A Blog Article by

Geek Must Have

Crib Sheets

John HR Schuster

Version 2.1e, 07/08/2020

Table of Contents

These are crib notes. Every time I look something up more than once I try to add it to this document. Something simple.

The PDF version of this document is available at this Link

1. MSAccess

1.1. Substring (MID)

The Microsoft Access Mid function extracts a substring from a string (starting at any position).

```
Mid ( text, start_position, [number_of_characters] )
```

https://www.techonthenet.com/access/functions/string/mid.php

1.2. Date Difference (DateDif)

The DateDiff() function returns the difference between two dates.

DateDiff

```
DateDiff(datepart, date1, date2, firstdayofweek, firstweekofyear)

'*--- Return the difference between two dates, in years:
SELECT DateDiff("yyyy", #13/01/1998#, #09/05/2017#);
```

Parameter	Description
datepart	Required. The part to return. Can be one of the following values: * yyyy = Year . q = Quarter . m = month . y = Day of the year . d = Day . w = Weekday . ww = Week . h = hour . n = Minute . s = Second

date1 and date2 Required. The two dates to calculate the difference between firstdayofweek Optional. Specifies the first day of the week. Can be one of the following values: 0 = Use the NLS API setting 1 = Sunday (this is default) 2 = Monday 3 = Tuesday 4 = Wednesday 5 = Thursday 6 = Friday 7 = Saturday firstdayofyear Optional. Specifies the first week of the year. Can be one of the following values: 0 = Use the NLS API setting 1 = Use the first week that includes Jan 1st (default) 2 = Use the first week in the year that has at least 4 days 3 = Use the first full week of the year

https://www.w3schools.com/sql/func_msaccess_datediff.asp

Format	Explanation

General Date	Displays date based on your system settings Long Date Displays date based on your system's long date setting Medium Date Displays date based on your system's medium date setting Short Date Displays date based on your system's short date setting Long Time Displays time based on your system's long time setting Medium Time Displays time based on your system's medium time setting Short Time Displays time based on your system's short time setting https://www.techonthenet.com/access/functions/date/format.php

1.3. Format values

The Format function does exactly the same thing as formatting a number or a date within a cell in a spreadsheet, except it does so from within the code itself. If you wish to display a number in a message box or on a user form, this function is very useful for making it readable, particularly if it is a large number

Examples

```
Format(1234567.89, "#,###.#")

format(([On_Hand]-[Qty_Needed]), "#,##0[Black];(#,##0)[Red];0;0")

sReturn = Format(sValueIS, "$###,###,###,##0[Black];($###,###,###,##0)[Red];;")
```

Table 1. Predefined Formats

Format Name	Description
General Number	Display the number as is.
Currency	Display the number with currency symbol. Use thousand separator. Enclose in brackets if negative. Display to two decimal places.
Fixed	Display at least one digit to the left and two digits to the right of the decimal point.
Standard	Display number with thousand separator. Display to two decimal places.
Percent	Display number multiplied by 100 with a percent sign (%) appended after. Display to two decimal places.
Scientific	Use standard scientific notation.
Yes/No	Display No if number is 0; otherwise, display Yes.
True/False	Display False if number is 0; otherwise, display True.
On/Off	Display Off if number is 0; otherwise, display On.

A number of characters can be used to define a user-defined format, as shown below. The format string can have up to four sections separated by semicolons (;). This is so different formats can be

applied to different values, such as to positive and negative numbers. For example, you may wish to show brackets/parentheses around a negative value

Table 2. Section Detail

Number of Sections	Formatting
One section only	Applies to all values
Two sections	First section for positive values, second section for negative values
Three sections	First section for positive values, second section for negative values, third section for zeros
Four sections	First section for positive values, second section for negative values, third section for zeros, fourth section for null values

Table 3. User-Defined Formats

Character	Description
Null String	No formatting.
0	Digit placeholder. Displays a digit or a zero. If there is a digit for that position, then it displays the digit; otherwise, it displays 0. If there are fewer digits than zeros, you will get leading or trailing zeros. If there are more digits after the decimal point than there are zeros, then the number is rounded to the number of decimal places shown by the zeros. If there are more digits before the decimal point than zeros, these will be displayed normally.
#	Digit placeholder. This displays a digit or nothing. It works the same as the preceding zero placeholder, except that leading and trailing zeros are not displayed. For example, 0.75 would be displayed using zero placeholders, but this would be .75 using # placeholders.
.Decimal point.	Only one permitted per format string. This character depends on the settings in the Windows Control Panel.
%	Percentage placeholder. Multiplies number by 100 and places % character where it appears in the format string.
,	Thousand separator. This is used if 0 or # placeholders are used and the format string contains a comma. One comma to the left of the decimal point means to round to the nearest thousand (e.g., 0,). Two adjacent commas to the left of the thousand separator indicate rounding to the nearest million (e.g., 0,,).
E- E+	Scientific format. This displays the number exponentially.
:	Time separator-used when formatting a time to split hours, minutes, and seconds.
/	Date separator-this is used when specifying a format for a date.
- + $ ( )	Displays a literal character. To display a character other than listed here, precede it with a backslash (\).

Table 4. Predefined Date and Time Formats

Format Name	Description
General Date	Display a date and/or time. For real numbers, display date and time. Integer numbers display time only. If there is no integer part, then display only time.
Long Date	Displays a long date as defined in the international settings of the Windows Control Panel.
Medium Date	Displays a date as defined in the short date settings of the Windows Control Panel, except it spells out the month abbreviation.
Short Date	Displays a short date as defined in the International settings of the Windows Control Panel.
Long Time	Displays a long time as defined in the International settings of the Windows Control Panel.
Medium Time	Displays time in a 12-hour format using hours, minutes, and seconds and the AM/PM format.
Short Time	Displays a time using 24-hour format (e.g., 18:10).

https://sourcedaddy.com/ms-access/format-function.html

1.4. Check if folder or file exists

Access detect file

```
Function FileExists(ByVal strFile As String, Optional bFindFolders As Boolean) As
Boolean
    'Purpose:   Return True if the file exists, even if it is hidden.
    'Arguments: strFile: File name to look for. Current directory searched if no path
included.
    '           bFindFolders. If strFile is a folder, FileExists() returns False
unless this argument is True.
    'Note:      Does not look inside subdirectories for the file.
    'Author:    Allen Browne. http://allenbrowne.com June, 2006.
    Dim lngAttributes As Long

    'Include read-only files, hidden files, system files.
    lngAttributes = (vbReadOnly Or vbHidden Or vbSystem)

    If bFindFolders Then
        lngAttributes = (lngAttributes Or vbDirectory) 'Include folders as well.
    Else
        'Strip any trailing slash, so Dir does not look inside the folder.
        Do While Right$(strFile, 1) = "\"
            strFile = Left$(strFile, Len(strFile) - 1)
        Loop
    End If

    'If Dir() returns something, the file exists.
    On Error Resume Next
    FileExists = (Len(Dir(strFile, lngAttributes)) > 0)
End Function

Function FolderExists(strPath As String) As Boolean
    On Error Resume Next
    FolderExists = ((GetAttr(strPath) And vbDirectory) = vbDirectory)
End Function

Function TrailingSlash(varIn As Variant) As String
    If Len(varIn) > 0 Then
        If Right(varIn, 1) = "\" Then
            TrailingSlash = varIn
        Else
            TrailingSlash = varIn & "\"
        End If
    End If
End Function
```

1.5. Copy file

Access Copy File

```
'---------------------------------------------------------------------------------
```

```
--
' Procedure : CopyFile
' Author    : Daniel Pineault, CARDA Consultants Inc.
' Website   : http://www.cardaconsultants.com
' Purpose   : Copy a file
'             Overwrites existing copy without prompting
'             Cannot copy locked files (currently in use)
' Copyright : The following is release as Attribution-ShareAlike 4.0 International
'             (CC BY-SA 4.0) - https://creativecommons.org/licenses/by-sa/4.0/
' Req'd Refs: None required
'
' Input Variables:
' ~~~~~~~~~~~~~~~~
' sSource - Path/Name of the file to be copied
' sDest - Path/Name for copying the file to
'
' Revision History:
' Rev       Date(yyyy/mm/dd)        Description
'
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
' 1         2007-Apr-01             Initial Release
'---------------------------------------------------------------------------------
--
Public Function CopyFile(sSource As String, sDest As String) As Boolean
On Error GoTo CopyFile_Error

    FileCopy sSource, sDest
    CopyFile = True
    Exit Function

CopyFile_Error:
    If Err.Number = 0 Then
    ElseIf Err.Number = 70 Then
        MsgBox "The file is currently in use and therfore is locked and cannot be
copied at this" & _
               " time.  Please ensure that no one is using the file and try again.",
vbOKOnly, _
               "File Currently in Use"
    ElseIf Err.Number = 53 Then
        MsgBox "The Source File '" & sSource & "' could not be found.  Please validate
the" & _
               " location and name of the specifed Source File and try again",
vbOKOnly, _
               "File Currently in Use"
    Else
        MsgBox "MS Access has generated the following error" & vbCrLf & vbCrLf &
"Error Number: " & _
               Err.Number & vbCrLf & "Error Source: CopyFile" & vbCrLf & _
               "Error Description: " & Err.Description, vbCritical, "An Error has
Occurred!"
    End If
```

```
    Exit Function
End Function
```

https://www.devhut.net/2010/09/29/ms-access-vba-copy-a-file/

1.6. Hide second column in Combo box

In Visual Basic, the ColumnWidth property setting is an Integer value that represents the column width in twips. You can specify a width or use one of the following predefined settings.

When you use a `0` as a ColumnWidth, that columns is not available in vba.

Table 5. Special Column Width Values

Setting	Description
0	Hides the column.
1	(Default) Sizes the column to the default width.

The Alternative method which allows the column to be used is setting the first columns width to something like `5"`

1.7. Refresh Subform with new QueryDef SQL

You can't requery, you have to refresh the subform source object:

```
MySubformControl.SourceObject = ""
MySubformControl.SourceObject = "Query.MyQuery"
```

1.8. Convert expression to data type (C)

The function name determines the return type as shown in the following:

Cast and Convert

```
-- CAST Syntax:
CAST ( expression AS data_type [ ( length ) ] )

-- CONVERT Syntax:
CONVERT ( data_type [ ( length ) ] , expression [ , style ] )
```

1.9. Replace

The REPLACE() function replaces all occurrences of a substring within a string, with a new substring.

The search is case-insensitive.

Also look at the STUFF() function.

Replace

```
REPLACE(string, old_string, new_string)
```

Source:

1.10. SQL LIKE

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the LIKE operator:

% - The percent sign represents zero, one, or multiple characters

_ - The underscore represents a single character

MS Access uses an asterisk (*) instead of the percent sign (%), and a question mark (?) instead of the underscore (_).

1.11. SQL LIKE with Underscore

Like UNderscore

```
 where something LIKE '%[_]d'
```

Source: link:https://stackoverflow.com/questions/5821/sql-server-escape-an-underscore[https://stackoverflow.com/questions/5821/sql-server-escape-an-underscore, window='_blank'/

1.12. Create Table

- By Copying all columns from another table Syntax The syntax for the CREATE TABLE AS statement when copying all of the columns in SQL is:

Create Table Select

```
CREATE TABLE new_table
  AS (SELECT * FROM old_table);
```

Source: https://www.techonthenet.com/sql/tables/create_table2.php

1.13. C data conversion

Function	Return Type	Range for expression argument
CBool	Boolean	Any valid string or numeric expression.
CByte	Byte	0 to 255.
CCur	Currency	-922,337,203,685,477.5808 to 922,337,203,685,477.5807.
CDate	Date	Any valid date expression.
CDbl	Double	-1.79769313486231E308 to-4.94065645841247E-324 for negative values; 4.94065645841247E-324 to 1.79769313486232E308 for positive values.
CDec	Decimal	+/-79,228,162,514,264,337,593,543,950,335 for zero-scaled numbers, that is, numbers with no decimal places. For numbers with 28 decimal places, the range is +/-7.9228162514264337593543950335. The smallest possible non-zero number is 0.0000000000000000000000000001.
CInt	Integer	-32,768 to 32,767; fractions are rounded.
CLng	Long	-2,147,483,648 to 2,147,483,647; fractions are rounded.
CSng	Single	-3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values.
CStr	String	Returns for CStr depend on the expression argument.
CVar	Variant	Same range as Double for numerics. Same range as String for non-numerics.

1.14. Creating User Properties (Tables / Queries)

You can create user-defined properties for persistent DAO objects, such as tables and queries. You can't create properties for nonpersistent objects, such as recordsets. To create a user-defined property, you must first create the property, using the Database's CreateProperty method. You then append the property using the Properties collection's Append method. That's all there is to it.

Using the example of a field's Description property, the following code demonstrates just how easy it is:

```
Public Sub SetFieldDescription(strTableName As String, _
    strFieldName As String, _
    varValue As Variant, _
)
    Dim dbs As DAO.Database
    Dim prop As DAO.Property
    Set dbs = CurrentDb

    'Create the property
    Set prop = dbs.CreateProperty("Description", dbText, varValue)

    'Append the property to the object Properties collection
    dbs(strTableName)(strFieldName).Properties.Append prop
    Debug.Print dbs(strTableName)(strFieldName).Properties("Description")

    'Clean up
    Set prop = Nothing
    Set dbs = Nothing
End Sub
```

https://sourcedaddy.com/ms-access/setting-and-retrieving-built-in-object-properties.html

1.15. Turn off datasheet Sort/Filter

On design view go to the properties page. Under the "Other" or "All" tab find Shortcut Menu. Change that property from Yes to No and save. Be warned though that this will disable all shortcuts for the form and not just the drop down filter/sort menus on column headings in datasheet view.

1.16. Determine the Current Object

The CurrentObjectName property is set by Microsoft Access to a string expression containing the name of the active object.

Current Object

```
intCurrentType = Application.CurrentObjectType
strCurrentName = Application.CurrentObjectName
```

1.17. Update LastUpdated and UpdatedBy

These two fields are used in most my code to show when the row was last touched by someone.

Update Audit

```
Private Sub Form_BeforeUpdate(Cancel As Integer)
    '\*--- TimeStanp any change
    Me.LastUpdated = Now()
    Me.UpdatedBy = SetUserName()
End Sub
```

1.18. Determine Current subroutine

Current sub

```
    msgbox  Application.VBE.ActiveCodePane.CodeModule)
    '*--- will return something like
    '* Form_frmIMMTemplateImport
```

1.19. Requery a form from another form

```
e.dirty = false
Forms!frmLegacy.Requery
```

Source: http://www.utteraccess.com/forum/Requery-Form-Form-t2001669.html

1.20. Relinking table VBA

Relgenink Tables

```
Function ReLinkTable(strTable As String, strPath As String) As Boolean
  ' Comments: Re-links the named table to the named path
  ' Params  : strTable     Table name of the linked table
  ' strPath : full path name of the database containing the real table
  ' Returns : True if successful, False otherwise

  Dim fOK As Boolean
  Dim dbs As DAO.Database
  Dim tdf As DAO.TableDef
  Dim strPrefix As String
  Dim strNewConnect As String

  fOK = False

  On Error GoTo PROC_ERR

  Set dbs = CurrentDb()
  Set tdf = dbs.TableDefs(strTable)

  strPrefix = Left$(tdf.Connect, InStr(tdf.Connect, "="))
  strNewConnect = strPrefix & strPath

  tdf.Connect = strNewConnect
  tdf.RefreshLink

  fOK = True

PROC_EXIT:
  dbs.Close
  ReLinkTable = fOK
  Exit Function

PROC_ERR:
  Resume PROC_EXIT
End Function
```

http://www.fmsinc.com/microsoftaccess/databasesplitter/

1.21. Hidden properties of datasheet

The properties in Access related to datasheet.

Table 6. Hidden Properties

Property	Meaning and Usage
ColumnHidden	Exists on columns in the datasheet, controls whether the column is visible or not.

Property	Meaning and Usage
ColumnWidth	Exists on columns in the datasheet, controls the width of the column.
DatasheetBackColor	Exists on the datasheet itself, specifies the background color for the whole datasheet.
DatasheetCellsEffect	Exists on the datasheet itself, handles whether special effects are used for the cells (flat, raised, or sunken are the only effects supported).
DatasheetFontHeight	Exists on the datasheet itself, this unfortunately named property specifies the font size.
DatasheetFontItalic	Exists on the datasheet itself, controls whether all of the text is italic.
DatasheetFontName	Exists on the datasheet itself, controls the name of the font.
DatasheetFontUnderline	Exists on the datasheet itself, controls whether all of the text is underlined.
DatasheetFontWeight	Exists on the datasheet itself, controls whether the text is bolded.
DatasheetForeColor	Exists on the datasheet itself, specifies the foreground color for the whole datasheet.
DatasheetGridlinesBehavior	Exists on the datasheet itself, controls which gridlines will be displayed (if any).
DatasheetGridlinesColor	Exists on the datasheet itself, specifies the color of the gridlines.
FrozenColumns	Exists on the datasheet itself, specifies how many columns have been frozen by the user (discussed later in the article).
ShowGrid	Exists on the datasheet itself, but has been superseded by the DatasheetGridlinesBehavior property.
SubdatasheetExpanded	Exists on the datasheet itself, specifies whether all subdatasheets should be expanded. (Access 2000 only)
SubdatasheetHeight	Exists on the datasheet itself, specifies the number of records to display for subdatasheets (a scrollbar appears if there are more records than this property allows). (Access 2000 only)
SubdatasheetName	Exists on the datasheet itself, specifies the name of the table's subdatasheet. (Access 2000 only)
TabularCharSet	Exists on the datasheet itself, and is hidden. It specifies the font character set and can often cause bad things to happen if it's set to an incorrect value. It's best not to set it, or to set it to 1 (which uses the DEFAULT_CHARSET for the machine).

With the exception of the Subdatasheet properties, you have no direct design-time access to these properties: None of these properties show up in the datasheet property sheet. As a result, they can only be set at runtime from VBA code in order to make changes. Interestingly, none of the properties are exposed by ADO or ADOX, so if you want to change them, you'll have to use DAO.

While you can't access these properties through property sheets, many of them can be set in the user interface. They are, for example, what's changed when you set the font of a datasheet from the Format menu. For full control over the datasheet, though, you'll want to explicitly set the properties in code and save the object when you're done.

It's worth noting that a datasheet is a form it says so right in the object browser. The object

browser considers the datasheet columns to be the controls on the form. As a result, a datasheet can consist of any control that can be displayed, which means all TextBox, ComboBox, and CheckBox controls.

Source: https://docs.microsoft.com/en-us/previous-versions/office/developer/office-2003/aa217449(v=office.11)?redirectedfrom=MSDN

1.22. Change Hide / Show of a field on datasheet

The code to show and hide the columns is in a routine called ShowHideColumn.

```
Private Function ShowHideColumn()
    Dim sfrm As SubForm
    Dim ctl As Control
    Dim stCtl As String

    Set sfrm = Me.sfrmHideShowColumns
    For Each ctl In Me.Controls
        If TypeOf ctl Is Access.CheckBox Then
            stCtl = "tb" & Mid$(ctl.Name, 3)
            sfrm.Form(stCtl).ColumnHidden = _
                       Not ctl.Value
        End If
    Next ctl
End Function
```

1.23. Select Case (If-Then-Else)

The Microsoft Access Case statement can only be used in VBA code. It has the functionality of an IF-THEN-ELSE statement.

```
Select Case test_expression

   Case condition_1
      result_1
   Case condition_2
      result_2
   ...
   Case condition_n
      result_n

 [ Case Else
      result_else ]

End Select
```

Source: https://www.techonthenet.com/access/functions/advanced/case.php

1.24. Access columns of Combo control

Use 0 to refer to the first column, 1 to refer to the second column, and so on. Use 0 to refer to the first row, 1 to refer to the second row, and so on. For example, in a list box containing a column of customer IDs and a column of customer names, you could refer to the customer name in the second column (1) and fifth (4) row as:

If the user has made no selection when you refer to a column in a combo box or list box, the Column property setting will be Null. You can use the IsNull function to determine if a selection has been made

```
Forms!Contacts!Customers.Column(1, 4)

'*--- Empty combo selection
If IsNull(Forms!Customers!Country)
  Then MsgBox "No selection."
End If
```

1.25. Set Filter on Subform from Main Form

```
'*--- lstBoxSheets is the subForm
Me.LstBoxSheets.Form.Filter = "prjCategory='General'"
Me.LstBoxSheets.Form.FilterOn = True
```

1.26. Reset query Column Order setting

So, when you open a query in Datasheet view, and the column order has not been messed with and saved, the column order displayed is determined by the OrdinalPosition and the value of that property corresponds to the order in which your columns appear in the query design grid (OrdinalPosition is 0 based, so 0 is the first column).

Then … when you move the column while viewing the query in Datasheet view, and subsequently save that change in the column order, Access creates the ColumnOrder property for each of the columns in the query. This property is not visible in the query design grid, but is definately there.

```
Public Sub ResetColumnOrder(strQueryName)
    Dim fld As DAO.Field
    Dim qdf As DAO.QueryDef

    Set qdf = CurrentDb.QueryDefs(strQueryName)

    For Each fld In qdf.Fields
        On Error Resume Next
        fld.Properties.Delete "ColumnOrder"
    Next fld

End Sub
```

1.27. Open Dynaset, change SQL

1.28. Open recordset, add row

```
Dim dbCurrent As Database
Dim rsNotes As Recordset
Dim sSQL As String

    sSQL = "Select * from PrePos where PostType = 'Help';"
    Set dbCurrent = CurrentDb
    Set rsNotes = dbCurrent.OpenRecordset(sSQL, dbOpenDynaset, dbSeeChanges)
    With rsNotes
        If .EOF Then
            .AddNew
            ![PostIMMTable] = gsNewTableName
            ![PostIMMField] = gsNewFieldName
            ![PostPtype] = gsPtype
            ![Notes] = Me.txtNotes
            ![UpdatedDate] = Now()
            ![UpdatedBy] = gsUserName
            .Update
            .Close
        End If
    End With
    Set rsNotes = Nothing
    Set dbCurrent = Nothing
```

1.29. Select Cased Statement (VBA)

```
Select Case test_expression

   Case condition_1
      result_1
   Case condition_2
      result_2
   ...
   Case condition_n
      result_n

 [ Case Else
      result_else ]

End Select
```

1.30. Coalesce Function

Access does not have Coalesce function, this quick VBA equivalent. You pass it an array of values.

Coalesce Function

```
Function Coalesce(ParamArray varValues()) As Variant
'returns the first non null value, similar to SQL Server Coalesce() function
'Patrick Honorez --- www.idevlop.com
    Dim i As Long
    Coalesce = Null
    For i = LBound(varValues) To UBound(varValues)
        If Not IsNull(varValues(i)) Then
            Coalesce = varValues(i)
            Exit Function
        End If
    Next
End Function
```

https://stackoverflow.com/questions/247858/coalesce-alternative-in-access-sql

1.31. IsNull

The MS Access IsNull() function returns TRUE (-1) if the expression is a null value, otherwise FALSE (0):

IsNull

```
SELECT ProductName, UnitPrice * (UnitsInStock + IIF(IsNull(UnitsOnOrder), 0,
UnitsOnOrder))
FROM Products;
```

Source: https://www.w3schools.com/sql/sql_isnull.asp

1.32. Print

Print

```
PRINT msg_str | @local_variable | string_expr
```

msg_str Is a character string or Unicode string constant. For more information, see Constants (Transact-SQL).

@ local_variable Is a variable of any valid character data type. @local_variable must be char, nchar, varchar, or nvarchar, or it must be able to be implicitly converted to those data types.

string_expr Is an expression that returns a string. Can include concatenated literal values, functions, and variables. For more information, see Expressions (Transact-SQL).

Print can not be used in Functions

1.33. List Columns (Schema)

List Columns

```
select schema_name(tab.schema_id) as schema_name,
    tab.name as table_name,
    col.column_id,
    col.name as column_name,
    t.name as data_type,
    col.max_length,
    col.precision
from sys.tables as tab
    inner join sys.columns as col
        on tab.object_id = col.object_id
    left join sys.types as t
    on col.user_type_id = t.user_type_id
order by schema_name,
    table_name,
    column_id;
```

Source: https://dataedo.com/kb/query/sql-server/list-table-columns-in-database

List Columns (More Detail)

```
SELECT
     SysTbls.name AS [Table Name]
    ,SysCols.name AS [Column Name]
    ,ExtProp.value AS [Extended Property]
    ,Systyp.name AS [Data Type]
    ,CASE WHEN Systyp.name IN('nvarchar','nchar')
               THEN (SysCols.max_length / 2)
          WHEN Systyp.name IN('char')
               THEN SysCols.max_length
          ELSE NULL
          END AS 'Length of Column'
    ,CASE WHEN SysCols.is_nullable = 0
               THEN 'No'
          WHEN SysCols.is_nullable = 1
               THEN 'Yes'
          ELSE NULL
          END AS 'Column is Nullable'
    ,SysObj.create_date AS [Table Create Date]
    ,SysObj.modify_date AS [Table Modify Date]
FROM sys.tables AS SysTbls
   LEFT JOIN sys.extended_properties AS ExtProp
         ON ExtProp.major_id = SysTbls.[object_id]
   LEFT JOIN sys.columns AS SysCols
         ON ExtProp.major_id = SysCols.[object_id]
         AND ExtProp.minor_id = SysCols.column_id
   LEFT JOIN sys.objects as SysObj
         ON SysTbls.[object_id] = SysObj.[object_id]
   INNER JOIN sys.types AS SysTyp
         ON SysCols.user_type_id = SysTyp.user_type_id
WHERE class = 1 --Object or column
  AND SysTbls.name IS NOT NULL
  AND SysCols.name IS NOT NULL
```

1.34. Close form if it is open

Close Open form

```
    '\*-- Close out main menu if open
    If CurrentProject.AllForms("frmAAP_ObjectMenuList").IsLoaded = True Then
        DoCmd.Close acForm, "frmAAP_ObjectMenuList"
    End If
```

Source: https://social.msdn.microsoft.com/Forums/office/en-US/6ea91117-eb79-41dd-9c98-382f0577f45e/how-do-i-test-to-see-if-a-form-is-open

1.35. Close all forms and Reports

This is good to run when closing out an application

Close all forms

```
Sub CloseAllFormsReports()
On Error GoTo CloseAllFormsReports_err

'*--- Close all open forms
Do While Forms.Count > 0
   DoCmd.Close acForm, Forms(0).Name
Loop

'*--- Close all open reports
Do While Reports.Count > 0
    DoCmd.Close acReport, Reports(0).Name
Loop


CloseAllFormsReports_Exit:
    Exit Sub

CloseAllFormsReports_err:
    LogError
    Resume CloseAllFormsReports_Exit

End Sub
```

1.36. Sleep, wait, timer

The Sleep in Kerel32 will cause errors on some users environment. This small function will do the same thing without Kernel32.

Timer

```
Function WaitTime(n As Double)
'Function that wait an amount of time n in seconds
TWait = Time
TWait = DateAdd("s", n, TWait)
Do Until TNow >= TWait
     TNow = Time
Loopgen-doco

End Function
```

Source: https://stackoverflow.com/questions/469347/is-there-an-equivalent-to-thread-sleep-in-vba

1.37. Set focus to Form in split form

Since all of the data fields are both parts of the split form. A focus to a field goes to the list on top.

Focus to Form in split form

```
    '*--- TRICK to get focus into form part of split form
    Me.btnClose.SetFocus  '*--- Must be a button
    Me.QueryName.SetFocus   '*--- First text field on form
```

1.38. Close all Windows VB Editor

AFter updating an application the VB editor has many windows open. This can affect performance and increase change of a DB fail.

This code put ito a module can be run by clicking into and running it F5.

Close all VBE windows

```
Sub Close_All_VBE_Windows() 'CR v5207

'// Source: https://access-programmers.co.uk/foru...
'// Thanks to: Colin Ridders (https://access-programmers.co.uk/foru...)
On Error GoTo Err_Handler

Dim vbWin As VBIDE.Window

For Each vbWin In Application.VBE.Windows
     If (vbWin.Type = vbext_wt_CodeWindow Or _
         vbWin.Type = vbext_wt_Designer) And _
         Not vbWin Is Application.VBE.ActiveWindow Then
             vbWin.Close
     End If
 Next

Exit_Handler:
    Exit Sub

Err_Handler:
'CR 02/02/2016 - added error handling to fix issue in 64-bit Office
    If err.Number = 424 Then Resume Next 'object required
    MsgBox "Error " & err.Number & " in Close_All_VBE_Windows procedure: " &
err.Description
    Resume Exit_Handler

End Sub
```

1.39. Close all VB Editor Windows

There is not a simple Access command to close all open VBA editor windows.

Here is a subrotuine that you can add to the project and then just go to where it is saved and `F5` run.

Close ALL VBA Editor Windows

```
Sub Close_All_VBE_Windows() 'CR v5207

'// Source: https://access-programmers.co.uk/foru...
'// Thanks to: Colin Ridders (https://access-programmers.co.uk/foru...)
On Error GoTo Err_Handler

Dim vbWin As VBIDE.Window

For Each vbWin In Application.VBE.Windows
     If (vbWin.Type = vbext_wt_CodeWindow Or _
         vbWin.Type = vbext_wt_Designer) And _
         Not vbWin Is Application.VBE.ActiveWindow Then
             vbWin.Close
     End If
 Next

Exit_Handler:
    Exit Sub

Err_Handler:
'CR 02/02/2016 - added error handling to fix issue in 64-bit Office
    If err.Number = 424 Then Resume Next 'object required
    MsgBox "Error " & err.Number & " in Close_All_VBE_Windows procedure: " &
err.Description
    Resume Exit_Handler

End Sub
```

Reference: https://www.youtube.com/watch?v=uGES0z7eqO0

1.40. Pass parameters to Form

It is possible to pass parameter to a form from with the VBA OpenForm command. This can help to reduce the number of global variables.

Pass parameters to form

```
    '*--- Open edit form in ()Add) MODE
    DoCmd.OpenForm "frmSAM_PersonalMaintenance", acNormal, , , acFormAdd, , "ADD"
```

In this example `"ADD"` is the parameter (OpenARG) being passed to the

frmSAM_PersonalMaintenance form.

Inside of the frmSAM_PersonalMaintenance sub here is an example of how to use the (OpenArg)

Use parameter in form

```
    If Me.OpenArgs = "ADD" Then ①
        '*-- Set deault values
        Me.SQLType = 1              '*--- Access sql type
        Me.ReturnsRecords = True    '*--- Select queries
        Me.QueryType = 16            '*--- Local Read only, not passthrough
        Me.AccessRole = 11          '*--- Owner role
        Me.DSN_ID = 8               '*--- Default DSN
        Me.Category = "Private"     '*--- Private owner query

        Me.CreatedBy = gsUserName
        Me.CreatedDate = Now()
    End If
```

① me.OpenARg is the one parm being sent to form.

You could put multiple elements in the OpenArg parameter. By combining your values into one string separated by a character that would not be in your string, you can overcome the OpenArgs limitation and still use DoCmd to pass them. The OpenArgs parameter is a string that the form can read once it is opened. Calling the form, add the string to the OpenArgs parameter like this

Setting Multiple Parms

```
DoCmd.OpenForm "frmName", , , , , , "cboCategory|" & txtCategoryID
```

Using multiple parms

```
Private Sub Form_Load()
  Dim intPos As Integer
  Dim strControlName As String
  Dim strValue As String

  If Len(Me.OpenArgs) > 0 Then
    ' Position of the pipe
    intPos = InStr(Me.OpenArgs, "|")

    If intPos > 0 Then

      ' Retrieve Control Name from the first part of the string
      strControlName = Left$(Me.OpenArgs, intPos - 1)

      ' Retrieve Value to Assign from the end of the string
      strValue = Mid$(Me.OpenArgs, intPos + 1)

      ' Assign the value to the control
      Me(strControlName) = strValue

    End If
  End If
End Sub
```

Source: https://www.fmsinc.com/MicrosoftAccess/forms/openargs/index.htm

The following example shows how to use the OpenArgs property to prevent a form from being opened from the navigation pane.

OpenARg in Security

```
Private Sub Form_Open(Cancel As Integer)

If Me.OpenArgs() <> "Valid User" Then
    MsgBox "You are not authorized to use this form!", _
        vbExclamation + vbOKOnly, "Invalid Access"
    Cancel = True
End If
End Sub
```

2. MySQL

2.1. Like with Join

```
SELECT table1.\*, table2.z
FROM table1
INNER JOIN table2
  ON table2.name LIKE CONCAT('%', table1.name, '%')
 AND table1.year = table2.year
```

2.2. Update with Where

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

2.3. Update with Join

```
UPDATE T1, T2,
[INNER JOIN | LEFT JOIN] T1 ON T1.C1 = T2. C1
SET T1.C2 = T2.C2,
    T2.C3 = expr
WHERE condition
```

https://www.mysqltutorial.org/mysql-update-join/

2.4. Count Distinct

You can use the DISTINCT clause within the COUNT function. For example, the SQL statement below returns the number of unique departments where at least one employee makes over $55,000 / year.

```
SELECT COUNT(DISTINCT department) AS "Unique departments"
FROM employees
WHERE salary > 55000;
```

2.5. Count nulls

```
select sum(case when FirstName IS NULL then 1 else 0 end) as NUMBER_OF_NULL_VALUE from
DemoTable;
```

Source: https://www.tutorialspoint.com/how-to-count-null-values-in-mysql

2.6. Add a Auto Increment column to a table

Add Auto Increment

```
ALTER TABLE ThreeSeasons ADD column id INT NOT NULL AUTO_INCREMENT unique first
```

2.7. Case (Select)

The CASE statement goes through conditions and return a value when the first condition is met (like an IF-THEN-ELSE statement). So, once a condition is true, it will stop reading and return the result.

If no conditions are true, it will return the value in the ELSE clause.

If there is no ELSE part and no conditions are true, it returns NULL.

Case

```
SELECT OrderID, Quantity,
CASE
    WHEN Quantity > 30 THEN "The quantity is greater than 30"
    WHEN Quantity = 30 THEN "The quantity is 30"
    ELSE "The quantity is under 30"
END
FROM OrderDetails;
```

2.8. DBA Display all columns of Tables in DB

MySQL Column List

```
select
    tab.name as table_name,
    col.column_id,
    col.name as column_name,
    t.name as data_type,
    col.max_length,
    col.precision
from sys.tables as tab
    inner join sys.columns as col
        on tab.object_id = col.object_id
    left join sys.types as t
    on col.user_type_id = t.user_type_id
```

3. SQL Server

3.1. Concat

```
CONCAT(string1, string2, ...., string_n)
```

3.2. Cast Number into string

```
CAST(expression AS datatype(length))
```

Table 7. Parameter Values

Value	Description
expression	Required. The value to convert
datatype	Required. The datatype to convert expression to. Can be one of the following: bigint, int, smallint, tinyint, bit, decimal, numeric, money, smallmoney, float, real, datetime, smalldatetime, char, varchar, text, nchar, nvarchar, ntext, binary, varbinary, or image
(Length)	Optional. The length of the resulting data type (for char, varchar, nchar, nvarchar, binary and varbinary) Technical Details

3.3. Round

Round

```
ROUND(number, decimals, operation)

-- Round number to nearest decimal
round(DIHTA.ValueIS, 0, 1)
```

Table 8. Round arguments

Parameter	**Description**
number	Required. The number to be rounded
decimals	Required. The number of decimal places to round number to
operation	Optional. If 0, it rounds the result to the number of decimal. If another value than 0, it truncates the result to the number of decimals. Default value is 0

3.4. Find string within a string

```
SELECT CHARINDEX('t', 'Customer') AS MatchPosition;
```

3.5. Create Table Select (Copy Table)

The SELECT INTO statement copies data from one table into a new table.

```
SELECT *
INTO newtable [IN externaldb]
FROM oldtable
WHERE condition;
```

Practical Example

```
select distinct FieldName
    into New_DealPathway_B2
from dbo.Pre_Migration_Guide
```

3.6. IF then else

```
IF Boolean_expression
     { sql_statement | statement_block }
[ ELSE
     { sql_statement | statement_block } ]
```

3.7. Immediate If (IIF)

```
IIF(condition, value_if_true, value_if_false)
```

3.8. Variables

```
-- Declare a variable with a data type
DECLARE @model_year SMALLINT;

-- Set a variable to a value
SET @model_year = 2018;

-- Use variable in query
SELECT
    product_name,
    model_year,
    list_price
FROM
    production.products
WHERE
    model_year = @model_year
ORDER BY
    product_name;

-- Set Variable in query
SELECT
    @product_name = product_name,
    @list_price = list_price
FROM
    production.products
WHERE
    product_id = 100;
```

3.9. Substring (Substr)

```
SUBSTRING(string, start, length)

-- Example
SELECT
    email,
    SUBSTRING(
        email,
        CHARINDEX('@', email)+1,
        LEN(email)-CHARINDEX('@', email)
    ) domain
FROM
    sales.customers
ORDER BY
    email;
```

3.10. Update with a Join

```
UPDATE
    t1
SET
    t1.c1 = t2.c2,
    t1.c2 = expression,
    ...
FROM
    t1
    [INNER | LEFT] JOIN t2 ON join_predicate
WHERE
    where_predicate;
```

3.11. Charindex (InStr)

```
SELECT
    email,
    SUBSTRING(
        email,
        CHARINDEX('@', email)+1,
        LEN(email)-CHARINDEX('@', email)
    ) domain
FROM
    sales.customers
ORDER BY
    email;
```

3.12. Len (Length) of string

```
SELECT
    email,
    SUBSTRING(
        email,
        CHARINDEX('@', email)+1,
        LEN(email)-CHARINDEX('@', email)
    ) domain
FROM
    sales.customers
ORDER BY
    email;
```

3.13. Copy Table

The SELECT INTO statement copies data from one table into a new table.

```
SELECT *
INTO newtable [IN externaldb]
FROM oldtable
WHERE condition;
```

Source: https://www.w3schools.com/sql/sql_select_into.asp

https://www.w3schools.com/sql/sql_select_into.asp

3.14. Parameters in SP

Create a query torepeatedly to get the data for different sales people, you could instead parameterize the query and turn it into a stored procedure like:

```
create procedure getSalesperson
@sp varchar(25)
as
select SalesPerson, Mon, amount
from SalesData
where SalesPerson = @sp;
Go

-- Run the SP
declare @sp varchar(25)
set @sp = 'Jack'
exec getSalesperson @sp
```

https://www.mssqltips.com/sqlservertip/2981/using-parameters-for-sql-server-queries-and-stored-procedures/

3.15. CharIndex (Substring)

The CHARINDEX() function searches for a substring in a string, and returns the position. If the substring is not found, this function returns 0. Note: This function performs a case-insensitive search.

CharIndex

```
CHARINDEX(substring, string, start)
```

Source: https://www.w3schools.com/sql/func_sqlserver_charindex.asp

3.16. Declare (Variables)

In SQL Server (Transact-SQL), a variable allows a programmer to store data temporarily during the execution of code.

Declare Variable

```
-- Declare the variable to be used.
DECLARE @MyCounter int;

-- Initialize the variable.
SET @MyCounter = 0;

-- Alternative example
DECLARE @techonthenet VARCHAR(50);

SET @techonthenet = 'Example showing how to declare variable';
```

3.17. Case then

The CASE statement goes through conditions and returns a value when the first condition is met (like an IF-THEN-ELSE statement). So, once a condition is true, it will stop reading and return the result. If no conditions are true, it returns the value in the ELSE clause.

Case then

```
CASE expression
   WHEN value_1 THEN result_1
   WHEN value_2 THEN result_2
   WHEN value_n THEN result_n
   ELSE result
END

-- Or

CASE
   WHEN condition_1 THEN result_1
   WHEN condition_2 THEN result_2
   WHEN condition_n THEN result_n
   ELSE result
END
```

Source: https://www.w3schools.com/sql/sql_case.asp

Source: https://www.techonthenet.com/sql_server/functions/case.php

3.18. Adding columns with Nulls

When you add a null to a total the total becomes null. Change Nulls into zero

Adding Nulls

```
-- Method 1 SQL Server specific
isnull(P01,0)

-- Method 2  ANSI Standard
COALESCE(P01,0)
```

Source: https://stackoverflow.com/questions/1088648/sql-sum-3-columns-when-one-column-has-a-null-value

3.19. Line Continuation (Backslash)

Continuation

```
<first section of string> \
<continued section of string>
```

Source: https://docs.microsoft.com/en-us/sql/t-sql/language-elements/sql-server-utilities-statements-backslash?view=sql-server-ver15

3.20. Merge (UpSert)

Lets create a MERGE statement to INSERT or UPDATE a row in table ClientData:

Merge

```
MERGE dbo.ClientData AS [Target]
USING (SELECT 12345 AS clientId) AS [Source]
ON [Target].clientId = [Source].clientId
WHEN MATCHED THEN
  UPDATE SET [Target].data='Update', [Target].updatedDateUtc = GetUtcDate()
WHEN NOT MATCHED THEN
  INSERT (clientId, data) VALUES ([Source].clientId, 'Insert');
```

Source: https://myadventuresincoding.wordpress.com/2016/01/05/sql-server-how-to-write-an-upsert-using-merge/

4. SQLite

4.1. DSNLess ODBC connection

```
DRIVER=SQLite3 ODBC
Driver;Database=c:\mydb.db;LongNames=0;Timeout=1000;NoTXN=0;SyncPragma=NORMAL;StepAPI=
0;

-- Zortero Example
ODBC;DSN=Zotero;Database=C:\Users\{userDirectory}\Zotero\link_zotero.sqlite;StepAPI=0;
SyncPragma=OFF;NoTXN=0;Timeout=;ShortNames=0;LongNames=0;NoCreat=0;NoWCHAR=0;FKSupport
=0;JournalMode=;OEMCP=0;LoadExt=;BigInt=0;JDConv=0;;TABLE=collections
```

This is some VB code to create the DSNLess Connection

DSNLess VB Code

```
'//Name     :   AttachDSNLessTable
'//Purpose  :   Create a linked table to SQL Server without using a DSN
'//Parameters
'//     stLocalTableName: Name of the table that you are creating in the current
database
'//     stRemoteTableName: Name of the table that you are linking to on the SQL Server
database
'//     stServer: Name of the SQL Server that you are linking to
'//     stDatabase: Name of the SQL Server database that you are linking to
'//     stUsername: Name of the SQL Server user who can connect to SQL Server, leave
blank to use a Trusted Connection
'//     stPassword: SQL Server user password
Function AttachDSNLessTable(stLocalTableName As String, stRemoteTableName As String,
stServer As String, stDatabase As String, Optional stUsername As String, Optional
stPassword As String)
    On Error GoTo AttachDSNLessTable_Err
    Dim td As TableDef
    Dim stConnect As String

For Each td In CurrentDb.TableDefs
        If td.Name = stLocalTableName Then
            CurrentDb.TableDefs.Delete stLocalTableName
        End If
    Next

If Len(stUsername) = 0 Then
        '//Use trusted authentication if stUsername is not supplied.
        stConnect = "ODBC;DRIVER=SQL Server;SERVER=" & stServer & ";DATABASE=" &
stDatabase & ";Trusted_Connection=Yes"
    Else
        '//WARNING: This will save the username and the password with the linked table
information.
        stConnect = "ODBC;DRIVER=SQL Server;SERVER=" & stServer & ";DATABASE=" &
stDatabase & ";UID=" & stUsername & ";PWD=" & stPassword
    End If
    Set td = CurrentDb.CreateTableDef(stLocalTableName, dbAttachSavePWD,
stRemoteTableName, stConnect)
    CurrentDb.TableDefs.Append td
    AttachDSNLessTable = True
    Exit Function

AttachDSNLessTable_Err:

AttachDSNLessTable = False
    MsgBox "AttachDSNLessTable encountered an unexpected error: " & Err.Description

End Function
```

https://docs.microsoft.com/en-us/office/troubleshoot/access/create-dsn-less-connection-linkted-table

4.2. Get last ID of a table

Last Record

```
Dim lastID As Integer
lastID = DMax("IDField","YourTable")
```

4.3. Copy Text Box to Clipboard

The following example illustrates how to copy the contents of a text box named txtNotes to the Clipboard. .Copy to Clipboard

```
Private Sub cmdCopy_Click()
   Me!txtNotes.SetFocus
   DoCmd.RunCommand acCmdCopy
End Sub
```

https://docs.microsoft.com/en-us/office/vba/access/concepts/windows-api/send-information-to-the-clipboard

5. Powershell

5.1. Extract x Top rows of text file

```
get-content input.txt|select-object -first 10 >output.txt
```

https://stackoverflow.com/questions/28908638/extract-only-the-first-10-lines-of-a-csv-file-in-powershell

6. Document History

Table 9. Document History

Date	Version	Author	Description
07/08/2020	V2.1e	JHRS	close all VBE windows
06/14/2020	V2.1d	JHRS	Charindex, isnull, Variables
06/08/2020	V2.1c	JHRS	added MySQL DBVA
06/04/2020	V2.1b	JHRS	Initial version