

Unlocking the Power of Database Analysis with MBS FileMaker Plugin Functions

When managing and optimizing FileMaker databases, understanding the details of your data is crucial. Whether you're looking to optimize performance, audit data quality, or get insights into database structure, having precise statistical information about your database can be invaluable. The [MBS FileMaker Plugin](#) offers powerful functions for querying detailed statistics on fields, records, and entire tables. In this blog post, we'll explore three key functions — [FM.FieldStatistics](#), [FM.RecordStatistics](#), and [FM.TableStatistics](#) — and how to use them to gain deep insights into your FileMaker databases.

1. FM.FieldStatistics: Dive into Field Details

The `FM.FieldStatistics` function allows you to query statistics on a specific field within a table. This can be useful for understanding the distribution of data within a particular field and for identifying any anomalies or patterns.

Function Syntax:

```
MBS( "FM.FieldStatistics"; FileName; TableName; FieldName )
```

Parameters:

- **FileName:** The target database name. Leave empty to query across all databases.
- **TableName:** The name of the table where the field resides.
- **FieldName:** The name of the field you want to analyze.

Example Usage:

```
MBS( "FM.FieldStatistics"; "Contacts.fmp12"; "Contacts";  
"PrimaryKey" )
```

Result: The function returns a JSON object containing detailed statistics about the specified field, such as the number of records that contain data, average length of entries, and more.

2. FM.RecordStatistics: Analyze Individual Records

The `FM.RecordStatistics` function provides insights into a specific record within a table. This can help you understand the distribution and quality of data at the record level.

Function Syntax:

```
MBS( "FM.RecordStatistics"; FileName; TableName;  
PrimaryKeyName; PrimaryKeyValue )
```

Parameters:

- **FileName:** The target database name. Leave empty to query across all databases.
- **TableName:** The name of the table containing the record.
- **PrimaryKeyName:** The name of the primary key field used to identify the record.
- **PrimaryKeyValue:** The value of the primary key for the record you want to analyze.

Example Usage:

```
MBS( "FM.RecordStatistics"; "Contacts.fmp12"; "Contacts";  
"PrimaryKey"; "12345" )
```

Result: The JSON output includes detailed statistics about the specified record, such as the data types of fields, the amount of text, and container sizes.

3. FM.TableStatistics: Get the Big Picture

The `FM.TableStatistics` function provides a comprehensive overview of an entire table, summarizing data types, field usage, and container sizes across all records.

Function Syntax:

```
MBS( "FM.TableStatistics"; FileName; TableName )
```

Parameters:

- **FileName:** The target database name. Leave empty to query across all databases.
- **TableName:** The name of the table you want to analyze.

Example Usage:

```
MBS( "FM.TableStatistics"; "Contacts.fmp12"; "Contacts" )
```

Result Example:

```
{
```

```
"FileName": "",
"TableName": "Contacts",
"RecordCount": 4,
"MinRowID": 1,
"MaxRowID": 4,
"ColumnCount": 35,
"CountText": 20,
"CountEmpty": 107,
"CountNumber": 5,
"CountDate": 2,
"CountTime": 1,
"CountTimestamp": 1,
"CountContainer": 4,
"CountExternalContainer": 1,
"ContainerByteSize": 1162565,
"TextLengths": 132,
"AllTextLengths": 365,
"CountDPI_": 2,
"SizeDPI_": 8,
"CountFNAM": 3,
"SizeFNAM": 90,
"CountSIZE": 2,
"SizeSIZE": 8,
"CountPNGf": 1,
"SizePNGf": 727569,
"CountFILE": 1,
"SizeFILE": 390,
"CountJPEG": 1,
"SizeJPEG": 434606
```

```
}
```

Explanation of Results:

- **FileName:** Name of the file queried.
- **TableName:** Name of the table queried.
- **RecordCount:** Total number of records in the table.
- **MinRowID/MaxRowID:** The range of record IDs.
- **ColumnCount:** Number of columns in the table.
- **CountText/CountEmpty/CountNumber/CountDate/CountTime/CountTimestamp:** Counts of different types of fields.
- **CountContainer/CountExternalContainer:** Number of container fields, including external containers.
- **ContainerByteSize:** Total byte size of internal containers.
- **TextLengths/AllTextLengths:** Lengths of text data in fields.

Practical Example: Analyzing Table Sizes

If you want to analyze the sizes of all tables in a FileMaker database, you can use the `FM.TableStatistics` function in a script to loop over all tables and store their statistics. Here's a sample script:

```
Go to Layout [ "TableStatistics" (TableStatistics) ;
Animation: None ]
#
Set Variable [ $FileName ; Value: Get(FileName) ]
Set Variable [ $TableNames ; Value:
MBS( "FM.QueryBaseTableNames"; $FileName) ]
#
# Loop through each table
Set Variable [ $count ; Value: ValueCount($TableNames) ]
Set Variable [ $index ; Value: 1 ]
#
Loop [ Flush: Defer ]
    Set Variable [ $TableName ; Value: GetValue($TableNames;
$index) ]
    Set Variable [ $JSON ; Value: MBS( "FM.TableStatistics";
$FileName; $TableName ) ]
    #
    New Record/Request
    Set Field [ TableStatistics::JSON ; $JSON ]
    Set Field [ TableStatistics::Name ; $TableName ]
    Commit Records/Requests [ With dialog: Off ]
    #
    Set Variable [ $index ; Value: $index + 1 ]
    Exit Loop If [ $index > $count ]
End Loop
```

This script retrieves statistics for all tables in the specified database and stores the results in a `TableStatistics` table.

Conclusion

Understanding the structure and content of your FileMaker databases is essential for effective management and optimization. The MBS FileMaker Plugin's `FM.FieldStatistics`, `FM.RecordStatistics`, and `FM.TableStatistics` functions provide powerful tools for querying detailed statistics at the field, record, and table levels. By leveraging these functions, you can gain valuable insights into your data, identify potential issues, and make informed decisions to enhance your database's performance and reliability.

Explore these functions today and see how they can help you get the most out of your FileMaker databases!