Using ExecuteSQL with Parameters is important

Today we got a problem with SQL. It's not the usual SQL injection worries, but more the convenience of building SQL statements containing values, which causes problem.

We see a lot of SQL queries to lookup values, but people love to include the

```
MBS( "FM.ExecuteFileSQL"; Get(FileName); "SELECT Name FROM Contracts WHERE ID=" & $ID )
```

Now if ID is a text, but $ID contains the number 1234, this causes an error:

[MBS] ERROR: FQL0018/(1:32): An expression contains incompatible data types.
You would need to use quotes at least:

```
MBS( "FM.ExecuteFileSQL"; Get(FileName); "SELECT Name FROM Contracts WHERE ID= '" & $ID & "'" )
```

Although you don't see the double quotes having single quotes easily here. This may work, until someone includes a single quote in that variable.

When you use parameters in the query, you can avoid this:

```
MBS( "FM.ExecuteFileSQL"; Get(FileName); "SELECT Name FROM Contracts WHERE ID = ?"; 9; 13; $ID )
```

You can pass parameter as text or number even if it does not the match the field type. FileMaker will converted the data type automatically to match.

Next, lets assume you have a variable containing a number and you make a query

```
Set Variable [ $x ; Value: 123.45 ]
Show Custom Dialog [ MBS( "FM.ExecuteFileSQL"; Get(FillName); "SELECT Name, Price FROM Items WHERE Price= " & $x; 9; 13 ) ]
```

If $x is a number, it will be converted by FileMake to a text. Or it is a text already with some user entered number. Now if the field is empty, you get a syntax error, because your SQL ends in a = character. But you may have noted that the number is passed directly here without parameter. So a German user typing "123,45" with comma as decimal separator will cause a SQL error. SQL always uses dot as decimal separator, so the query fails in Germany with comma, but works in USA.

```
Show Custom Dialog [ MBS( "FM.ExecuteFileSQL"; Get(FillName); "SELECT Name, Price FROM Items WHERE Price = ?"; 9; 13; $x ) ]
```

Using parameters helps you to avoid problems here and this works for any locale. If needed, you can use GetAsNumber() to explicit convert text to number and use localized decimal separators to get a numeric value.

So please use parameters for SQL statements.