

Excel Access to Databases

(All puns intended)*

*File: misnotes-dbandexcel.tex.

How to work with (get) database files in Excel?

Main ways:

1. Export from Access as Excel (.xls) files, then open from Excel.
2. Use the Query facility in Excel to access Access.

RTFM: Chapter 24, "Using External Database Files," in Walkenbach.

How to work in Excel with database files?

3. Use VBA in Excel and the DAO (data access object) facility to directly read and write to Access or any ODBC-compliant DBMS.

It can give you more or less complete access to a DBMS, including one on a network.

Will briefly cover DAO for Access. ODBC is more general and (slightly) more complicated. See the online help.

DBMS tables in Excel via export from Access

1. Begin by getting the data/database you need into MS Access.

Can be imported from various other DBMSs. Also, can read, e.g., comma-delimited files.

2. Manipulate the data within Access, e.g., by creating queries which you then store.
3. Export the table or query, as an Excel (.xls) file. (Use Save As/Export under the File menu.)
4. Launch Excel and open the exported file.

DAO (Data Access Object) Model

- In Excel, in the VBA editor:

Tools | References. . .

Pick:

Microsoft DAO 3.5 Object Library [Excel 97]

Microsoft DAO 3.6 Object Library [Excel 2000]

for each workbook in which you do DAO.

Example VBA Code with DAO

See: DAO-example.xls; online help in Access
(not Excel!)

```
Option Explicit
```

```
Sub testDAO()
```

```
Dim dbe As DAO.DBEngine
```

```
Dim db As DAO.Database
```

```
Dim rst As DAO.Recordset
```

```
Dim qd As QueryDef
```

```
Dim strDBName As String
```

```
Dim strSQL As String
```

```
strDBName = CurDir & "\spj-begin.mdb"
```

Example VBA Code with DAO con't.

```
'36 for Excel 2000, 35 for Excel 97
'Set dbe = CreateObject("DAO.DBEngine.36")
Set dbe = CreateObject("DAO.DBEngine.35")
Set db = dbe.Workspaces(0).OpenDatabase(strDBName)
Set rst = db.OpenRecordset("sp", dbOpenTable)

MsgBox rst.RecordCount & _
    " records in SP table."

strSQL = "SELECT DISTINCT SP.[P#], " & _
    "SP.QTY FROM SP " & _
    "WHERE QTY = 300 ORDER BY " & _
    "SP.[P#] DESC"
Set qd = db.CreateQueryDef("", strSQL)
Set rst = qd.OpenRecordset(dbOpenSnapshot)

With Application.Worksheets("Sheet1")
    .Cells(1, 1).CopyFromRecordset rst
    .Cells(3, 2).Formula = "=sum(B1:B2)"
End With
```

Example VBA Code with DAO con't.

```
' cf. MoveFirst, MoveNext  
rst.MoveLast  
MsgBox rst![P#] & " is the last part number."  
  
db.Close  
End Sub
```

Also: To add in Solver (from the worksheet/Excel view):

Tools|Add-Ins. . .

then Solver Add-In

And see the last page of the case: Submit by email!