REPLACING CHARACTER ZERO BY "X"
INSPECT DATA4 REPLACING LEADING "0" BY SPACE
```

Figure 7-5 The ALL Subscript

REPLACING Phrase

-identifier-3 or literal-1

Is the subject field (the item whose occurrences are replaced).

Identifier-3 can be:

- An elementary alphanumeric data item
- A numeric data item with USAGE DISPLAY
- An external floating point item

Literal-1 must be non-numeric, and can be any figurative constant that does not begin with the word ALL. If literal-1 is a figurative constant, it is considered to be a 1-character nonnumeric literal.

Identifier-5 or literal-3

Is the substitution field

Identifier-5 can be:

- An elementary alphanumeric data item
- A numeric data item with USAGE DISPLAY
- An external floating point item

Literal-3 must be nonnumeric, and can be any figurative constant that does not begin with the word ALL.

The following replacement rules apply:

When the subject field is a figurative constant, the single-character substitution field(which must be 1 character in length) replaces each character in the inspected item equivalent to the figurative constant.

When the substitution field is a figurative constant, the substitution field replaces each non-overlapping occurrence of the subject field in the inspected item.

When the subject and substitution fields are character-strings, the character-string specified in the substitution field replaces each non-overlapping occurrence of the subject field in the inspected item