Data Analysis Using VBA Programming Language

Sajjade Daraqshan Arshiya Zill-E-Ilahi Shaheen[1], Professor Dr.Abhijeet Kaiwade[2], Sajjade Gulfishan Tamkanat[3]
[1, 2, 3] Department Of Computer Science
[1, 2, 3] Shri Jagdish Prasad Jhabarmal Tibbrewala University, Vidyanagari, Jhunjhunu, Rajasthan

Abstract- *Excel is popular data analysis .Excel is comfortable using for every user. It is a desktop application data analysis in excel workbook using VBA [Visual basic Application]. It is so popular as a desktop application .It is a difficult task to state objective facts that satisfy a scientific approach. The language compared concerning. Excel takes care of no more mistakes.Excel is very powerful data handler. The VBA language concerning history and syntax and semantics.*

Keywords- Proper Case Conversion, Update Description, Comparing String and updating Corresponding Data, Clear Data Column.

I. INTRODUCTION

Static Scripting Languages have arrived in the mainstream market of Desktop Application languages. Excel is Powerful data analysis. VBA having certain properties are_

- They are statically typed
- They focus on short syntax
- They are interpreted directly compiled.

These properties are attractive for Desktop Application Development because they allow rapid prototyping, fast code changes and fast testing routines .Depending on the selected languages there may be also other differences that make that language valuable. However, this paper is not comparing not static languages to dynamic languages. In this introduction I will give some details about VBA language history, syntax and semantics and popularity of the language .VBA using in excel .Data comparison in different workbook or sheets, number comparison of two columns and updating numbers if they have exactly match of two different sheets or two different workbooks. In the end there will be conclusion listing all findings and presenting a recommendation based on the need of programmers and users.

II. VBA HISTORY

Visual Basic for application is an Object Oriented Programming Language. It is a software application consists of various individual objects, each of which has its own set of features and uses. An Excel application contents cells, worksheets, charts, pivot tables, drawing shapes. Each object has its own set of features which are called as properties and its own set of uses called method.

III. SYNTAX AND SEMANTICS

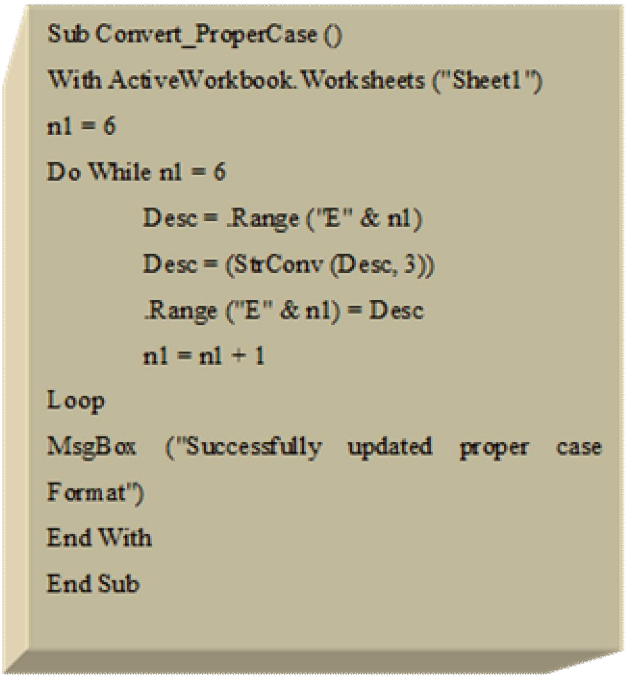
In this paper performing data analysis of different operation using visual basic application syntax and semantics of this languages.

[On Button Click Event]

A. Proper Case Conversion:-

In proper case conversion we have to do first character of the row should be in capital case letter. This function used for Description column data into proper case format. While writing this function programmer should remove special symbol, Special characters using suitable visual basic application code.

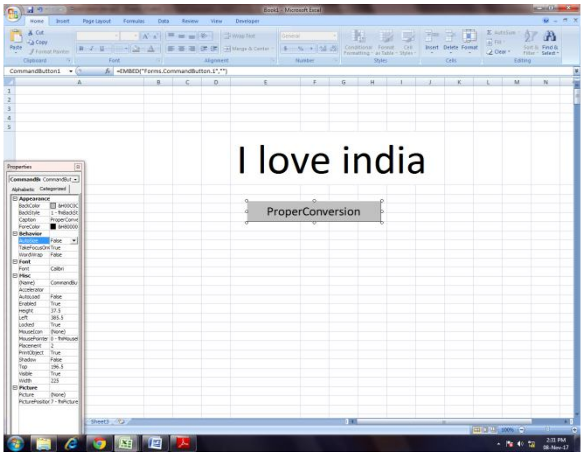
Procedure of Proper Case Conversion



```
Sub Convert_ProperCase ()
With ActiveWorkbook.Worksheets ("Sheet1")
n1 = 6
Do While n1 = 6
        Desc = .Range ("E" & n1)
        Desc = (StrConv (Desc, 3))
        .Range ("E" & n1) = Desc
        n1 = n1 + 1
Loop
MsgBox ("Successfully updated proper case
Format")
End With
End Sub
```

Figure 1.

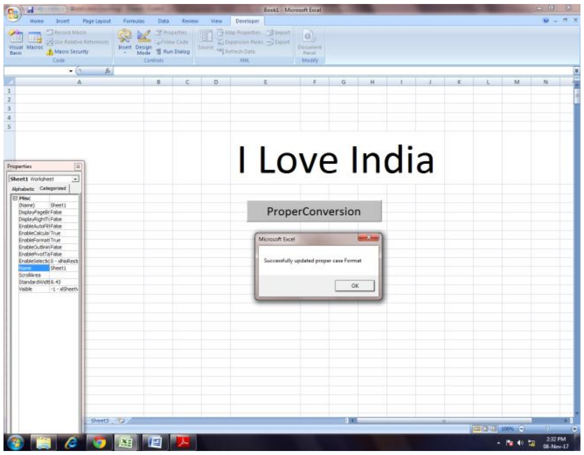
Experimental Results:-



I love india

Figure 2.

After



I Love India

Microsoft Excel

Successfully updated proper case Format

Figure 3.

Output generated by Procedure of Proper Case Conversion program.

B. **Update Description:-**

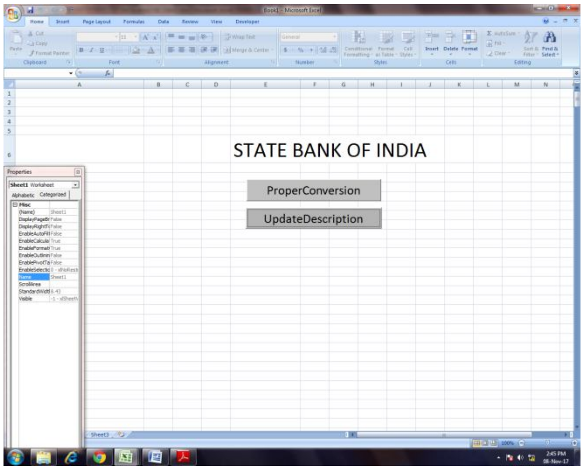
Reducing the sentence and making one words first letter should be in capital letter using this VBA function.

Function of Update Description

```
Private Sub CommandButton2_Click()
'Function Generate Description ()
With ActiveWorkbook.Worksheets("Sheet1")
   n1 = 6
  'LastRow = .UsedRange.Rows.Count
Do While n1 = 6
Desc = .Range("E" & n1)
Desc_Split = Split(Desc, " ")
Word_count = UBound(Desc_Split)
Word_count = Word_count + 1
  If Word_count = 2 Then
Desc_Text = Left(Desc_Split(0), 6)
Desc_Text = Desc_Text&Left(Desc_Split(1), 6)
ElseIfWord_count = 3 Then
Desc_Text = Left(Desc_Split(0), 4)
Desc_Text = Desc_Text&Left(Desc_Split(1), 4)
Desc_Text = Desc_Text&Left(Desc_Split(2), 4)
ElseIfWord_count = 4 Then
Desc_Text = Left(Desc_Split(0), 4)
Desc_Text = Desc_Text&Left(Desc_Split(1), 3)
Desc_Text = Desc_Text&Left(Desc_Split(2), 3)
Desc_Text = Desc_Text&Left(Desc_Split(3), 3)
ElseIfWord_count = 5 Then
Desc_Text = Left(Desc_Split(0), 3)
Desc_Text = Desc_Text&Left(Desc_Split(1), 2)
Desc_Text = Desc_Text&Left(Desc_Split(2), 2)
Desc_Text = Desc_Text&Left(Desc_Split(3), 2)
Desc_Text = Desc_Text&Left(Desc_Split(4), 2)
  End If
  n1 = n1 + 1
Loop
MsgBoxDesc_Text
End With
'End Function

End Sub
```

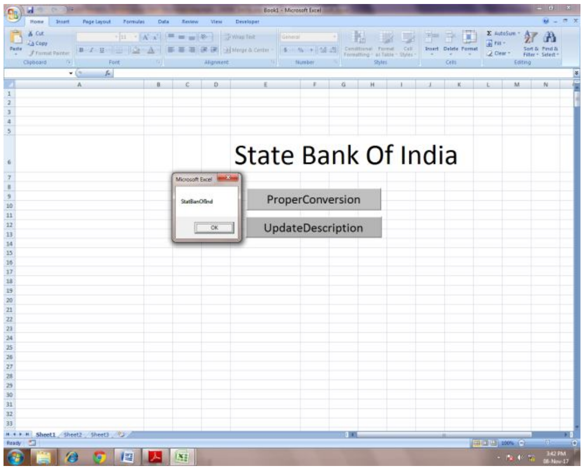
Experimental Results:-



STATE BANK OF INDIA

Figure 4.

After



State Bank Of India

Microsoft Excel

StatBanOfInd

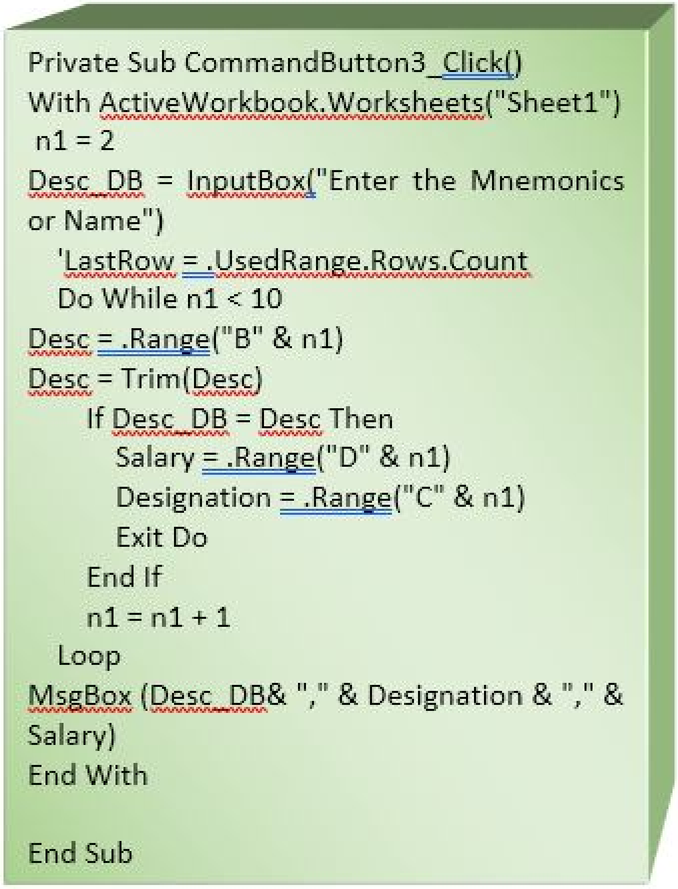
OK

Figure 5.

C. Update Numbers Corresponding Mnemonics:-

This function is equivalent the mnemonics of user data in column from workbook or work sheets then if it is found updating and fetching of corresponding row.

Procedure for Update Numbers Corresponding Mnemonics

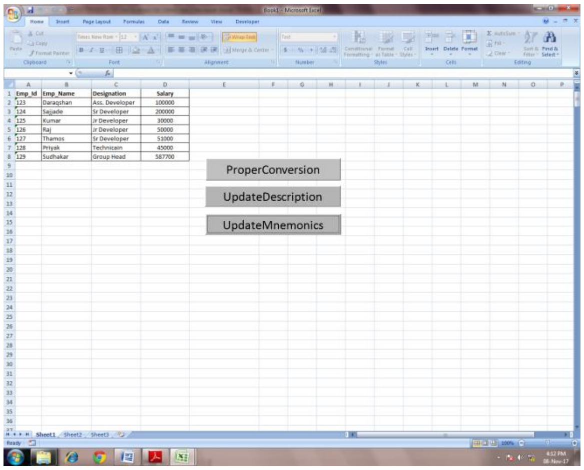


```
Private Sub CommandButton3_Click()
With ActiveWorkbook.Worksheets("Sheet1")
 n1 = 2
Desc_DB = InputBox("Enter the Mnemonics
or Name")
   'LastRow = .UsedRange.Rows.Count
   Do While n1 < 10
Desc = .Range("B" & n1)
Desc = Trim(Desc)
      If Desc_DB = Desc Then
         Salary = .Range("D" & n1)
         Designation = .Range("C" & n1)
         Exit Do
      End If
      n1 = n1 + 1
   Loop
MsgBox (Desc_DB& "," & Designation & "," &
Salary)
End With

End Sub
```

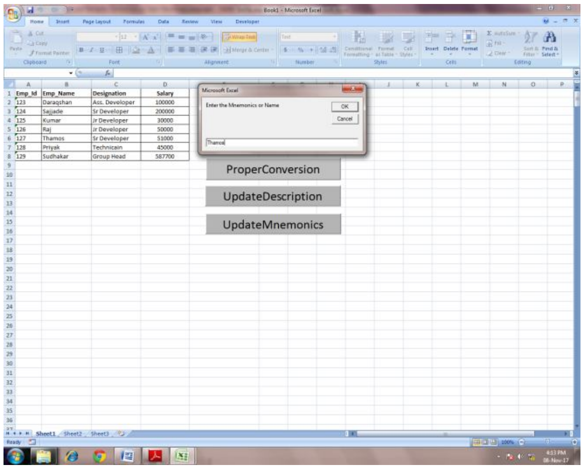
Figure 6.

Experimental Results:-



Emp_Id	Emp_Name	Designation	Salary
123	Daraqshan	Ass. Developer	100000
124	Sajjade	Sr Developer	200000
125	Kumar	Jr Developer	30000
126	Raj	Jr Developer	50000
127	Thamos	Sr Developer	51000
128	Priyak	Technicain	45000
129	Sudhakar	Group Head	587700

Figure 7.



Emp_Id	Emp_Name	Designation	Salary
123	Daraqshan	Ass. Developer	100000
124	Sajjade	Sr Developer	200000
125	Kumar	Jr Developer	30000
126	Raj	Jr Developer	50000
127	Thamos	Sr Developer	51000
128	Priyak	Technicain	45000
129	Sudhakar	Group Head	587700

Microsoft Excel

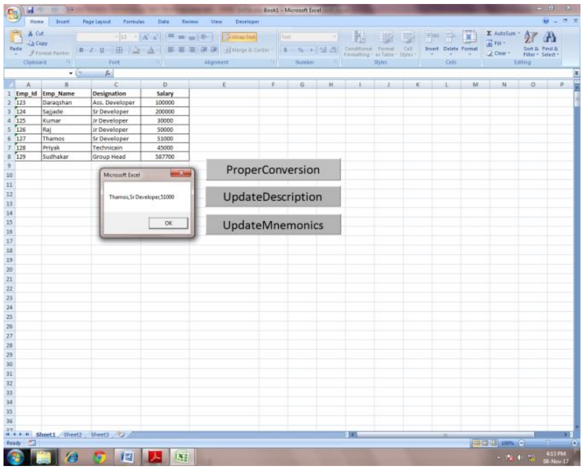
Enter the Mnemonics or Name

Thamos

OK

Cancel

Figure 8.



Emp_Id	Emp_Name	Designation	Salary
123	Daraqshan	Ass. Developer	100000
124	Sajjade	Sr Developer	200000
125	Kumar	Jr Developer	30000
126	Raj	Jr Developer	50000
127	Thamos	Sr Developer	51000
128	Priyak	Technicain	45000
129	Sudhakar	Group Head	587700

Microsoft Excel

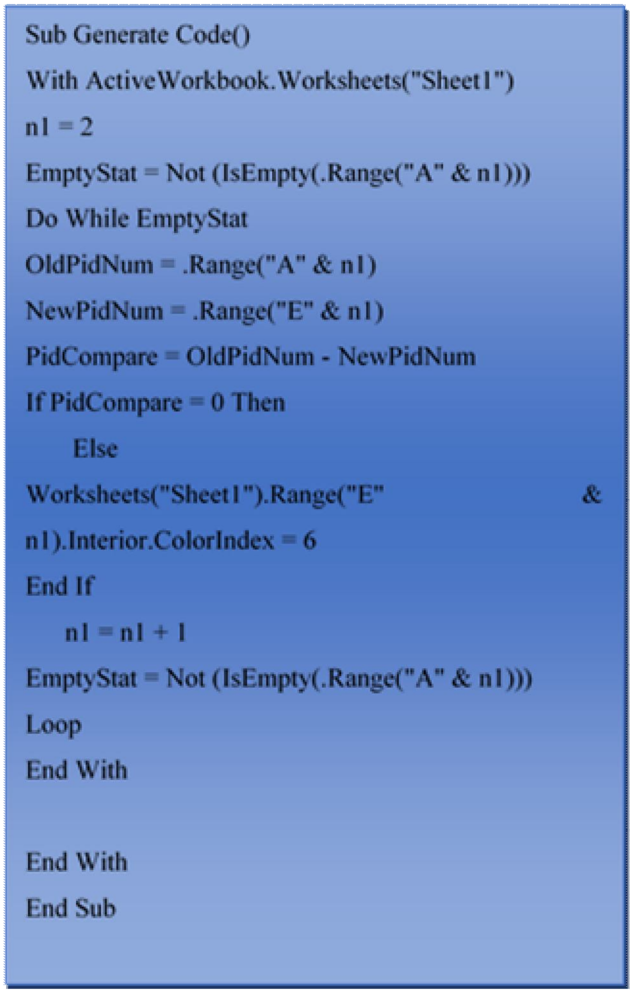
Thamos,Sr Developer,51000

OK

Figure 9.

D. Comparison of Two columns:-

This function is used to for checking or comprising of two columns data and it is different found making the yellow color of columns cell.



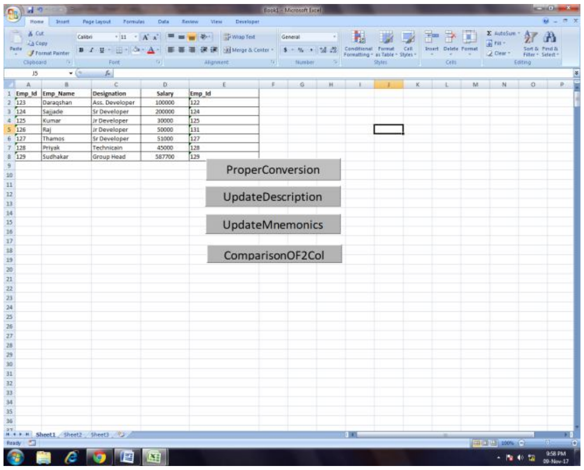
```
Sub Generate Code()
With ActiveWorkbook.Worksheets("Sheet1")
n1 = 2
EmptyStat = Not (IsEmpty(.Range("A" & n1)))
Do While EmptyStat
OldPidNum = .Range("A" & n1)
NewPidNum = .Range("E" & n1)
PidCompare = OldPidNum - NewPidNum
If PidCompare = 0 Then
    Else
Worksheets("Sheet1").Range("E"          &
n1).Interior.ColorIndex = 6
End If
    n1 = n1 + 1
EmptyStat = Not (IsEmpty(.Range("A" & n1)))
Loop
End With

End With
End Sub
```

Figure 10.

Before:-

Comparison of two columns value will not match then it will change of cell color. If value will found it will remain same.

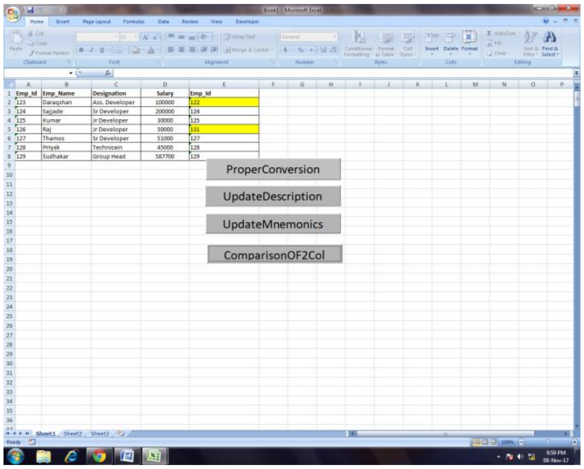


	A	B	C	D	E
1	Emp_Id	Emp_Name	Designation	Salary	Emp_Id
2	123	Daraqshan	Ass. Developer	100000	122
3	124	Sajjade	Sr Developer	200000	124
4	125	Kumar	Jr Developer	30000	125
5	126	Raj	Jr Developer	50000	131
6	127	Thamos	Sr Developer	51000	127
7	128	Priyak	Technicain	45000	128
8	129	Sudhakar	Group Head	587700	129

Figure 11.

After

Comparing value from column 'A' and column 'E' is not match Columns value then change will particular cell color.



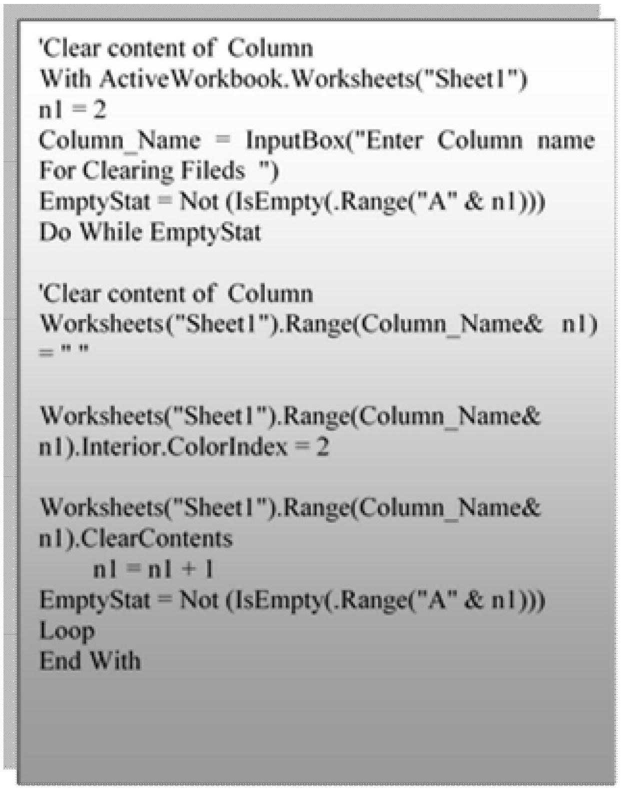
	A	B	C	D	E
1	Emp_Id	Emp_Name	Designation	Salary	Emp_Id
2	123	Daraqshan	Ass. Developer	100000	122
3	124	Sajjade	Sr Developer	200000	124
4	125	Kumar	Jr Developer	30000	125
5	126	Raj	Jr Developer	50000	131
6	127	Thamos	Sr Developer	51000	127
7	128	Priyak	Technicain	45000	128
8	129	Sudhakar	Group Head	587700	129

Figure 12.

E. Clearing Any Column:-

This procedure is used to clear the content of the cell and clear the color of the column.

Procedure of Clearing Any Column:-



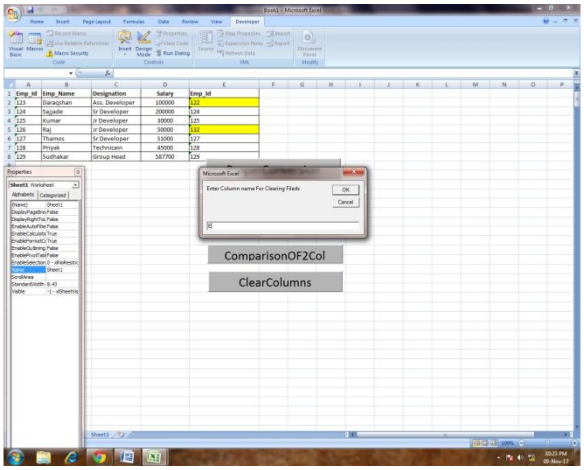
```
'Clear content of Column
With ActiveWorkbook.Worksheets("Sheet1")
n1 = 2
Column_Name = InputBox("Enter Column name
For Clearing Fileds ")
EmptyStat = Not (IsEmpty(.Range("A" & n1)))
Do While EmptyStat

'Clear content of Column
Worksheets("Sheet1").Range(Column_Name& n1)
= " "

Worksheets("Sheet1").Range(Column_Name&
n1).Interior.ColorIndex = 2

Worksheets("Sheet1").Range(Column_Name&
n1).ClearContents
     n1 = n1 + 1
EmptyStat = Not (IsEmpty(.Range("A" & n1)))
Loop
End With
```

Figure 13.



	A	B	C	D	E
1	Emp_Id	Emp_Name	Designation	Salary	Emp_Id
2	123	Daraqshan	Ass. Developer	100000	122
3	124	Sajjade	Sr Developer	200000	124
4	125	Kumar	Jr Developer	30000	125
5	126	Raj	Jr Developer	50000	132
6	127	Thamos	Sr Developer	51000	127
7	128	Priyak	Technicain	45000	128
8	129	Sudhakar	Group Head	587700	129

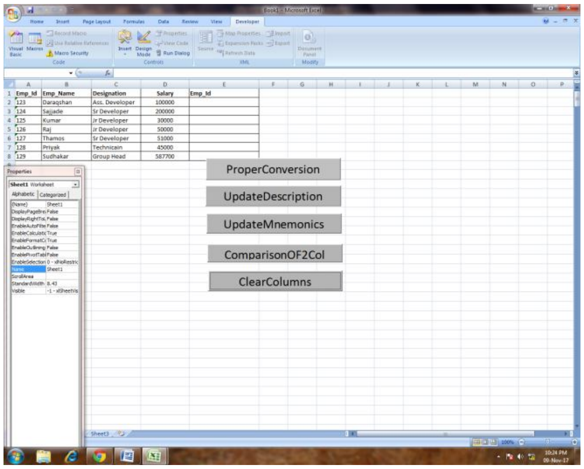
Sheet1 Worksheet

(Name)	Sheet1
DisplayPageBre	False
DisplayRightToL	False
EnableAutoFilte	False
EnableCalculatic	True
EnableFormatC	True
EnableOutlining	False
EnablePivotTabl	False
EnableSelection	0 - xlNoRestric
Name	Sheet1
ScrollArea	
StandardWidth	8.43
Visible	-1 - xlSheetVis

Enter Column name For Clearing Fileds

E

Figure 14.



	A	B	C	D	E
1	Emp_Id	Emp_Name	Designation	Salary	Emp_Id
2	123	Daraqshan	Ass. Developer	100000	
3	124	Sajjade	Sr Developer	200000	
4	125	Kumar	Jr Developer	30000	
5	126	Raj	Jr Developer	50000	
6	127	Thamos	Sr Developer	51000	
7	128	Priyak	Technicain	45000	
8	129	Sudhakar	Group Head	587700	

Sheet1 Worksheet

(Name)	Sheet1
DisplayPageBre	False
DisplayRightToL	False
EnableAutoFilte	False
EnableCalculatic	True
EnableFormatC	True
EnableOutlining	False
EnablePivotTabl	False
EnableSelection	0 - xlNoRestric
Name	Sheet1
ScrollArea	
StandardWidth	8.43
Visible	-1 - xlSheetVis

Figure 15.

IV. CONCLUSION

VBA programming is a useful for comparison for various fields. However ,huge amount of heterogeneous static , and Desktop applications a need for fast and effective programming techniques for the managing of data using excel.VBA also has serious problem of ambiguity from its manually data analysis. To solve the problem, system need expansive time and space consumption and the costs are getting more expansive with the explosive Desktop Application usage increase.

This paper proposes a fast programming using visual basic application with excel for data analysis for desktop application data managements. Using the data analysis comparison gives many advantages on both flexibility semantics compatibility and particle processing time. When we consider further extension of Static and Desktop application programming usages. The future work is data analysis using permutation and combination.

REFERENCES

[1] An Introduction to VBA in Excel By Rebort L McDonald

[2] www.tutorialspoint.com/vba/vba_quick_guide.html

[3] Notes on Visual Basic for Applications (VBA)By Konstantinos N. Vonatsos

[4] MicrosoftExcel VBA Programming for the Absolute Beginner Second Edition By Thomson Course Technology PTR