

엑셀 Live 24 강 - 보충자료

엑셀 VBA 를 활용한 관계형 DB 구축 매크로 명령문

2020.11. 28



목차

1. 시트의 DB 를 배열로 반환 (GET_DB 명령문)	3
2. 시트의 특정 ID 필드 반환 (GET_RECORDS 명령문)	3
3. DB 에 새로운 레코드 추가 (INSERT_RECORD 명령문).....	3
4. 기존 데이터 업데이트 (UPDATE_RECORD 명령문)	3
5. 데이터 삭제 (DELETE_RECORD 명령문)	3
6. 두개의 서로 다른 DB 연결 (CONNECT_DB 명령문)	4
7. 필터링 된 DB 를 배열로 반환 (FILTERED_DB 명령문).....	4
8. 배열을 시트 위 범위로 반환 (ARRAYTORNG 명령문)	14

1. 시트의 DB 를 배열로 반환 (Get_DB 명령문)

인수설명 :

Get_DB (시트, [ID 포함], [머릿글포함])

사용예제 :

Dim DB **as** Variant

DB = Get_DB (ActiveWorkBook.WorkSheets("시트명"))

2. 시트의 특정 ID 필드를 반환 (Get_Records 명령문)

인수설명 :

Get_Records (시트 , ID, "필드 1,필드 2,필드 3,...")

사용예제 :

Dim Data **as** Variant

Data = Get_Records (ActiveWorkBook.WorkSheets("시트명"), 1, "이름, 나이, 성별")

3. DB 에 새로운 레코드 추가 (Insert_Record 명령문)

인수설명 :

Insert_Record 시트, 값 1, 값 2, 값 3, ...

'DB 의 필드 순서와 동일하게 입력합니다, ID(순번)은 자동으로 지정됩니다.

4. 기존 데이터 업데이트 (Update_Record 명령문)

인수설명 :

Update_Record 시트, ID, 값 1, 값 2, 값 3 ...

'DB 의 필드 순서와 동일하게 입력합니다.

5. 데이터 삭제 (Delete_Record 명령문)

인수설명 :

Delete_Record 시트, ID

'시트의 첫번째 열(ID)를 참조하여 레코드를 삭제합니다. ID 는 반드시 첫번째 열에 있어야 합니다.

6. 두개의 서로 다른 DB 연결 (Connect_DB 명령문)

인수설명 :

Connect_DB (DB, 참조 ID 열번호, 참조시트, 필드명, [머릿글포함])

사용예제 :

Dim DB **as** Variant

DB = Get_DB (시트 1)

DB = Connect_DB (DB, 2, 시트 2, "이름, 나이, 성별")

7. 필터링 된 DB 를 배열로 반환 (Filtered_DB 명령문)

인수설명 :

Filtered_DB (DB, 조건, 열번호, [정확히일치]) '정확히일치 기본값은 False 입니다.

사용예제 :

Dim DB **as** Variant

DB = Get_DB (시트 1)

DB = Filtered_DB (DB, ">=30", 2) ' 두번째 열에서 값이 30 이상인 데이터만 필터링합니다.

8. DB 함수 전체 명령문 코드

```
Option Explicit
Option Compare Text
'#####
' 특정 워크시트에서 앞으로 추가해야 할 최대 ID 번호 리턴 (시트 DB 우측 첫번째 머릿글)
' i = Get_MaxID(Sheet1)
'#####
Function Get_MaxID(WS As Worksheet) As Long
With WS
    Get_MaxID = .Cells(1, .Columns.Count).End(xlToLeft).Value
    .Cells(1, .Columns.Count).End(xlToLeft).Value = .Cells(1, .Columns.Count).End(xlToLeft).Value + 1
End With
End Function
'#####
' 워크시트에 새로운 데이터를 추가해야 할 열번호 반환
' i = Get_InsertRow(Sheet1)
'#####
Function Get_InsertRow(WS As Worksheet) As Long
With WS: Get_InsertRow = .Cells(.Rows.Count, 1).End(xlUp).Row + 1: End With
End Function
```

```

#####
' 시트의 열 개수 반환 (이번 예제파일에서만 사용)
' i = Get_ColumnCnt(Sheet1)
#####
Function Get_ColumnCnt(WS As Worksheet, Optional Offset As Long = -1) As Long
With WS: Get_ColumnCnt = .Cells(1, .Columns.Count).End(xlToLeft).Column + Offset: End With
End Function

#####
' 시트에서 특정 ID 의 행 번호 반환 (-> 해당 행 번호 데이터 업데이트)
' i = get_UpdateRow(Sheet1, ID)
#####
Function get_UpdateRow(WS As Worksheet, ID)
Dim i As Long
Dim cRow As Long
With WS
    cRow = Get_InsertRow(WS) - 1
    For i = 1 To cRow
        If .Cells(i, 1).Value = ID Then get_UpdateRow = i: Exit For
    Next
End With
End Function

#####
' 특정 시트의 DB 정보를 배열로 반환 (이번 예제파일에서만 사용)
' Array = Get_DB(Sheet1)
#####
Function Get_DB(WS As Worksheet, Optional NoID As Boolean = False, Optional IncludeHeader As Boolean = False) As Variant

Dim cRow As Long
Dim cCol As Long
Dim offCol As Long

If NoID = False Then offCol = -1

With WS
    cRow = Get_InsertRow(WS) - 1
    cCol = Get_ColumnCnt(WS, offCol)
    Get_DB = .Range(.Cells(2 + Sgn(IncludeHeader), 1), .Cells(cRow, cCol))
End With

End Function

#####

```

```

'특정 시트에서 지정한 ID 의 필드 값 반환 (이번 예제파일 전용)
' Value = Get_Records(Sheet1, ID, "필드명")
#####
Function Get_Records(WS As Worksheet, ID, Fields)

Dim cRow As Long: Dim cCol As Long
Dim vFields As Variant: Dim vField As Variant
Dim vFieldNo As Variant
Dim i As Long: Dim j As Long

cRow = Get_InsertRow(WS) - 1
cCol = Get_ColumnCnt(WS)

If InStr(1, Fields, ",") > 0 Then vFields = Split(Fields, ",") Else vFields = Array(Fields)
ReDim vFieldNo(0 To UBound(vFields))

With WS
    For Each vField In vFields
        For i = 1 To cCol
            If .Cells(1, i).Value = Trim(vField) Then vFieldNo(j) = i: j = j + 1
        Next
    Next

    For i = 2 To cRow
        If .Cells(i, 1).Value = ID Then
            For j = 0 To UBound(vFieldNo)
                vFieldNo(j) = .Cells(i, vFieldNo(j))
            Next
        Exit For
    End If
Next

Get_Records = vFieldNo

End With

End Function

#####
' 시트에 새로운 레코드 추가 (반드시 첫번째 값은 ID, 나머지 값 순서대로 입력)
' Insert_Record Sheet1, ID, 필드 1, 필드 2, 필드 3, ..
#####
Sub Insert_Record(WS As Worksheet, ParamArray vaParamArr() As Variant)

```

```

Dim cID As Long
Dim cRow As Long
Dim vaArr As Variant: Dim i As Long: i = 2

With WS
    cRow = Get_InsertRow(WS)
    If InStr(1, .Cells(1, 1).Value, "ID") > 0 Then
        cID = Get_MaxID(WS)
        .Cells(cRow, 1).Value = cID
        For Each vaArr In vaParamArr
            .Cells(cRow, i).Value = vaArr
            i = i + 1
        Next
    Else
        For Each vaArr In vaParamArr
            .Cells(cRow, i - 1).Value = vaArr
            i = i + 1
        Next
    End If
End With

End Sub

'#####
' 시트에서 ID 를 갖는 레코드의 모든 값 업데이트 (반드시 첫번째 값은 ID 여야 하며, 나머지 값을 순서대로 입력)
' Update_Record Sheet1, ID, 필드 1, 필드 2, 필드 3, ...
'#####
Sub Update_Record(WS As Worksheet, ParamArray vaParamArr() As Variant)

    Dim cRow As Long
    Dim i As Long
    Dim ID As Variant

    If IsNumeric(vaParamArr(0)) = True Then ID = CLng(vaParamArr(0)) Else ID = vaParamArr(0)

    With WS
        cRow = get_UpdateRow(WS, ID)

        For i = 1 To UBound(vaParamArr)
            If Not IsMissing(vaParamArr(i)) Then .Cells(cRow, i + 1).Value = vaParamArr(i)
        Next
    End With

End Sub

```

```

#####
' 시트에서 ID 를 갖는 레코드 삭제
' Delete_Record Sheet1, ID
#####
Sub Delete_Record(WS As Worksheet, ID)

Dim cRow As Long

If IsNumeric(ID) = True Then ID = CLng(ID)

With WS
    cRow = get_UpdateRow(WS, ID)
    .Cells(cRow, 1).EntireRow.Delete
End With

End Sub

#####
' 배열의 외부 ID 키 필드를 본 시트 DB 와 연결하여 해당 외부 ID 키의 연관된 값을 배열로 반환
' Array = Connect_DB(Get_DB(Sheet1),2,Sheet2, "필드 1, 필드 2, 필드 3")
#####
Function Connect_DB(DB As Variant, ForeignID_Fields As Variant, FromWS As Worksheet, Fields As String,
Optional IncludeHeader As Boolean = False)

Dim cRow As Long: Dim cCol As Long
Dim vForeignID_Fields As Variant: Dim vForeignID_Field As Variant
Dim ForeignID As Variant
Dim vFields As Variant: Dim vField As Variant
Dim vID As Variant: Dim vFieldNo As Variant
Dim Dict As Object
Dim i As Long: Dim j As Long
Dim AddCols As Long

cRow = UBound(DB, 1)
cCol = UBound(DB, 2)
If InStr(1, Fields, ",") > 1 Then
    AddCols = Len(Fields) - Len(Replace(Fields, ",", "")) + 1
    vFields = Split(Fields, ",")
Else
    AddCols = 1
    vFields = Array(Fields)
End If

ReDim Preserve DB(1 To cRow, 1 To cCol + AddCols)

```



```

Set Dict = Get_Dict(FromWS)
vID = Dict("ID")

ReDim vFieldNo(0 To UBound(vFields))

For Each vField In vFields
    For i = 1 To UBound(vID)
        If vID(i) = Trim(vField) Then vFieldNo(j) = i: j = j + 1
    Next
Next

If InStr(1, ForeignID_Fields, ",") > 0 Then vForeignID_Fields = Split(ForeignID_Fields, ",") Else
vForeignID_Fields = Array(ForeignID_Fields)

For Each vForeignID_Field In vForeignID_Fields
    For i = 1 To cRow
        If IncludeHeader = True And i = 1 Then ForeignID = "ID" Else ForeignID = DB(i, Trim(vForeignID_Field))
        If Dict.Exists(ForeignID) Then
            For j = 1 To AddCols
                DB(i, cCol + j) = Dict(ForeignID)(vFieldNo(j - 1))
            Next
        End If
    Next
Next

Connect_DB = DB

End Function
'#####
' 특정 배열에서 Value 를 포함하는 레코드만 찾아 다시 배열로 반환
' Array = Filtered_DB(Array, "검색값", False)
'#####
Function Filtered_DB(DB, Value, Optional FilterCol, Optional ExactMatch As Boolean = False) As Variant

Dim cRow As Long
Dim cCol As Long
Dim vArr As Variant: Dim s As String: Dim filterArr As Variant: Dim Cols As Variant: Dim Col As Variant: Dim
Colcnt As Long
Dim isDateVal As Boolean
Dim vReturn As Variant: Dim vResult As Variant
Dim Dict As Object: Dim dictKey As Variant
Dim i As Long: Dim j As Long
Dim Operator As String

```

```

Set Dict = CreateObject("Scripting.Dictionary")

If Value <> "" Then
    cRow = UBound(DB, 1)
    cCol = UBound(DB, 2)
    ReDim vArr(1 To cRow)
    For i = 1 To cRow
        s = ""
        For j = 1 To cCol
            s = s & DB(i, j) & "|" ^ "
        Next
        vArr(i) = s
    Next

    If IsMissing(FilterCol) Then
        filterArr = vArr
    Else
        Cols = Split(FilterCol, ",")
        ReDim filterArr(1 To cRow)
        For i = 1 To cRow
            s = ""
            For Each Col In Cols
                s = s & DB(i, Trim(Col)) & "|" ^ "
            Next
            filterArr(i) = s
        Next
    End If

    If left(Value, 2) = ">=" Or left(Value, 2) = "<=" Or left(Value, 2) = ">" Or left(Value, 2) = "<" Then
        Operator = left(Value, 2)
        If IsDate(Right(Value, Len(Value) - 2)) Then isDateVal = True
    ElseIf left(Value, 1) = ">" Or left(Value, 1) = "<" Then
        Operator = left(Value, 1)
        If IsDate(Right(Value, Len(Value) - 1)) Then isDateVal = True
    Else: End If

    If Operator <> "" Then
        If isDateVal = False Then
            Select Case Operator
                Case ">"
                    For i = 1 To cRow
                        If CDBl(left(filterArr(i), Len(filterArr(i)) - 2)) > CDBl(Right(Value, Len(Value) - 1)) Then: vArr(i) =
left(vArr(i), Len(vArr(i)) - 2): vReturn = Split(vArr(i), "|"): Dict.Add i, vReturn
                    Next
                Case "<"

```

```

        For i = 1 To cRow
            If CDBl(left(filterArr(i), Len(filterArr(i)) - 2)) < CDBl(Right(Value, Len(Value) - 1)) Then: vArr(i) =
left(vArr(i), Len(vArr(i)) - 2): vReturn = Split(vArr(i), "|^"): Dict.Add i, vReturn
        Next
    Case ">=", "=>"
        For i = 1 To cRow
            If CDBl(left(filterArr(i), Len(filterArr(i)) - 2)) >= CDBl(Right(Value, Len(Value) - 2)) Then: vArr(i) =
left(vArr(i), Len(vArr(i)) - 2): vReturn = Split(vArr(i), "|^"): Dict.Add i, vReturn
        Next
    Case "<=", "=<"
        For i = 1 To cRow
            If CDBl(left(filterArr(i), Len(filterArr(i)) - 2)) <= CDBl(Right(Value, Len(Value) - 2)) Then: vArr(i) =
left(vArr(i), Len(vArr(i)) - 2): vReturn = Split(vArr(i), "|^"): Dict.Add i, vReturn
        Next
    End Select
Else
    Select Case Operator
    Case ">"
        For i = 1 To cRow
            If CDate(left(filterArr(i), Len(filterArr(i)) - 2)) > CDate(Right(Value, Len(Value) - 1)) Then: vArr(i)
= left(vArr(i), Len(vArr(i)) - 2): vReturn = Split(vArr(i), "|^"): Dict.Add i, vReturn
        Next
    Case "<"
        For i = 1 To cRow
            If CDate(left(filterArr(i), Len(filterArr(i)) - 2)) < CDate(Right(Value, Len(Value) - 1)) Then: vArr(i)
= left(vArr(i), Len(vArr(i)) - 2): vReturn = Split(vArr(i), "|^"): Dict.Add i, vReturn
        Next
    Case ">=", "=>"
        For i = 1 To cRow
            If CDate(left(filterArr(i), Len(filterArr(i)) - 2)) >= CDate(Right(Value, Len(Value) - 2)) Then:
vArr(i) = left(vArr(i), Len(vArr(i)) - 2): vReturn = Split(vArr(i), "|^"): Dict.Add i, vReturn
        Next
    Case "<=", "=<"
        For i = 1 To cRow
            If CDate(left(filterArr(i), Len(filterArr(i)) - 2)) <= CDate(Right(Value, Len(Value) - 2)) Then:
vArr(i) = left(vArr(i), Len(vArr(i)) - 2): vReturn = Split(vArr(i), "|^"): Dict.Add i, vReturn
        Next
    End Select
End If
Else
    If ExactMatch = False Then
        For i = 1 To cRow
            If filterArr(i) Like "*" & Value & "*" Then
                vArr(i) = left(vArr(i), Len(vArr(i)) - 2)
                vReturn = Split(vArr(i), "|^")
            End If
        Next
    End If
End If

```

```

        Dict.Add i, vReturn
    End If
Next
Else
    For i = 1 To cRow
        If filterArr(i) Like Value & "|" ^ " Then
            vArr(i) = left(vArr(i), Len(vArr(i)) - 2)
            vReturn = Split(vArr(i), "|" ^ " )
            Dict.Add i, vReturn
        End If
    Next
End If
End If

If Dict.Count > 0 Then
    ReDim vResult(1 To Dict.Count, 1 To cCol)
    i = 1
    For Each dictKey In Dict.Keys
        For j = 1 To cCol
            vResult(i, j) = Dict(dictKey)(j - 1)
        Next
        i = i + 1
    Next
End If

Filtered_DB = vResult
Else
    Filtered_DB = DB
End If

End Function

'#####
' 특정 시트의 DB 정보를 Dictionary 로 반환 (이번 예제파일에서만 사용)
' Dict = GetDict(Sheet1)
'#####
Function Get_Dict(WS As Worksheet) As Object

Dim cRow As Long: Dim cCol As Long
Dim Dict As Object
Dim vArr As Variant
Dim i As Long: Dim j As Long

Set Dict = CreateObject("Scripting.Dictionary")

```

```
With WS
    cRow = Get_InsertRow(WS) - 1
    cCol = Get_ColumnCnt(WS)

    For i = 1 To cRow
        ReDim vArr(1 To cCol - 1)
        For j = 2 To cCol
            vArr(j - 1) = .Cells(i, j)
        Next
        Dict.Add .Cells(i, 1).Value, vArr
    Next
End With

Set Get_Dict = Dict

End Function
```

9. 배열을 시트 위 범위로 반환 (ArrayToRng 명령문)

배열을 시트 위에 범위로 반환합니다.

```
Sub ArrayToRng(startRng As Range, Arr As Variant)

'#####
'오빠두엑셀 VBA 사용자지정함수 (https://www.oppadu.com)
'▶ ArrayToRng 함수
'▶ 배열을 범위 위로 반환합니다.
'▶ 인수 설명
'_____startRng    : 배열을 반환할 기준 범위(셀) 입니다.
'_____Arr          : 반환할 배열입니다.
'▶ 사용 예제
'Dim v As Variant
'ReDim v(0 to 1)
"v(0) = "a" : v(1) = "b"
'ArrayToRng Sheet1.Range("A1"), v
'#####

On Error GoTo SingleDimension:
startRng.Cells(1, 1).Resize(UBound(Arr, 1) - LBound(Arr, 1) + 1, UBound(Arr, 2) - LBound(Arr, 2) + 1) = Arr

Exit Sub
SingleDimension:
Dim tempArr As Variant: Dim i As Long
ReDim tempArr(LBound(Arr, 1) To UBound(Arr, 1), 1 To 1)
For i = LBound(Arr, 1) To UBound(Arr, 1)
    tempArr(i, 1) = Arr(i)
Next
startRng.Cells(1, 1).Resize(UBound(Arr, 1) - LBound(Arr, 1) + 1, 1) = tempArr

End Sub
```