

Option Compare Database  
Option Explicit

' 比例代表立候補者の惜敗率を求める。  
' 小選挙区に立候補していない候補者の惜敗率は0とする。

Function 惜敗率(候補者番号 As Integer) As Double

Dim DB As Database  
Dim RS1, RS2 As Recordset  
Dim SQL As String  
Dim SENKYOKU As String ' 選挙区  
Dim PREFECTURE As String ' 県  
Dim HYOUSUU As Long ' 得票数  
Dim HYOUSUU\_max As Long ' 最高得票

Set DB = DBEngine.Workspaces(0).Databases(0)

' 指定された候補者が小選挙区に立候補しているならば以下を実行する。

SQL = "SELECT \* FROM 小選挙区得票 WHERE "  
SQL = SQL & " 候補者番号 = " & 候補者番号 & ";"  
Set RS1 = DB.OpenRecordset(SQL, dbOpenSnapshot)  
If Not RS1.EOF Then

' 出馬選挙区、県、得票数を求める。

SENKYOKU = RS1![選挙区]  
PREFECTURE = RS1![県名]  
HYOUSUU = RS1![得票数]

' 出馬選挙区の最高得票数を求める。

SQL = "SELECT Max(得票数) AS 最高得票 FROM 小選挙区得票"  
SQL = SQL & " GROUP BY 選挙区, 県名"  
SQL = SQL & " HAVING 選挙区 = " & SENKYOKU  
SQL = SQL & " AND 県名 = " & PREFECTURE & ";"  
Set RS2 = DB.OpenRecordset(SQL, dbOpenSnapshot)  
HYOUSUU\_max = RS2![最高得票]  
RS2.Close

' 惜敗率を得票数/最高得票数とする。

惜敗率 = HYOUSUU / HYOUSUU\_max

' そうでなければ、惜敗率を 0 とする。

Else  
惜敗率 = 0  
End If

RS1.Close  
DB.Close

End Function

Option Compare Database  
Option Explicit

Sub 比例議席数決定()

```
Dim DB As Database
Dim Block As Recordset ' 比例代表ブロック一覧
Dim RS As Recordset
Dim SQL As String
Dim ttlVote As Long ' ブロック得票合計
Dim ttlMember As Integer ' ブロック定数
Dim total As Integer ' 議席数合計
```

```
Set DB = DBEngine.Workspaces(0).Databases(0)
```

```
' 比例代表ブロック毎に以下の処理を実行する。
Set Block = DB.OpenRecordset("比例代表ブロック", dbOpenTable)
While Not Block.EOF
```

```
' 当該ブロックの票数合計を求める。
SQL = "SELECT ブロック, Sum(得票数) AS 得票合計"
SQL = SQL & " FROM 比例代表得票 INNER JOIN 比例代表県"
SQL = SQL & " ON 比例代表得票.県名 = 比例代表県.県名"
SQL = SQL & " GROUP BY ブロック HAVING ブロック = '" & Block![ブロック] & "';"
Set RS = DB.OpenRecordset(SQL, dbOpenSnapshot)
ttlVote = RS![得票合計]
RS.Close
```

```
' 当該ブロックの定数を求める。
ttlMember = Block![ブロック定数]
```

```
' 当該ブロックの全党派について、議席数の初期値をブロック定数*票数/票数合計とする。
' (小数点以下切り捨て)
```

```
SQL = "INSERT INTO 比例代表議席数 SELECT ブロック, 党派,"
SQL = SQL & " Int(Sum(得票数)*" & ttlMember & "/" & ttlVote & ") AS 議席数,"
SQL = SQL & " Sum(得票数) AS 得票総数"
SQL = SQL & " FROM 比例代表得票 INNER JOIN 比例代表県 ON"
SQL = SQL & " 比例代表得票.県名 = 比例代表県.県名"
SQL = SQL & " GROUP BY ブロック, 党派"
SQL = SQL & " HAVING ブロック = '" & Block![ブロック] & "';"
DB.Execute SQL, dbFailOnError
```

```
' 当該ブロックの議席数合計を求める。
SQL = "SELECT Sum(議席数) AS 議席数合計 FROM 比例代表議席数"
SQL = SQL & " GROUP BY ブロック HAVING ブロック = '" & Block![ブロック] & "';"
Set RS = DB.OpenRecordset(SQL, dbOpenSnapshot)
total = RS![議席数合計]
RS.Close
```

```
' 議席数合計がブロック定数未満である限り以下の処理を繰り返す。
While total < ttlMember
```

```
' 当該ブロック内で票数/(議席数+1)が最も大きい党派の議席数を1増やす。
SQL = "SELECT * FROM 比例代表議席数"
SQL = SQL & " WHERE ブロック = '" & Block![ブロック] & "';"
SQL = SQL & " ORDER BY 得票総数/(議席数+1) DESC;"
Set RS = DB.OpenRecordset(SQL, dbOpenDynaset)
RS.Edit
RS![議席数] = RS![議席数] + 1
RS.Update
RS.Close
```

```
' 議席数合計を1増やす。
total = total + 1
```

```
Wend
Block.MoveNext
```

```
Wend
Block.Close
DB.Close
```

End Sub

Option Compare Database  
Option Explicit

```
' 比例代表の当選者を決定する。
Sub 比例当選者決定()
    Dim DB As Database
    Dim RS1, RS2, RS3 As Recordset
    Dim SQL As String
    Dim N As Integer      ' 獲得議席数
    Dim i As Integer

    Set DB = DBEngine.Workspaces(0).Databases(0)

    ' 比例代表の全立候補者を「落選」とし、惜敗率を設定する。
    SQL = "UPDATE 比例代表名簿 SET 当落 = False, 惜敗率 = 惜敗率(候補者番号);"
    DB.Execute SQL, dbFailOnError

    ' ブロック毎に以下の処理を実行する。
    Set RS1 = DB.OpenRecordset("比例代表ブロック", dbOpenSnapshot)
    While Not RS1.EOF

        ' 党派毎に以下の処理を実行する。
        SQL = "SELECT * FROM 比例代表議席数 WHERE ブロック = '" & RS1![ブロック] & "';"
        Set RS2 = DB.OpenRecordset(SQL, dbOpenSnapshot)
        While Not RS2.EOF

            ' 獲得議席数 N を求める。
            N = RS2![議席数]

            ' 当該ブロックおよび党派の立候補者一覧を求める。
            ' (小選挙区当選者を除く。名簿順位と惜敗率で並べ替える。)
            SQL = "SELECT * FROM 比例代表名簿 INNER JOIN 立候補者"
            SQL = SQL & " ON 比例代表名簿.候補者番号 = 立候補者.候補者番号"
            SQL = SQL & " WHERE ブロック = '" & RS1![ブロック] & "'"
            SQL = SQL & " AND 党派 = '" & RS2![党派] & "' AND 惜敗率 < 1"
            SQL = SQL & " ORDER BY 名簿順位, 惜敗率 DESC;"
            Set RS3 = DB.OpenRecordset(SQL, dbOpenDynaset)

            ' 上記リスト中で上位 N 人の候補者を当選とする。
            For i = 1 To N
                RS3.Edit
                RS3![当落] = True
                RS3.Update
                RS3.MoveNext
            Next

            RS3.Close
            RS2.MoveNext
        Wend

        RS2.Close
        RS1.MoveNext
    Wend

    RS1.Close
    DB.Close
End Sub
```