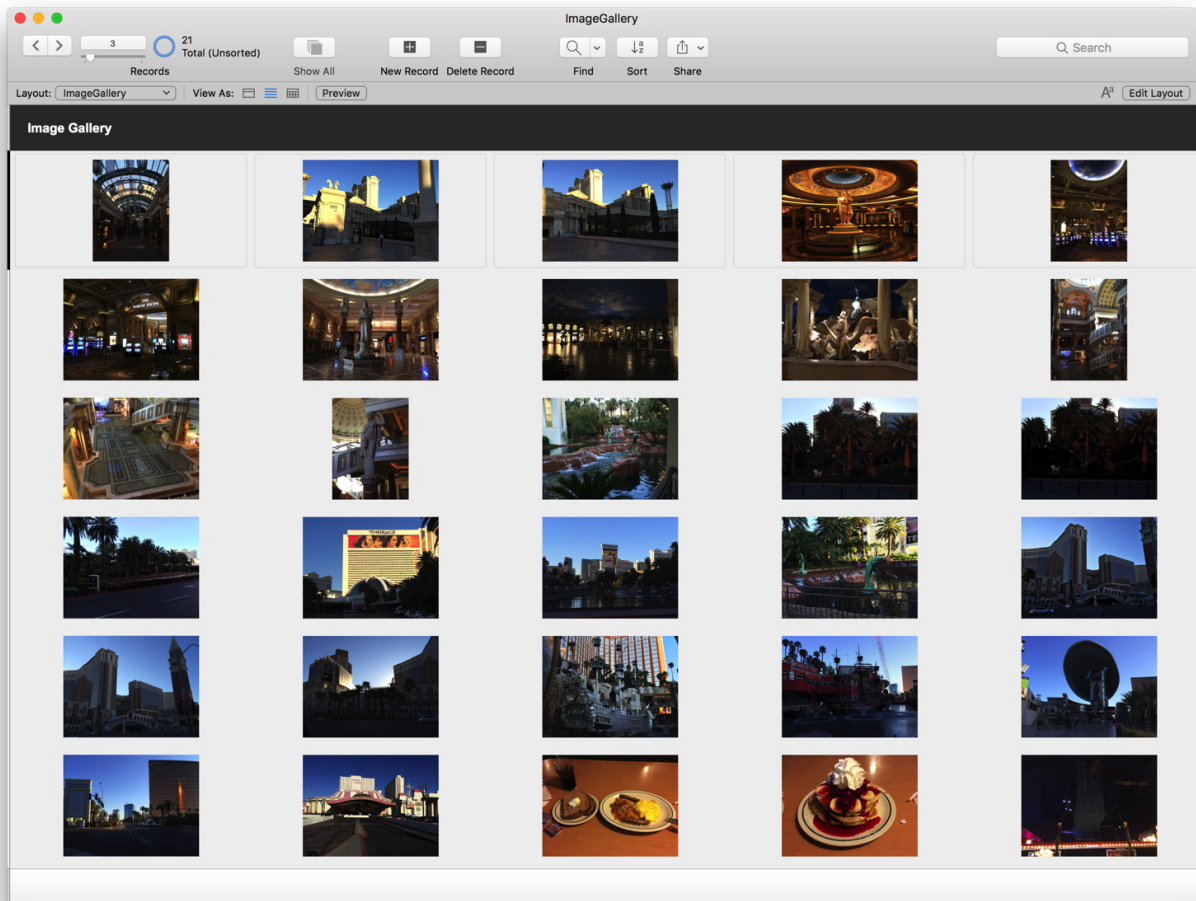


Image Gallery in FileMaker using SQL

For some weeks I had this idea on how to make a gallery for FileMaker, which shows images from a table, but in a grid style. As you see we still have rows, which show containers, but the assignment of which image shows in which container is defined in formulas:



On the layout, we put in five container fields all defined by a formula, which picks the right field via a custom function and using SQL. But let's start with a helping function **ImagesCountPerRow**:

```
Min( 5; Max ( 1; Round ( Get ( WindowWidth ) / 266 ; 0)))
```

This returns how many fields are needed for a given window width from 1 to 5. In the un-stored calculation of the formulas, we use a formula like this:

```

Let (
[
x = 1;
w = ImagesCountPerRow
];
If(w >= x ; GetImage ( Get ( RecordNumber ) ; x ; w); "")
)

```

This checks how many fields are needed and calls `GetImage` only, if the current container index (x) is needed. So we only use the number of containers in a row which are visible.

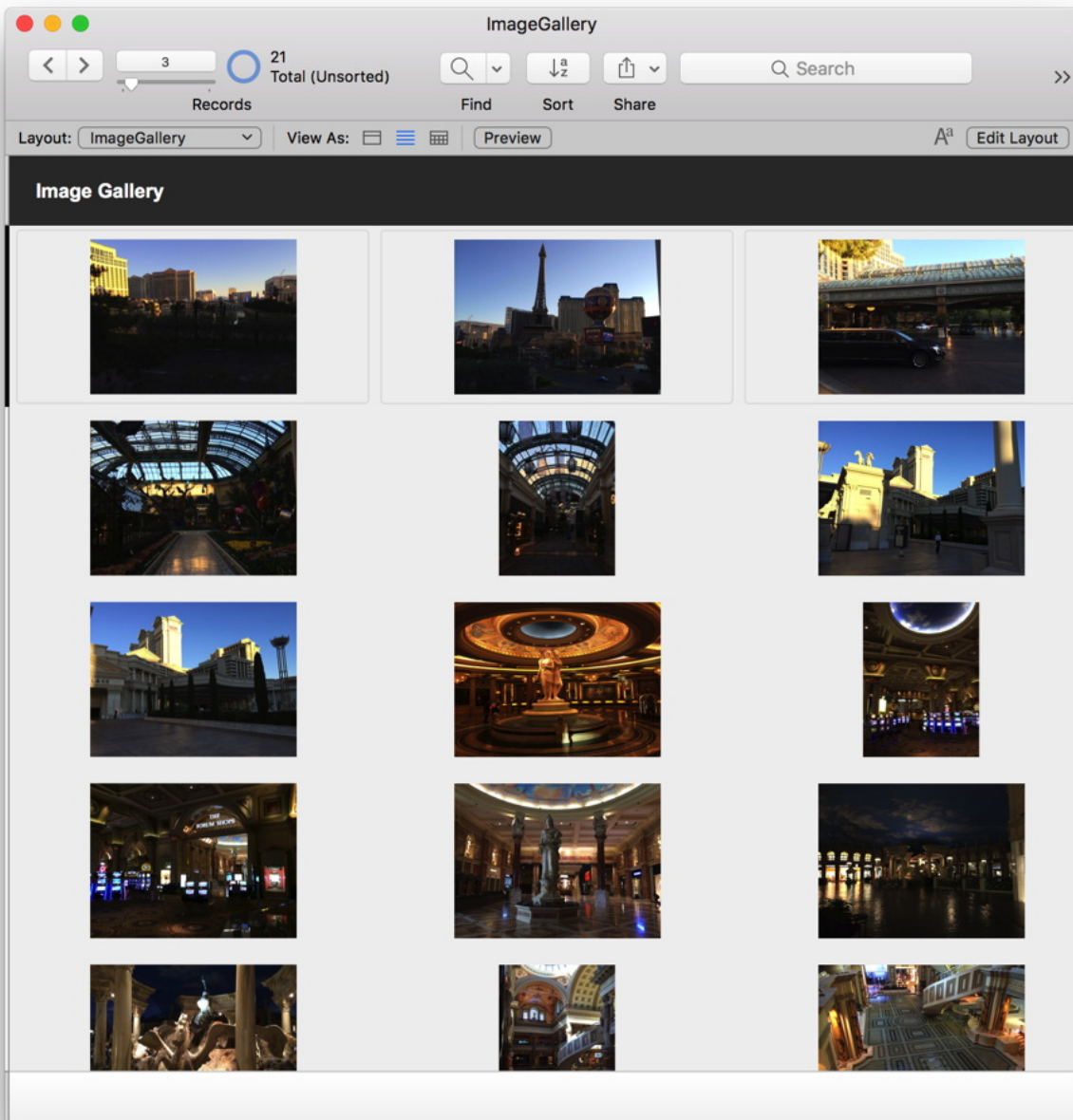
The `GetImage` function takes the number of the record, the number of the field and the field count per row. We calculate the index of the container value we need and fetch it with SQL:

function **GetImage**(nRecord; nField; nFieldCount)

```

Let ( [
n = (nRecord-1) * nFieldCount + nField-1;
sql = MBS( "FM.SQL.Execute"; ""; "SELECT Image FROM Images
OFFSET " & n & " ROWS FETCH FIRST 1 ROWS ONLY" );
error1 = MBS("IsError");
count = If(error1 = 0; MBS( "FM.SQL.RowCount"; sql; n; 0 ); "");
error2 = MBS("IsError");
value = If(error1 = 0 and 0 < count; MBS( "FM.SQL.Field"; sql; 0; 0 );
"" );
error3 = MBS("IsError");
r = If(error1 = 0; MBS( "FM.SQL.Release"; sql ); "")
]
; If(error3 = 0 and error2 = 0 and error1 = 0; value; "" )

```



As you see we use If() functions with error status to make sure we only call further functions if the previous function worked. So [FM.SQL.Execute](#) runs the SQL to find just the one container value we need using the offset command to skip some values before and only fetch one value. If we got a row, we fetch the field value of the first field. On the end we return the value or "" if not found.

On the layout, we can add a trigger for a script to refresh using the Refresh Window script step. This will cause the images to reformat and adjust to new window width.

The example is available with next plugin prerelease or today if you ask kindly.

In the example you also find the script to pick a folder via dialog and import all image files.