# Internal Table Operations

## Append

It is used to insert data at the last of the internal table

We will pass the record to the work area and then from the work area we will pass it to the internal table

After every **APPEND** statement, we should clear the work area using **CLEAR.**

A screenshot of a computer code

Description automatically generated

Figure Append keyword.

## Loop AT Internal Table

A black text on a white background

Description automatically generated

Figure Loop AT Internal Table

## Delete

It is use to delete the record from the internal table

We can delete using the where condition or using index the whole row

DELETE lt\_data WHERE pm = ‘D’.

DELETE lt\_data INDEX 3.

## Modify

LOOP AT it\_data INTO wa\_data.

IF wa\_data-ono = 1.

wa\_data-pm = 'N'.

MODIFY it\_data FROM wa\_data TRANSPORTING pm.

ENDIF.

ENDLOOP.

Here TRANSPORTING pm means, we are changing in the pm column the fields that we are going to change should be included after the TRANSPORTING keyword.

## Read Table

It is used to read the first matching record from the internal table.

we can get a single value using the Read table so we can use either using loop or for single value using the where condition or an index

We can use it for only the internal table 'with key' keyword after that where the condition.

**READ TABLE it\_data INTO wa\_data with KEY ono = 4.**

**READ TABLE it\_data INTO wa\_data INDEX 3**.

## Clear and Refresh

It is used to clear the contents of the internal table

* We are setting the value of internal table to zero both means same we are clearing whole internal table

WRITE: / 'Before using the clear or refresh'.

loop at it\_data into wa\_data.

WRITE: / wa\_data-ono , wa\_data-pm.

endloop.

**while refreshing or clearing data from internal table or work are we can use the both keyword clear as well as refresh**

**but in case of work Area, we need to use clear only we cannot use refresh for work area**

CLEAR it\_data.

REFRESH it\_data.

WRITE: / 'After using the clear or refresh'.

loop at it\_data into wa\_data.

WRITE: / wa\_data-ono , wa\_data-pm.

endloop.

## Describe Table

It returns the number of records in the internal table

**DESCRIBE TABLE it\_data LINES lv\_data.**

## Sort

It is used to sort the internal table. If we are not specifying anything then by default It sorts in the ascending order. If we want to sort in descending order, then we need to specify the keyword descending

If we do the sorting on the multiple columns then it is not just sorting its sub-sorting

* **SORT lt\_data BY ono.**
* **SORT it\_data BY ono DESENDING.**
* **SORT it\_data BY ono pm DESENDING.**

**What we think that we are sorting the internal table based on 2 columns but ono in Ascending order and pm in** **descending order**

* **SORT it\_data BY ono DESENDING pm DESENDING.**

## Collect

It is used to make sum of amount values based upon unique character values.

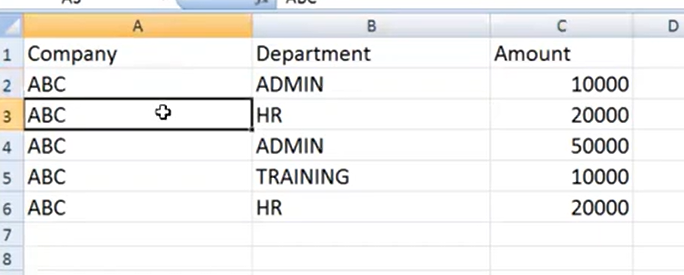


Figure RAW data

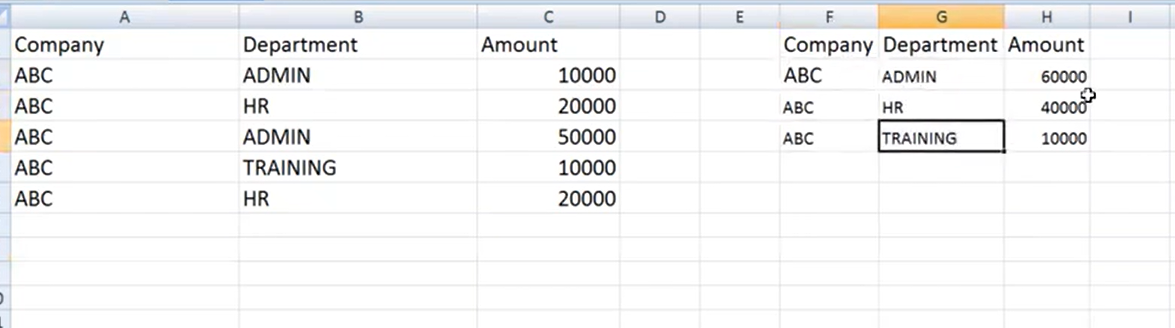


Figure ABC HR is unique character here on which we are doing sum

After collect data should go to different internal table

* **here make sure that after collect we should not store it in same internal table or else it will become endless loop**

LOOP AT it\_data into wa\_data.

COLLECT wa\_data into it\_temp\_data.

ENDLOOP.

# Tips

LOOP AT it\_data into wa\_data.

APPEND wa\_data into lt\_data.

ENDLOOP.

Here it becomes endless loop as we are reading from internal table and we are trying to append data in same internal table so it will become endless loop