

	A	B	C	D	E
1	String		smallint	Integer	Date
2	1	A	32768	2147483647	=DATE(RANDBETWEEN(1970,2025),RANDBETWEEN(1,12),RANDBETWEEN(1,28))
3	2	AB	=C2*-1	-2147483648	
4	3	ABC	0	0	
5	4	ABCD			
6	5	ABCDE			
7	6	ABCDEF			
8	7	ABCDEFG			
9	8	ABCDEFGH			
10	9	ABCDEFGHI			

	F	G
1	Float	Decimal
2	562949953421312	999999.99
3	-562949953421312	=G2*-1
4	0.562949953421312	0
5		
6		
7		
8		
9		
10		

	H
1	Timestamp
2	=DATE(RANDBETWEEN(1970,2025),RANDBETWEEN(1,12),RANDBETWEEN(1,28))+TIME(RANDBETWEEN(0,23),RANDBETWEEN(0,59),RANDBETWEEN(0,59))
3	
4	
5	
6	
7	
8	
9	
10	

TestData_prep_COV_ENDORSMNT-2.vb

```
1 Private Sub Refresh_data()
2     'Variable declaration
3     Dim wBook As Workbook
4     Dim wUPDlist As Worksheet
5     Dim wTabStructure As Worksheet
6     Dim wTestData As Worksheet
7     Dim wInsSubset As Worksheet
8
9
10    Dim rUpdKeys As Range
11    Dim rStructure As Range
12    Dim rSrchSrc As Range
13    Dim rSrchTgt As Range
14
15    Dim iRowCounter As Integer
16    Dim iCellRef As Integer
17    Dim sResult As String
18    Dim iDebug As Integer
19
20
21    'Setting values
22    Set wBook = ThisWorkbook
23    With wBook
24        Set wUPDlist = .Worksheets("Update_flag")
25        Set wTabStructure = .Worksheets("Table_structure")
26        'Set wTestData = .Worksheets("Test_run")
27        Set wTestData = .Worksheets("Test_data")
28        Set wInsSubset = .Worksheets("Ins_file_subset_for_test")
29    End With
30
31    Set rUpdKeys = wUPDlist.Range("Update_keys")
32    Set rStructure = wTabStructure.Range("Structure")
33    iRowCounter = 2
34    iDebug = 0
35
36    'Refresh file before placing test data
37    wInsSubset.UsedRange.Copy
38    wTestData.Activate
39    wTestData.Cells(1, 1).Select
40    wTestData.Paste
41
42
43    wTestData.UsedRange.Interior.ColorIndex = xlNone ' Reset coloring
44 End Sub
45
46
47 Sub generate_values()
48
49     ' data_prep Macro
50
51
52
53     Dim sDatatype As String      ' Data type of the column defined in target table.
54     This should be formated properly in Excel to begin with
55                                         ' the script can be extended to get it from a tabel
56                                         ' script !!!!'
57
58     Dim iLength As Integer       ' length of the column defined in target table.
59     Dim iQuotient As Integer
60     max_length
61
62     Dim iRemainder As Integer   ' This is used to generate character string with
63     max_length
64
65     Dim iDivisor As Integer     ' This is used to generate character string with
66                                         ' Divisor is selected as 10 for simplicity
```

```

59      Dim iLoopCounter As Integer      ' Loop counter used to generate the Char string with
60      max length
61      Dim sResult As String          ' Stores the resulting value
62      Dim found As Range            ' Range variable used to lookup
63      Dim iStringLength             ' Dim cell As Range
64      Dim iPosition As Integer       ' This variable is used while generating Test string
65      Dim iDebug As Integer
66
67      Dim objRegex As Object         ' For Regex
68
69
70      Dim StuctureRange As Range    ' The range of input to the script. This should
71      include last three empty colun for result too
72      Set StuctureRange = Range("Structure")
73      iDivisor = 10
74      iPosition = 3 ' This is the position of the length of the table column is
75      specified in the selected range
76      iDebug = 0
77
78
79      ' Button = MsgBox("Do you want to Clear the data before generating? The test data
80      From column L to O will be cleared if YES is chosen", vbYesNo)
81
82      ' If Button = vbYes Then
83          'Clear the data before refreshing
84          Sheets("Table_structure").Range(StuctureRange.Cells(1, 11),
85          StuctureRange.Cells(StuctureRange.Rows, StuctureRange.Columns)).ClearContents
86      'End If
87
88
89      ' This loop scans all the column name to generate the values
90      For j = 1 To StuctureRange.Rows.Count
91
92          ' code to generate char strings in following format
93
94          ABCDEFGHI1ABCDEFGHI2ABCDEFGHI3ABCDEFGHI4ABCDEFGHI5ABCDEFGHI6ABCDEFGHI7ABCDEFGHI
95          I8ABCDEFGHI9ABCDEFGHI10ABCDEFGHI11
96          ' When the length is less than 99 pattern will be ABCDEFGHIx and more than 99
97          ABCDEFGHxx
98          ' This goes on for each digit added to number
99          If (StuctureRange(j, 2).Value = "Varchar" Or StuctureRange(j, 2).Value = "Char")
100         ) Then
101
102             iRemainder = StuctureRange(j, iPosition).Value Mod iDivisor
103             iLoopCounter = Int(StuctureRange(j, iPosition).Value / iDivisor)
104             ' This logic is used for length up to 99
105
106             ' This logic is used for length more than 99
107             If iLoopCounter > 0 Then
108                 For i = 1 To iLoopCounter
109
110                     iStringLength = 9 - Int(Application.WorksheetFunction.Log10(i) / 1)
111                     Set found = Sheets("Data").Columns(1).Cells.Find(What:=iStringLength,
112                     After:=[A1], LookAt:=xlWhole, SearchOrder:=xlByRows,
113                                         SearchDirection:=
114                                         xlPrevious)
115
116                     If found Is Nothing Then
117                         sResult = "Failed"
118                     Else

```

TestData_prep_COV_ENDORSMNT-2.vb

```
110             sResult = sResult & Sheets("Data").Cells(found.Row, 2).Value & i
111
112         End If
113     Next i
114 End If
115
116 ' this section add the last piece of character string, which
117 Set found = Sheets("Data").Columns(1).Cells.Find(What:=iRemainder, After:=[A1], LookAt:=xlWhole, SearchOrder:=xlByRows, _
118                                         SearchDirection:=xlPrevious)
119     If found Is Nothing Then
120         sResult = sResult
121     Else
122         sResult = sResult & Sheets("Data").Cells(found.Row, 2).Value
123
124     End If
125
126     StructureRange(j, StructureRange.Columns.Count - 3).Select
127     ActiveCell.Value = sResult
128 'Code to generate Varchar/Char ends here
129 Else
130 'Code to generate Non string datatypes goes here
131     Set found = Sheets("Data").Rows(1).Cells.Find(What:=StructureRange(j, 2) .Value, After:=[A1], LookAt:=xlWhole, SearchOrder:=xlByColumns, _
132                                         SearchDirection:=xlPrevious)
133     If found Is Nothing Then
134         sResult = sResult
135     Else
136         sResult = Sheets("Data").Cells(2, found.Column).Value
137
138     End If
139     StructureRange(j, StructureRange.Columns.Count - 3).Select
140     ActiveCell.Value = sResult
141
142     For k = 1 To 2
143         sResult = Sheets("Data").Cells(2 + k, found.Column).Value
144         If sResult <> "" Then
145
146             StructureRange(j, StructureRange.Columns.Count - 3 + k).Select
147             ActiveCell.Value = sResult
148         End If
149     Next k
150
151     StructureRange(j, StructureRange.Columns.Count).Select
152     ActiveCell.Value = "X"
153
154 'Code to generate Non string datatypes ends here
155 End If
156
157     iCounter = iCounter + 1
158     sResult = ""           'clear the value generated for next iteration
159 Next j                         'Loop over the named range" StructureRange" end
here
160
161 Button = MsgBox("Execution completed. Do you want to switch to the sheet where
data was added?", vbYesNo)
162
163 If Button = vbYes Then
164     StructureRange.ParentActivate
165 End If
166
```

```

167 End Sub
168
169
170
171
172 Sub Update_test_data()
173     'Variable declaration
174     Dim wBook As Workbook
175     Dim wUPDlist As Worksheet
176     Dim wTabStructure As Worksheet
177     Dim wTestData As Worksheet
178     Dim wInsSubset As Worksheet
179
180
181     Dim rUpdKeys As Range
182     Dim rStructure As Range
183     Dim rSrchSrc As Range
184     Dim rSrchTgt As Range
185
186     Dim iRowCounter As Integer
187     Dim iCellRef As Integer
188     Dim sResult As String
189     Dim iDebug As Integer
190
191     'Setting values
192     Set wBook = ThisWorkbook
193     With wBook
194         Set wUPDlist = .Worksheets("Update_flag")
195         Set wTabStructure = .Worksheets("Table_structure")
196         'Set wTestData = .Worksheets("Test_run")
197         Set wTestData = .Worksheets("Test_data")
198         Set wInsSubset = .Worksheets("Ins_file_subset_for_test")
199     End With
200
201     Set rUpdKeys = wUPDlist.Range("Update_keys")
202     Set rStructure = wTabStructure.Range("Structure")
203     iRowCounter = 2
204     iDebug = 0
205
206     Call Refresh_data
207
208     'Refresh file before placing test data
209     'wInsSubset.UsedRange.Copy
210     'wTestData.Paste
211
212     'wTestData.UsedRange.Interior.ColorIndex = xlNone ' Reset coloring
213
214
215     ' Looping thru the update keys - Boundary testing
216     For Each keys In rUpdKeys
217
218         If iDebug = 1 Then
219             MsgBox keys
220         End If
221
222         ' Search the key in the Table structure tab. Here the coulmn name in the
223         ' table and file should be matched up.
224         ' Also the values should be used for testing should be present here
225         Set rSrchSrc = wTabStructure.Cells.Find(What:=keys, After:=[A1], LookAt:=
xlWhole, SearchOrder:=xlByRows, SearchDirection:=xlNext)
         'If search fail a message is displayed and the data update continues for
         'the rest of the columns

```

TestData_prep_COV_ENDORSMNT-2.vb

```
226      If rSrchSrc Is Nothing Then
227          sResult = "Failed"
228          MsgBox "Column name mismatch occurred. Search for " & Key & "
229              failed in worksheet named " & wTabStructure.Name
230
231      Else
232          ' This counter advances the row, making sure that only one value is
233          ' updated in a column
234
235          iCellRef = rSrchSrc.Column + 6
236          sResult = wTabStructure.Cells(rSrchSrc.Row, iCellRef).Value
237
238          If iDebug = 1 Then
239              MsgBox "Found column name " & wTabStructure.Cells(rSrchSrc.Row,
240                  iCellRef).Value
241              MsgBox "Found column name sResult=" & sResult
242          End If
243
244          ' This second search is to find where to place the test data
245          Set rSrchTgt = wTestData.Cells.Find(What:=wTabStructure.Cells(
246              rSrchSrc.Row, iCellRef).Value, After:=[A1], LookAt:=xlWhole,
247              SearchOrder:=xlByColumns, _
248              SearchDirection:=xlNext)
249
250          If rSrchTgt Is Nothing Then
251
252              'sResult = "Failed"
253              MsgBox "Column name mismatch occurred. Search for " & Key & "
254                  failed in worksheet named " & wTestData.Name
255
256          Else
257              For iCellRef = 10 To 12
258                  sResult = wTabStructure.Cells(rSrchSrc.Row,
259                      rSrchSrc.Column + iCellRef).Value
260                  If sResult <> "" Then
261                      With wTestData.Cells(iRowCounter, rSrchTgt.Column),
262                          .Value = sResult
263                          .Interior.ColorIndex = 6
264                      End With
265
266                      If iDebug = 1 Then
267                          MsgBox "Placing data " & sResult & " in cell " &
268                              iRowCounter & "," & rSrchTgt.Column
269                      End If
270                      iRowCounter = iRowCounter + 1
271                  End If
272                  'wTestData.Cells(iRowCounter, 13).Value = sResult
273                  'wTabStructure.Cells(i, 15).Value =
274                  wTabStructure.Cells(i, 12).Value
275
276                  Next iCellRef
277                  'wTestData.Cells(iRowCounter, rSrchTgt.Column).Value = ""
278                  'iRowCounter = iRowCounter + 1
279              End If
280          End If
281
282          sResult = ""           ' Clear the result value so that new value is picked up
283          ' in next loop
284
285      Next keys
286
287
288      ' Looping thru the update keys - Negative testing
289      For Each keys In rUpdKeys
```

```

278
279     If iDebug = 1 Then
280         MsgBox keys
281     End If
282
283     ' Search the key in the Table structure tab. Here the column name in the
284     ' table and file should be matched up.
285     Set rSrchSrc = wTabStructure.Cells.Find(What:=keys, After:=[A1], LookAt:=
286     xlWhole, SearchOrder:=xlByRows, SearchDirection:=xlNext)
287     'If search fail a message is displayed and the data update continues for
288     ' the rest of the columns
289     If rSrchSrc Is Nothing Then
290         sResult = "Failed"
291         MsgBox "Column name mismatch occurred. Search for " & Key &
292         failed in worksheet named " & wTabStructure.Name
293
294     Else
295         ' This counter advances the row, making sure that only one value is
296         updated in a column
297
298         iCellRef = rSrchSrc.Column + 6
299         sResult = wTabStructure.Cells(rSrchSrc.Row, iCellRef).Value
300
301         If iDebug = 1 Then
302             MsgBox "Found column name " & wTabStructure.Cells(rSrchSrc.Row,
303             iCellRef).Value
304             MsgBox "Found column name sResult=" & sResult
305             End If
306
307         ' This second search is to find where to place the test data
308         Set rSrchTgt = wTestData.Cells.Find(What:=wTabStructure.Cells(
309             rSrchSrc.Row, iCellRef).Value, After:=[A1], LookAt:=xlWhole,
310             SearchOrder:=xlByColumns, _
311             SearchDirection:=xlNext)
312
313         If rSrchTgt Is Nothing Then
314             'sResult = "Failed"
315             MsgBox "Column name mismatch occurred. Search for " & Key &
316             failed in worksheet named " & wTestData.Name
317
318         Else
319             iCellRef = 13
320             sResult = wTabStructure.Cells(rSrchSrc.Row, rSrchSrc.Column +
321             iCellRef).Value
322             If sResult <> "" Then
323                 With wTestData.Cells(iRowCounter, rSrchTgt.Column)
324                     .Value = sResult
325                     .Interior.ColorIndex = 6
326                 End With
327
328             If iDebug = 1 Then
329                 MsgBox "Placing data " & sResult & " in cell " &
330                 iRowCounter & "," & rSrchTgt.Column
331             End If
332             iRowCounter = iRowCounter + 1
333         End If
334             'wTestData.Cells(iRowCounter, 13).Value = sResult
335             'wTabStructure.Cells(i, 15).Value =
336             wTabStructure.Cells(i, 12).Value

```

TestData_prep_COV_ENDORSMNT-2.vb

```
328             'wTestData.Cells(iRowCounter, rSrchTgt.Column).Value = ""
329             'iRowCounter = iRowCounter + 1
330         End If
331     End If
332
333     sResult = ""           ' Clear the result value so that new value is picked up
334     in next loop
335 Next keys
336
337 ' Looping thru the update keys _ null check
338 For Each keys In rUpdKeys
339
340     If iDebug = 1 Then
341         MsgBox keys
342     End If
343
344     ' Search the key in the Table structure tab. Here the coulmn name in the
345     table and file should be matched up.
346     ' Also the values should be used for testing should be present here
347     Set rSrchSrc = wTabStructure.Cells.Find(What:="keys", After:=[A1], LookAt:=
348     xlWhole, SearchOrder:=xlByRows, SearchDirection:=xlNext)
349     'If search fail a message is displayed and the data update continues for
350     the rest of the columns
351     If rSrchSrc Is Nothing Then
352         sResult = "Failed"
353         MsgBox "Column name mismatch occured. Search for " & Key & "
354         failed in worksheet named " & wTabStructure.Name
355     Else
356         ' This counter advances the row, making sure that only one value is
357         updated in a column
358
359         iCellRef = rSrchSrc.Column + 6
360         sResult = wTabStructure.Cells(rSrchSrc.Row, iCellRef).Value
361
362         If iDebug = 1 Then
363             MsgBox "Found column name " & wTabStructure.Cells(rSrchSrc.Row,
364             iCellRef).Value
365             MsgBox "Found column name sResult=" & sResult
366         End If
367
368         ' This second search is to find where to place the test data
369         Set rSrchTgt = wTestData.Cells.Find(What:=wTabStructure.Cells(
370             rSrchSrc.Row, iCellRef).Value, After:=[A1], LookAt:=xlWhole,
371             SearchOrder:=xlByColumns, _
372             SearchDirection:=xlNext)
373
374         If rSrchTgt Is Nothing Then
375             sResult = "Failed"
376             MsgBox "Column name mismatch occured. Search for " & Key & "
377             failed in worksheet named " & wTestData.Name
378
379         Else
380             With wTestData.Cells(iRowCounter, rSrchTgt.Column)
381                 .Value = ""
382                 .Interior.ColorIndex = 6
383             End With
384             iRowCounter = iRowCounter + 1
385         End If
386     End If
```

```
380
381      sResult = ""           ' Clear the result value so that new value is picked up
382      in next loop
383      Next keys
384
385
386      'Button As VbMsgBoxResult
387
388      Button = MsgBox("Execution completed. Do you want to switch to the sheet where
389      data was added?", vbYesNo)
390
391      If Button = vbYes Then
392          wTestData.Activate
393
394      End If
395
396  End Sub
397
```