Lecture Notes – Session 12 – IS201

Brandt Redd – 2020-02-10

Cities – Reorganize Data

Start with the "Cities Spreadsheet" in Content – Data Sets

```
Option Explicit

Sub Reorganize()
    Dim citiesSheet As Worksheet
    Dim tableSheet As Worksheet
    Dim i As Integer

    Set citiesSheet = Sheets("Cities")

    Set tableSheet = Sheets.Add(, citiesSheet)
    tableSheet.Name = "Table"
    tableSheet.Range("A1").Value = "City"
    tableSheet.Range("A1").Font.Bold = True
    tableSheet.Range("B1").Value = "State"
    tableSheet.Range("B1").Font.Bold = True

    i = 2
    citiesSheet.Activate
    Range("A1").Activate
    Do Until IsEmpty(ActiveCell.Value)
        tableSheet.Cells(i, 1) = ActiveCell.Value
        ActiveCell.Offset(1, 0).Activate
        tableSheet.Cells(i, 2) = ActiveCell.Value
        ActiveCell.Offset(2, 0).Activate
        i = i + 1
    Loop
End Sub
```

Alternative exclusively using indices

```
Sub Reorganize_UsingIndices()
    Dim citiesSheet As Worksheet
    Dim tableSheet As Worksheet
    Dim i As Integer

    Set citiesSheet = Sheets("Cities")

    Set tableSheet = Sheets.Add(, citiesSheet)
    tableSheet.Name = "Table"
    tableSheet.Range("A1").Value = "City"
    tableSheet.Range("A1").Font.Bold = True
    tableSheet.Range("B1").Value = "State"
    tableSheet.Range("B1").Font.Bold = True

    i = 0
    Do Until IsEmpty(citiesSheet.Cells(i * 3 + 1, 1).Value)
        tableSheet.Cells(i + 2, 1).Value = citiesSheet.Cells(i * 3 + 1, 1).Value
        tableSheet.Cells(i + 2, 2).Value = citiesSheet.Cells(i * 3 + 2, 1).Value
        i = i + 1
    Loop

    With tableSheet.Sort
        .SortFields.Clear
        .SortFields.Add2 Key:=Range(tableSheet.Range("B2"),
tableSheet.Range("B2").End(xlDown)) _
            , SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
        .SetRange Range(tableSheet.Range("A1"), tableSheet.Range("B2").End(xlDown))
        .Header = xlYes
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With
End Sub
```

Alternative Using Navigation

```
Option Explicit

Sub Reorganize_UsingNavigation()
    Dim citiesSheet As Worksheet
    Dim tableSheet As Worksheet
    Dim i As Integer
    Dim city As Variant
    Dim state As Variant

    Set citiesSheet = Sheets("Cities")

    Set tableSheet = Sheets.Add(, citiesSheet)
    tableSheet.Name = "Table"
    tableSheet.Range("A1").value = "City"
    tableSheet.Range("A1").Font.Bold = True
    tableSheet.Range("B1").value = "State"
    tableSheet.Range("B1").Font.Bold = True

    tableSheet.Range("A2").Activate
    citiesSheet.Activate
    citiesSheet.Range("A1").Activate
    Do Until IsEmpty(ActiveCell.value)
        city = ActiveCell.value
        state = ActiveCell.Offset(1, 0).value
        ActiveCell.Offset(3, 0).Activate

        tableSheet.Activate
        ActiveCell.value = city
        ActiveCell.Offset(0, 1).value = state
        ActiveCell.Offset(1, 0).Activate

        citiesSheet.Activate
    Loop
End Sub
```

Loops: FillIn

```
Option Explicit

Sub FillIn()
    Dim count As Integer
    Dim i As Integer
    count = ActiveCell.value
    For i = 1 To count
        ActiveCell.Offset(i - 1, 0).value = i
    Next i
End Sub
```

Strings and Arrays

```
Option Explicit

Sub Parse()
    Dim a() As String
    Dim i As Integer
    a = Split(ActiveCell.value, ",")
    For i = LBound(a) To UBound(a)
        ActiveCell.Offset(1, i).value = Trim(a(i))
    Next i
End Sub
```

Nested Loops: TimesTable

```
Sub TimesTable()
    Dim a As Variant
    Dim x As Integer
    Dim y As Integer
    Dim i As Integer, j As Integer

    a = Split(ActiveCell.value, "x")
    x = a(0)
    y = a(1)
    For i = 1 To y
        For j = 1 To x
            ActiveCell.Offset(i - 1, j - 1).value = i * j
        Next j
    Next i
End Sub
```

Nested Loops and Excel Navigation: TimesTable

```
Sub TimesTable()
    Dim r As Range
    Dim x As Integer, y As Integer
    Dim i As Integer, j As Integer

    Set r = Range(ActiveCell, ActiveCell.End(xlToRight))
    x = r.Count
    Set r = Range(ActiveCell, ActiveCell.End(xlDown))
    y = r.Count
    For i = 1 To y - 1
        For j = 1 To x - 1
            ActiveCell.Offset(i, j) = ActiveCell.Offset(i, 0) * ActiveCell.Offset(0, j)
        Next j
    Next i
End Sub
```

Navigation Properties

- Range
- Cells
- Value
- Activate
- ActiveCell
- ActiveSheet
- Offset
- Select
- CurrentRegion
- End(xlDown), End(xlUp), End(xlToLeft), End(xlToRight)