Home x Install x Full Index x Tutorial x EnumerateControls x FindOutTableExists x
UseVariablesInSQL x CreateRecordsetFrom x AddRecordToRecordset x CountRecordsRecordset x
LimitsRecordset x MixAccess2baseAndUNO ! User's Guide x AllForms x DatabaseWindow x
ShortcutNotationMore x DLookupSamples x CalculatedField x MultiSelectListboxSelectForm x
FillAutoControlValue x CarryToNewRecord x BrowseThruControls x TipTextForLongValues x
AskBeforeSaving x Sync2Combos x ZoomOnImage x AddAllToBox x KeepFormsSynchro x
SelectListboxOnFirstLetters x MoveItemsBetweenListboxes x SimulateTabbed x SearchStandalone x
CalculatorDialog x ExploreTables x ExtractDataTable x FindPositionRecordset x DMedian function x
DPercentile x ImportImages x ExportImages x CrossTabQuery x

CrossTabQuery

tags:
HowTo

(Q) How can I write a query similar to the crosstab queries I find in MSAccess (also elsewhere) ? To produce something like:

No. Sales per year	1996	1997	1998	Total
Margaret Peacock	31	81	44	156
Janet Leverling	18	71	38	127
Nancy Davolio	26	55	42	123
Laura Callahan	19	54	31	104
Andrew Fuller	16	41	39	96
etc ...				

(R) A crosstab query aggregates data in the form of a matrix. Example: products sales by period. The issue is that the periods to be considered are in the database and can vary over time. Additionally periods can be years, quarters, months, ...
Doing this is feasible by *generating* with a Basic function the appropriate SQL statement.

The solution presented here will work for the SUM() and COUNT() aggregate functions.

Let's consider next tables:

- Products table

Fields	Field Type	Primary
CategoryID	BigInt	
ProductName	Text	
QuantityPerUnit	Text	
ReorderLevel	Integer	
SupplierID	BigInt	
UnitPrice	Number	
UnitsInStock	Integer	
UnitsOnOrder	Integer	
Discontinued	Boolean	
ProductID	BigInt	Y
Picture	Image	

- Customers table

Fields	Field Type	Primary
Address	Text	

City	Text	
CompanyName	Text	
ContactName	Text	
Country	Text	
CustomerID	Text	Y
Fax	Text	
Phone	Text	
PostalCode	Text	
Region	Text	

- Orders table

Fields	Field Type	Primary
CustomerID	Text	
EmployeeID	Integer	
Freight	Number	
OrderDate	Date/Time	
OrderID	BigInt	Y
RequiredDate	Date/Time	
ShipAddress	Text	
ShipCity	Text	
ShipCountry	Text	
ShipName	Text	
ShippedDate	Date/Time	
ShipPostalCode	Text	
ShipRegion	Text	
ShipVia	Integer	

- Order Details table

Fields	Field Type	Primary
Discount	Float	
OrderID	BigInt	Y
ProductID	BigInt	Y
Quantity	Integer	
UnitPrice	Number	

A crosstab query needs at least next inputs:

- One or more row headings: several database fields which will appear in front of each row and on which the aggregation function will be applied => [Rowheadings]] or their aliases
- One column heading: the field varying horizontally (periods, ...) => (ColHeading] or its alias
- The (numeric) value which will be aggregated => [Aggregate]
- The FROM expression listing the concerned tables and the associated WHERE clause => [FromExpression]

- and, optionally, one or more sort keys.

The target is to produce an SQL statement which will look like:

```
        SELECT
                [RowheadingAlias(0)],
                [RowheadingAlias(1)],
                ...
                SUM( CASE [ColHeadingAlias] WHEN 'ColValue0' THEN [Data] ELSE 0 END ) As [Col
                SUM( CASE [ColHeadingAlias] WHEN 'ColValue1' THEN [Data] ELSE 0 END ) As [Col
                SUM( CASE [ColHeadingAlias] WHEN 'ColValue2' THEN [Data] ELSE 0 END ) As [Col
                ...
                SUM( [Data] ) As [All]
        FROM
                (SELECT RowHeading(0),
                        RowHeading(1),
                        ...
                        ColHeading,
                        Aggregate As [Data]
                FROM FromExpression
                GROUP BY RowHeadingAlias(0),RowHeadingAlias(1), ColHeadingAlias
                )
        GROUP BY RowHeadingAlias(0),RowHeadingAlias(1)
        ORDER BY [All] | OrderBy
```

The resulting SQL statement could be afterwards:

- displayed with the **OpenSQL** action
- browed as a **Recordset**
- stored as a new Query with **CreateQueryDef**

Examples:

```
Dim sSql As String
        sSql = MakeCrossTab( _
        "[FirstName] || ' ' || [LastName] As [Name]" _
                , "YEAR([OrderDate]) || 'Q' || QUARTER([OrderDate]) As [Quarter]" _
                , "Count(*)" _
                , "[Employees] INNER JOIN [Orders] ON ([Employees].[EmployeeID]=[Orders].[Emp
                , "DESC" _
                )
        OpenSQL(sSql, dbSQLPassThrough)
```

will produce:

Name	1996Q3	1996Q4	1997Q1	1997Q2	1997Q3	1997Q4	1998Q1	1998Q2	All
Margaret Peacock	15	16	18	18	22	23	32	12	156
Janet Leverling	7	11	19	16	10	26	28	10	127
Nancy Davolio	11	15	10	10	18	17	29	13	123
Laura Callahan	11	8	19	9	14	12	19	12	104
Andrew Fuller	8	8	9	10	11	11	19	20	96
Robert King	3	8	6	12	13	5	14	11	72
Michael Suyama	9	6	6	8	5	14	14	5	67
Anne Dodsworth	2	3	2	6	4	7	15	4	43
Steven Buchanan	4	7	3	4	6	5	12	1	42

Similarly:

```
Dim sSql As String
Const dbSQLPassThrough = 64
        sSql = MakeCrossTab( _
                "[Customers].[CompanyName] As [Customer], [Products].[ProductName] AS [Name]"
                , "YEAR([OrderDate]) || 'Q' || QUARTER([OrderDate]) As [Quarter]" _
                , "SUM([Order Details].[UnitPrice]*[Quantity]*(1-[Discount]))" _
                , "[Order Details], [Products], [Orders], [Customers] " _
                        & "WHERE [Order Details].[ProductID] = [Products].[ProductID] " _
                        & "AND [Order Details].[OrderID] = [Orders].[OrderID] " _
                        & "AND [Customers].[CustomerID] = [Orders].[CustomerID] " _
                        & "AND YEAR([Orders].[OrderDate]) = 1997" _
                , "[Customer]" _
                )
        OpenSQL(sSql, dbSQLPassThrough)
```

will produce next result:

Customer	Name	1997Q1	1997Q2	1997Q3	1997Q4	All
Alfreds Futterkiste	Lakkalikööri	0	0	0	270	270
Alfreds Futterkiste	Aniseed Syrup	0	0	0	60	60
Alfreds Futterkiste	Vegie-spread	0	0	0	878	878
Alfreds Futterkiste	Spegesild	0	0	18	0	18
Alfreds Futterkiste	Chartreuse verte	0	0	283,5	0	283,5
Alfreds Futterkiste	Rössle Sauerkraut	0	0	513	0	513
Ana Trujillo Emparedados y helados	Mascarpone Fabioli	0	0	0	320	320
Ana Trujillo Emparedados y helados	Camembert Pierrot	0	0	340	0	340
Ana Trujillo Emparedados y helados	Singaporean Hokkien Fried Mee	0	0	70	0	70
etc ...						

Note that in the context of HSQLDB as database management system:

- the readability of the arguments is strongly improved by using the square brackets [] as delimitors of table- and fieldnames instead of double quotes;
- either the double quotes or the square brackets are MANDATORY;
- the table- and fieldnames are case-sensitive.

Code

Next function will do the job:

```
Public Function MakeCrossTab( _
                Byval psRowHeading As String _
                , Byval psColHeading As String _
                , Byval psAggregate As String _
                , Byval psFromExpression As String _
                , Byval psSortBy As String _
                ) As String

Dim sQuery As String, sSubQuery As String, vRowHeading() As Variant, sGroupBy As String, sSort
Dim sDataQuery As String, oData As Object, oField As Object, sCase As String, sValue As String
Dim i As Integer
        vRowHeading() = Split(psRowHeading, ",")
        If UBound(vRowHeading) < 0 Then Exit Function

        '       SUBQUERY
        sSubQuery = "SELECT " & vRowHeading(0)
        For i = 1 To UBound(vRowHeading)
                sSubQuery = sSubQuery & "," & vRowheading(i)
        Next i
        sSubQuery = sSubQuery & ", " & psColHeading & ", " & psAggregate & " AS [Data] FROM "
        sGroupBy = AliasOf(vRowHeading(0))
        For i = 1 To UBound(vRowHeading)
                sGroupBy = sGroupBy & ", " & AliasOf(vRowHeading(i))
        Next i
        sSubQuery = sSubQuery & sGroupBy & "," & AliasOf(psColHeading)

        '       MAIN QUERY
        '       Identify all distinct column headings
        sDataQuery = "SELECT DISTINCT " & psColHeading & " FROM " & psFromExpression & " ORDE
        Set oData = CurrentDb().OpenRecordset(sDataQuery,, dbSQLPassThrough, dbReadOnly)
        Set oField = oData.Fields(0)
        '       Build CASE sentences
        sCase = ""
        For i = 0 To UBound(vRowHeading)
                scase = sCase & AliasOf(vRowHeading(i)) & ", "
        Next i
        With oData      '       Recordset
                Do While Not .EOF
                        sValue = CStr(oField.Value)             '       Force string
                        sCase = sCase & "SUM( CASE " & AliasOf(psColHeading) & " WHEN '" & sV
                        .MoveNext
                Loop
                .mClose()
        End With
        sCase = sCase & "SUM( [Data] ) As [All]"
        '       Final query
        Select Case UCase(psSortBy)
                Case "", "ASC"          :               sSortBy = "ORDER BY [All] ASC"
                Case "DESC"             :               sSortBy = "ORDER BY [All] DESC"
                Case Else               :               sSortBy = "ORDER BY " & psSortBy
        End Select
        sQuery = "SELECT " & sCase & " FROM (" & sSubQuery & ") GROUP BY " & sGroupBy & sSort

        '       Store SQL
        MakeCrossTab = sQuery

End Function
```

It calls next small function:

```
Function AliasOf(ByVal psString As String) As String
Dim iPos As Integer
        iPos = InStr(psString, " AS ")
        If iPos > 0 Then AliasOf = Right(psString, Len(psString) - iPos - 3) Else AliasOf = ps
End Function
```

See also

CreateQueryDef
Execute
OpenSQL
Recordset

Refer to ...

Basic module
CrossTab

Bookmark this page » » CrossTabQuery