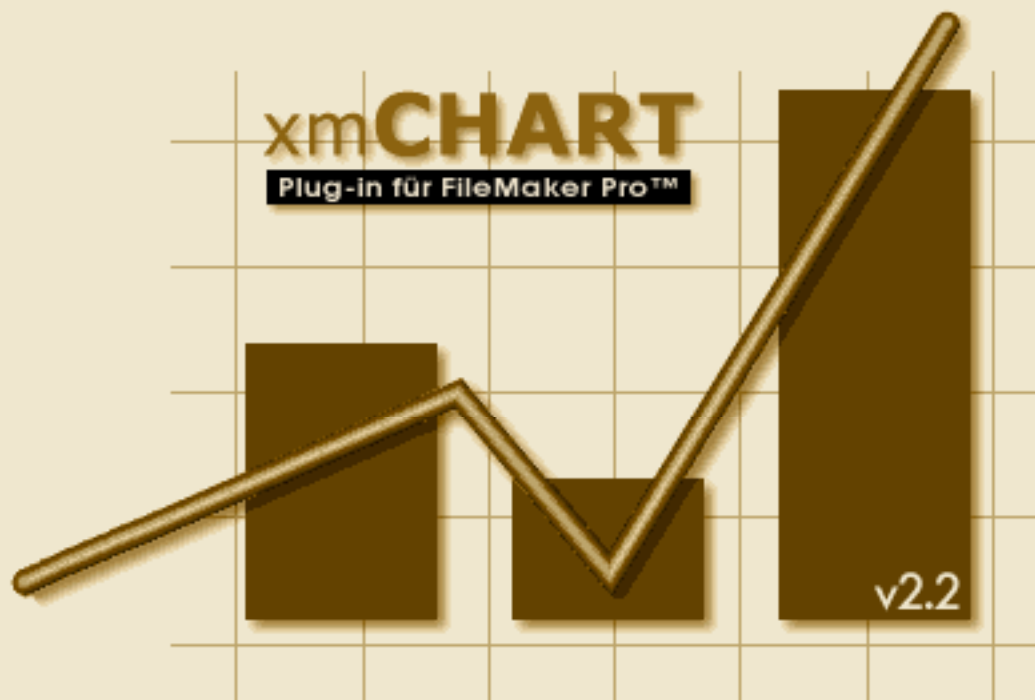


# Beispiele



*X2max*  
www.x2max.com

FileMaker Pro is a registered trademark of FileMaker, Inc.  
© 1997-2002 by X2max Software. All rights reserved.

# Inhaltsverzeichnis

## Beispiele

<b>Einführung</b>	3
<b>Beispiel 1</b>	5
<b>Beispiel 2</b>	9
<b>Beispiel 3</b>	16
<b>Beispiel 4</b>	22
<b>Beispiel 5</b>	29
<b>Beispiel 6</b>	38

# Einführung

Nachfolgend werden anhand einiger ausgewählter Beispiele die praktischen Einsatzmöglichkeiten von xmCHART aufgezeigt. Es wurden dazu bewußt umfangreichere Beispiele ausgewählt, um einerseits die Vielseitigkeit und Mächtigkeit von xmCHART zu verdeutlichen und andererseits das wichtige Zusammenspiel zwischen FileMaker Pro Scripts und xmCHART zu erläutern. Mittels FileMaker Pro Scripts ist es möglich, die Gestaltung von Diagrammen vollkommen zu automatisieren, so dass "auf Knopfdruck" aussagekräftige Diagramme erstellt werden können. Alle hier angeführten Beispiele stehen als voll funktionsfähige, nicht durch Passwort geschützte FileMaker Pro Dateien im .FP3 und .FP5 Format zur Verfügung.

## **Einbindung von xmCHART in FileMaker Pro Dateien:**

Im allgemeinen sind dazu folgende FileMaker Pro Felder und Scripts notwendig:

- *FileMaker Pro Felder:*

Zweckmäßigerweise sind das Variablenfelder, welche durch das Prefix "g" (global) gekennzeichnet werden. In den nachfolgenden Beispielen werden stets die folgenden drei Variablenfelder verwendet:

*gFunktionen:* enthält alle für xmCHART benötigten Funktionsaufrufe.

*gDiagramm:* übernimmt das von xmCHART erstellte Diagramm.

*gFehler:* enthält Informationen über eventuell im Zuge der Diagrammerstellung aufgetretene Fehler.

Um die praktische Handhabung zu erleichtern, können zusätzlich sog. Platzhalter verwendet werden. Dazu wird eine weitere Textvariable, zum Beispiel mit dem Namen *gFunktionenMitPlatzhalter* eingeführt. Diese enthält gleich wie die Textvariable *gFunktionen* alle für xmCHART benötigten Funktionsaufrufe, nur anstelle der "variablen" Daten — das heißt den Daten welche üblicherweise aus Datenbankfeldern übernommen werden — werden Platzhalter angeführt. Zweckmäßig erweisen sich Platzhalter, zum Beispiel für die Diagrammdaten, Achsenbeschriftungen oder Legendentexte.

Platzhalter werden üblicherweise durch spitze Klammern gekennzeichnet "<>". Zum Beispiel:

```
ChartData(<gDaten>)  
AxisMajorTickLabelTexts(x;<gBeschriftungen>)
```

Für jeden Platzhalter wird eine eigenen Textvariable angelegt, das heißt zum Beispiel für <gDaten> eine Textvariable *gDaten*, für <gBeschriftungen> eine Textvariable *gBeschriftungen* usw. Der Inhalt dieser Textvariablen wird in der Regel durch ein FileMaker Pro Script aufbereitet. Abschließend werden alle Platzhalter mittels eines weiteren FileMaker Pro Scripts durch die tatsächlichen Werte ersetzt. Die Arbeitsweise mit Platzhaltern wird in den ersten drei Beispielen erläutert.

- *FileMaker Pro Scripts:*

Zur automatischen Diagrammerstellung wird mindestens ein FileMaker Pro Script benötigt, welches meist, um die Übersichtlichkeit zu erhöhen, in mehrere Teilscripts unterteilt wird. Die Namen der Teilscripts beginnen in allen Beispielen mit einem Unterstrich, zum Beispiel *\_PlatzhalterAustauschen*.

Um sich einfach und rasch zurechtzufinden, werden die oben verwendeten Feldbezeichnungen in allen Beispielen beibehalten.

In den mitgelieferten Beispieldateien wurde versucht, die wesentlichen Schritte, ausgehend von der Erfassung der darzustellenden Werte, bis hin zum fertigen Diagramm, übersichtlich und nachvollziehbar darzustellen. In einer professionellen Datenbanklösung werden in der Regel aber, weder die zur Erstellung der Funktionsaufrufe notwendigen Scripts, noch die Funktionsaufrufe selbst, für den Benutzer sichtbar sein, so dass ein Diagramm meist auf "Knopfdruck" erstellt werden kann. Es bleibt dem Entwickler einer Datenbanklösung überlassen, welche Optionen zur Gestaltung eines Diagramms er dem Benutzer zugänglich macht.

Da das FileMaker Pro Plug-in API nur die Ein- und Ausgabe eines Textstrings erlaubt, gibt es keine plattformübergreifende Möglichkeit, direkt von xmCHART aus auf Datenbankfelder zuzugreifen. Das heißt, sämtliche zur Erstellung eines Diagramms notwendigen Daten müssen mittels der Funktion `ChartData()` an xmCHART übergeben werden. Deshalb nimmt üblicherweise das zum Aufbau der Funktion `ChartData()` benötigte Script eine zentrale Stellung ein. Mögliche Vorgangsweisen dazu, werden anhand unterschiedlicher Problemstellungen in den nachfolgend angeführten Beispielen gezeigt. Die restlichen für xmCHART benötigten Funktionsaufrufe können meist durch einfache Textverknüpfungsfunktionen per Script hinzugefügt werden.

# Beispiel 1

## Aufgabenstellung:

Im ersten Beispiel wird eine Serie von Werten in Form eines Tortendiagramms dargestellt (Abb. 8). Dabei sollen die einzelnen Tortensegmente sowohl mit dem Namen der Kategorie als auch dem tatsächlichen Wert und dem prozentuellen Anteil beschriftet werden. Die dazugehörige FileMaker Pro Datei ist xmBEISPIEL1.FP3 bzw. xmBEISPIEL1.FP5.

## Lösung:

### FileMaker Pro Felder: (Abb. 1)

Neben den in der Einleitung beschriebenen Variablenfeldern *gFunktionen*, *gDiagramm*, *gFehler* und *gFunktionenMitPlatzhalter* werden noch die folgenden Felder benötigt:

- Das Textfeld *Name*, welches den Namen der Kategorie enthält.
- Das Zahlenfeld *Wert*, welches den Wert der Kategorie enthält.
- Die Textvariablen *gDaten*, *gBeschriftungen* und *gFormat*. Diese korrespondieren mit den entsprechenden Platzhaltern *<gDaten>*, *<gBeschriftungen>* und *<gFormat>* und dienen zum Speichern der aktuellen Werte.
- Optional die Textvariablen *gIstAktiv* und *gVersion*. Falls xmCHART aktiv ist, enthält *gIstAktiv* den Wert "1", ansonst ist *gIstAktiv* leer. *gVersion* enthält die genaue Bezeichnung der momentan aktiven xmCHART Version, zum Beispiel "xmCHART 2.2".

<u>Feldname</u>	<u>Feldtyp</u>	<u>Optionen</u>
[1]	Formel	Indiziert, = 1
Name	Text	
Wert	Zahl	
gFunktionenMitPlatzhal...	Variable	Text
gFunktionen	Variable	Text
gDiagramm	Variable	Medien
gFehler	Variable	Text
gVersion	Variable	Text
gIstAktiv	Variable	Text
gDaten	Variable	Text
gBeschriftungen	Variable	Text
gFormat	Variable	Text

Abb. 1

**FileMaker Pro Scripts:**

- FileMaker Pro Script *DiagrammErstellen*: (Abb. 2)

Das Script *DiagrammErstellen* erzeugt das gewünschte Diagramm und ist aus mehreren Teilscrippts aufgebaut. Zuerst kann optional abgefragt werden, ob die passende FileMaker Pro Version vorhanden ist (Script: *\_IstFMVersion4.0+*) und ob xmCHART aktiv ist (Script: *\_IstPluginAktiv*).

```
Fenster fixieren
Script ausführen [Teilscrippts, '_IstFMVersion4.0+']
Script ausführen [Teilscrippts, '_IstPluginAktiv']
Script ausführen [Teilscrippts, '_gDatenErstellen']
Script ausführen [Teilscrippts, '_gBeschriftungenErstellen']
Script ausführen [Teilscrippts, '_PlatzhalterAustauschen']
Script ausführen [Teilscrippts, '_DiagrammZeichnen']
```

Abb. 2

(1) *\_gDatenErstellen*: (Abb. 3)

Das Teilsript *\_gDatenErstellen* entnimmt aus der Datenbank, genauer gesagt aus dem Zahlenfeld *Wert*, die benötigten Daten und speichert diese in Form einer Zeichenkette — die einzelnen Zahlenwerte durch Leerzeichen getrennt — in *gDaten*. Dies geschieht mittels einer Schleife über alle Datensätze.

```
Feld angeben ['gDaten', '']
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Feld angeben ['gDaten', 'gDaten & Wert & " "]
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
```

Abb. 3

(2) *\_gBeschriftungenErstellen*: (Abb. 4)

Das Teilsript *\_gBeschriftungenErstellen* entnimmt aus der Datenbank die Kategorienamen und speichert diese in Form einer Zeichenkette in *gBeschriftungen*. Die einzelnen Namen sind dabei unter Hochkomma zu setzen und durch Strichpunkte zu trennen. Zusätzlich wird zu jedem Namen *gFormat* hinzugefügt. Die Textvariable *gFormat* enthält zwei Formatanweisungen zur Darstellung der Zahlenwerte. Dabei legt die erste Formatanweisung die Ausgabe der absoluten Werte fest, die zweite Formatanweisung die Ausgabe der prozentuellen Werte. Formatanweisungen werden ausführlich in *xmReferenz* behandelt, die verschiedenen Optionen zur Beschriftung von Tortendiagrammen in *xmTutorial*, Abschnitt *Stile*. Zu beachten ist, dass bei der Eingabe von Texten oder Schriftnamen die erforderlichen Hochkommas im Script Editor immer doppelt eingegeben werden müssen. Zum Beispiel:

```
gFunktionen & "TitleText("Diagramm 1")"
```

```

Feld angeben ['gBeschriftungen', '']
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Feld angeben ['gBeschriftungen', 'gBeschriftungen & "" & Name & gFormat & "";', '']
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
Feld angeben ['gBeschriftungen', 'Links(gBeschriftungen, Länge(gBeschriftungen)-1)']

```

Abb. 4

### (3) *\_PlatzhalterAustauschen*: (Abb. 5)

Das Teilsript *\_PlatzhalterAustauschen* ersetzt die Platzhalter in *gFunktionenMitPlatzhalter* (Abb. 6) durch die tatsächlichen Werte.

```

Feld angeben ['gFunktionen', 'Austauschen(gFunktionenMitPlatzhalter, "<gDaten>", gDaten)']
Feld angeben ['gFunktionen', 'Austauschen(gFunktionen, "<gBeschriftungen>", gBeschriftungen)']

```

Abb. 5

```

OpenDrawing(400;300)
  ChartData(<gDaten>)
  PieChart(oval+label+shadow)
  BorderStyle(all;0) // Ränder ausblenden
  PieChartAuxLines() // Beschriftungen mit Hilfslinien
  LabelTexts(1;<gBeschriftungen>)
  LabelStyle(all;;;center) // Beschriftungen mittig
  LabelBackground(all;lightYellow;1;;2)
CloseDrawing()

```

Abb. 6

### (4) *\_DiagrammZeichnen*: (Abb. 7)

Im Teilsript *\_DiagrammZeichnen* werden alle in *gFunktionen* zusammengefaßten Befehle durch die externe Funktion *xmCH-DrawChart()* an *xmCHART* übergeben. Tritt bei der Erstellung des Diagramms kein Fehler auf, so wird abschließend das Diagramm aus der Zwischenablage in das Medienfeld *gDiagramm* kopiert.

```

Feld angeben ['gFehler', 'Extern("xmCH-DrawChart", gFunktionen)']
Wenn ['IstLeer(gFehler)']
  Einsetzen ['gDiagramm']
Ende-Wenn

```

Abb. 7

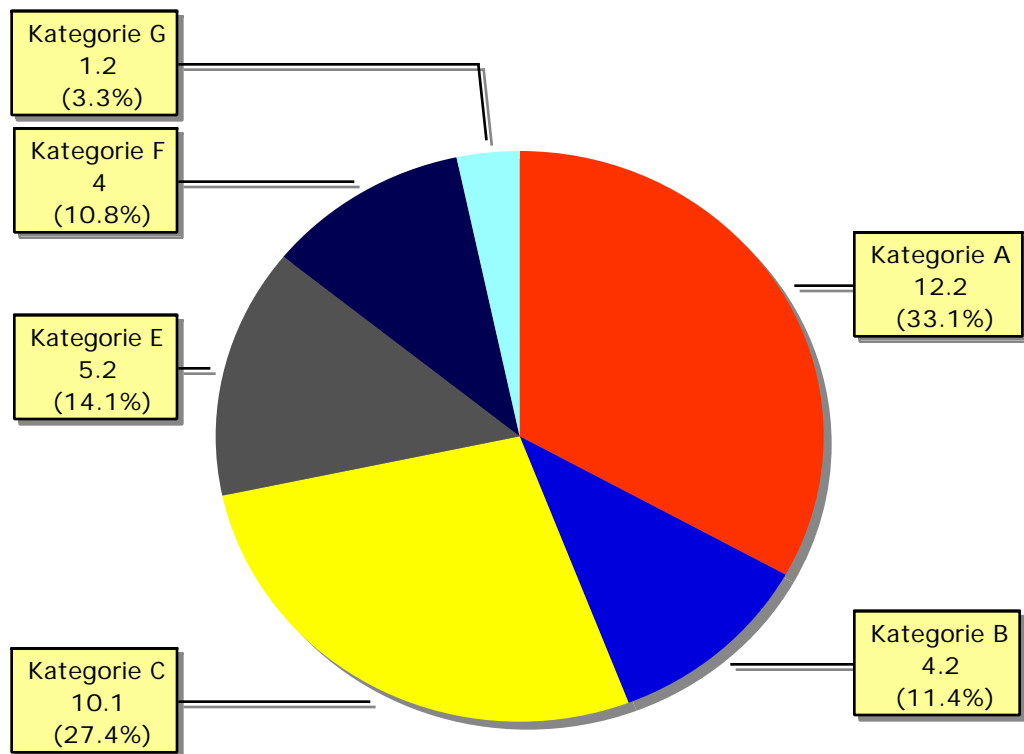
**Ergebnis:**

Abb. 8

**Varianten:**

Die Beschriftung kann auch innerhalb oder aufgeteilt, sowohl innerhalb als auch außerhalb des Diagramms platziert werden. Zusätzlich kann ein Text auch in der Mitte des Tortendiagramms platziert werden. Details mit Beispielen sind in *xmTutorial*, Abschnitt *Diagramme* zu finden.



# Beispiel 2

## Aufgabenstellung:

Im zweiten Beispiel sollen zeitabhängige Daten in Form eines sog. Gantt-Diagramms dargestellt werden (Abb. 18). Konkret sollen die unterschiedlichen Zeiträume für eine Serie von Aufgaben entlang einer horizontalen Zeitachse aufgetragen werden. Dabei soll abhängig vom darzustellenden Zeitraum — Tage, Wochen, Monate oder Jahre — automatisch eine passende Skalierung gefunden werden. Die verschiedenen Skalierungen können durch Ändern der Zeiträume (*von-bis*) in der dazugehörigen FileMaker Pro Datei xmBEISPIEL2.FP3 bzw. xmBEISPIEL2.FP5 ausprobiert werden.

## Lösung:

### FileMaker Pro Felder: (Abb. 9)

Neben den in der Einleitung beschriebenen Variablenfeldern *gFunktionen*, *gDiagramm*, *gFehler* und *gFunktionenMitPlatzhalter* werden noch die folgenden Felder benötigt:

- Das Textfeld *Aufgabe*, welches den Namen der Aufgabe enthält.
- Die Datumsfelder *VonZeitraum1*, *BisZeitraum1*, *VonZeitraum2* und *BisZeitraum2*, wodurch für jede Aufgabe max. zwei Teilabschnitte festgelegt werden können. In diesem konkreten Beispiel wurde die Unterteilung einer Aufgabe auf zwei Teile beschränkt, tatsächlich kann in xmCHART aber jede Aufgabe in beliebig viele Abschnitte unterteilt werden.
- Die Textvariablen *gDaten*, *gXSkalierung*, *gXAchsenbeschriftungen* und *gYAchsenbeschriftungen* korrespondieren mit den entsprechenden Platzhaltern *<gDaten>*, *<gXSkalierung>*, *<gXAchsenbeschriftungen>* und *<gYAchsenbeschriftungen>* und enthalten die tatsächlichen Werte.
- Die restlichen Datums- und Zahlenvariablen werden zur Berechnung der unterschiedlichen Skalierungszeiträume und Skalenbeschriftungen benötigt.
- Optional die Textvariablen *gIstAktiv* und *gVersion*. Falls xmCHART aktiv ist, enthält *gIstAktiv* den Wert "1", ansonst ist *gIstAktiv* leer. *gVersion* enthält die genaue Bezeichnung der momentan aktiven xmCHART Version, zum Beispiel "xmCHART 2.2".

<u>Feldname</u>	<u>Feldtyp</u>	<u>Optionen</u>
[1]	Formel	Indiziert, = 1
Aufgabe	Text	
VonZeitraum1	Datum	
BisZeitraum1	Datum	
VonZeitraum2	Datum	
BisZeitraum2	Datum	
gFunktionenMitPlatzhal...	Variable	Text
gFunktionen	Variable	Text
gDiagramm	Variable	Medien
gFehler	Variable	Text
gDaten	Variable	Text
gXSkalierung	Variable	Text
gXAchsenbeschriftungen	Variable	Text
gYAchsenbeschriftungen	Variable	Text
gDatum	Variable	Datum
gMinDatum	Variable	Datum
gMaxDatum	Variable	Datum
gAnfangsdatum	Variable	Datum
gEnddatum	Variable	Datum
gAnzahlJahre	Variable	Zahl
gAnzahlMonate	Variable	Zahl
gAnzahlWochen	Variable	Zahl
gAnzahlTage	Variable	Zahl
gXSkalierungsfaktor	Variable	Zahl
gZaehler	Variable	Zahl
gIstAktiv	Variable	Text
gVersion	Variable	Text

Abb. 9

**FileMaker Pro Scripts:**

- FileMaker Pro Script *DiagrammErstellen*: (Abb. 10)

Das Script *DiagrammErstellen* erzeugt das gewünschte Diagramm und ist aus mehreren Teilscripits aufgebaut. Zuerst kann optional abgefragt werden, ob die passende FileMaker Pro Version vorhanden ist (Script: *\_IstFMVersion4.0+*) und ob xmCHART aktiv ist (Script: *\_IstPluginAktiv*).

```

Fenster fixieren
Script ausführen [Teilscripits, '_IstFMVersion4.0+']
Script ausführen [Teilscripits, '_IstPluginAktiv']
Script ausführen [Teilscripits, '_gXSkalierungErstellen']
Script ausführen [Teilscripits, '_gXAchsenbeschriftungErstellen']
Script ausführen [Teilscripits, '_gYAchsenbeschriftungErstellen']
Script ausführen [Teilscripits, '_gDatenErstellen']
Script ausführen [Teilscripits, '_Platzhalter Austauschen']
Script ausführen [Teilscripits, '_DiagrammZeichnen']

```

Abb. 10

(1) *\_gXSkalierungErstellen*: (Abb. 11)

Das Teilsript *\_gXSkalierungErstellen* berechnet zu den vorgegebenen Zeiträumen die passende Skalierung. Dazu wird zuerst das Teilsript *\_\_MinMaxBerechnen* aufgerufen, in welchem alle Datensätze nach dem frühesten und dem spätesten Datum durchsucht werden. Diese werden in *gMinDatum* und *gMaxDatum* gespeichert. Anschließend wird im Teilsript *\_\_ZeitraumBerechnen* aus *gMinDatum* und *gMaxDatum* die Anzahl der Jahre, Monate, Wochen und Tage berechnet. Dabei ist zu beachten, dass für die Berechnung der x-Achsenkala vier unterschiedliche Berechnungsmethoden zur Verfügung stehen. Diese sind abhängig vom darzustellenden Zeitraum, das heißt, ist der darzustellende Zeitraum größer als 36 Monate, dann wird eine Jahresskala verwendet, ergänzt durch eine Feinunterteilung in Quartale (Script: *\_\_Skala: Jahre*). Ist der Zeitraum größer als 14 Wochen und kleiner gleich drei Jahre so wird eine Monatsskala verwendet (Teilsript: *\_\_Skala: Monate*). Ist der darzustellende Zeitraum größer als eine Woche und kleiner gleich 14 Wochen, so wird eine Wochenskala verwendet, ergänzt durch eine Feinunterteilung in Tage (Teilsript: *\_\_Skala: Wochen*). Bei einem Zeitraum kleiner gleich einer Woche wird eine Tagesskala verwendet (Teilsript: *\_\_Skala: Tage*).

```

Script ausführen [Teilscripts, '__MinMaxBerechnen']
Script ausführen [Teilscripts, '__ZeitraumBerechnen']
Wenn ['gAnzahlMonate > 36']
  Script ausführen [Teilscripts, '__Skala: Jahre']
Sonst
  Wenn ['gAnzahlWochen > 14']
    Script ausführen [Teilscripts, '__Skala: Monate']
  Sonst
    Wenn ['gAnzahlWochen > 1']
      Script ausführen [Teilscripts, '__Skala: Wochen']
    Sonst
      Script ausführen [Teilscripts, '__Skala: Tage']
    Ende-Wenn
  Ende-Wenn
Ende-Wenn

```

Abb. 11

(2) *\_gXAchsenbeschriftungenErstellen*: (Abb. 12)

Das Teilsript *\_gXAchsenbeschriftungenErstellen* ist ähnlich aufgebaut wie das Teilsript *\_gXSkalierungErstellen*. Für die vier unterschiedlichen Skalen werden fünf Beschriftungsarten zur Verfügung gestellt. Bei einer Jahresskala, d.h. der Zeitraum ist größer als 36 Monate wird der Beginn eines Jahres beschriftet, z.B. "Jan 99" (Script: *\_\_Texte: Jahre*). Bei einer Monatsskala stehen zwei Beschriftungen zur Verfügung: ist der Zeitraum größer als 12 Monate und kleiner gleich 36 Monate, so werden die Monate nur durch den Anfangsbuchstaben beschriftet (Script:

*\_\_Texte: MonateSchmal*), liegt der Zeitraum zwischen 14 Wochen und 12 Monaten, so werden die Monate durch die ersten 3 Buchstaben beschriftet (Script: *\_\_Texte: MonateBreit*). Die Januarmonate werden zusätzlich noch durch die Jahreszahl ergänzt. Bei einem Zeitraum größer einer Woche und kleiner gleich 14 Wochen werden die Wochenbeginne (Montage) mit Tag und Monat beschriftet (Script: *\_\_Texte: Wochen*). Falls der Zeitraum nur eine Woche umfaßt, so werden die einzelnen Tage durch Wochentag, Tag und Monat beschriftet (Script: *\_\_Texte: Tage*).

Die einzelnen Skalentexte sind dabei unter Hochkomma zu setzen und durch einen Strichpunkt zu trennen. Zu beachten ist, dass die bei der Eingabe von Texten erforderlichen Hochkommas im Script Editor immer doppelt eingegeben werden müssen.

```
Feld angeben ['gXAchsenbeschriftungen', '']
Wenn ['gAnzahlMonate > 36']
  Script ausführen [Teilscrip, '__Texte: Jahre']
Sonst
  Wenn ['gAnzahlMonate > 12']
    Script ausführen [Teilscrip, '__Texte: MonateSchmal']
  Sonst
    Wenn ['gAnzahlWochen > 14']
      Script ausführen [Teilscrip, '__Texte: MonateBreit']
    Sonst
      Wenn ['gAnzahlWochen > 1']
        Script ausführen [Teilscrip, '__Texte: Wochen']
      Sonst
        Script ausführen [Teilscrip, '__Texte: Tage']
      Ende-Wenn
    Ende-Wenn
  Ende-Wenn
Ende-Wenn
```

Abb. 12

### (3) *\_gYAchsenbeschriftungenErstellen*: (Abb. 13)

Das Teilsript *\_gYAchsenbeschriftungenErstellen* entnimmt aus der Datenbank die Beschriftungstexte für die y-Achse und speichert diese in Form einer Zeichenkette in *gYAchsenbeschriftungen*. Die einzelnen Namen sind dabei unter Hochkomma zu setzen und durch einen Strichpunkt zu trennen.

```
Feld angeben ['gYAchsenbeschriftungen', '']
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Feld angeben ['gYAchsenbeschriftungen', 'gYAchsenbeschriftungen & "" & Aufgabe & "" ;']
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
Feld angeben ['gYAchsenbeschriftungen', 'Links(gYAchsenbeschriftungen; Länge(gYAchsenbes...
```

Abb. 13

(4) *\_gDatenErstellen*: (Abb. 14)

Das Teilsript *\_gDatenErstellen* entnimmt mittels einer Schleife über alle Datensätze — ein Datensatz entspricht einer Aufgabe — aus der Datenbank die Werte *VonZeitraum1*, *BisZeitraum1*, *VonZeitraum2* und *BisZeitraum2* und speichert diese in Form einer Zeichenkette in *gDaten*. Dabei wird nur die Anzahl der Tage seit *gAnfangsdatum* gespeichert; zusätzlich wird die Anzahl der Tage durch Multiplizieren mit *gXSkalierungsfaktor* an die verwendete Skalierung angepasst. Da jeder Datensatz einer Datenserie entspricht, muß nach jedem Datensatz noch ein Strichpunkt an die Zeichenkette angefügt werden. Zusätzlich wird noch die Vollständigkeit der Daten überprüft, das heißt, der Zeitraum jeder Aufgabe muß eindeutig definiert sein, andernfalls wird er ignoriert.

```
Feld angeben ['gDaten', '']
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Wenn [' NICHT IstLeer(VonZeitraum1) UND NICHT IstLeer(BisZeitraum1)']
    Feld angeben ['gDaten', 'gDaten & ZahlZuText((VonZeitraum1 - gAnfangsdatum) * gXSkalierung...
    Wenn [' NICHT IstLeer(VonZeitraum2) UND NICHT IstLeer(BisZeitraum2)']
      Feld angeben ['gDaten', 'gDaten & ZahlZuText((VonZeitraum2 - gAnfangsdatum) * gXSkalieru...
    Sonst
      Feld angeben ['gDaten', 'gDaten & ";"]
    Ende-Wenn
  Sonst
    Wenn [' NICHT IstLeer(VonZeitraum2) UND NICHT IstLeer(BisZeitraum2)']
      Feld angeben ['gDaten', 'gDaten & ZahlZuText((VonZeitraum2 - gAnfangsdatum) * gXSkalieru...
    Sonst
      Feld angeben ['gDaten', 'gDaten & "0 0;"]
    Ende-Wenn
  Ende-Wenn
Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
Feld angeben ['gDaten', 'Links(gDaten, Länge(gDaten) - 1)']
```

Abb. 14

(5) *\_PlatzhalterAustauschen*: (Abb. 15)

Das Teilsript *\_PlatzhalterAustauschen* ersetzt die Platzhalter in *gFunktionenMitPlatzhalter* (Abb. 16) durch die tatsächlichen Werte.

```
Feld angeben ['gFunktionen', 'Austauschen(gFunktionenMitPlatzhalter; "<gDaten>"; gDaten)']
Feld angeben ['gFunktionen', 'Austauschen(gFunktionen; "<gXSkalierung>"; gXSkalierung)']
Feld angeben ['gFunktionen', 'Austauschen(gFunktionen; "<gXAchsenbeschriftungen>"; gXAch...
Feld angeben ['gFunktionen', 'Austauschen(gFunktionen; "<gYAchsenbeschriftungen>"; gYAch...
```

Abb. 15

```

OpenDrawing(400;300)
  ChartData(<gDaten>)
  GanttChart(shadow)
  FillStyle(all;darkYellow)
  ShadowStyle(all;1)
  Scaling(x;linear;<gXSkalierung>)
  ScalingOptions(y;on) // y-Skala von oben nach unten
  // Achsen
  AxisMajorTickLabelTexts(x;<gXAchsenbeschriftungen>)
  AxisMajorTickLabelTexts(y;<gYAchsenbeschriftungen>)
  AxisLine(all;0)
  AxisMajorTicks(y;0)
  AxisMajorTicks(all;0)
  AxisMinorTicks(all;0)
  // Raster
  MajorGridLineWidths(x;y;0) // kein horiz. Raster
  MinorGridLineColors(all;all;lightGray)
  GridFrame(xy;1;gray)
CloseDrawing()

```

Abb. 16

(6) *\_DiagrammZeichnen*: (Abb. 17)

Im Teilsript *\_DiagrammZeichnen* werden alle in *gFunktionen* zusammengefaßten Befehle durch die externe Funktion *xmCH-DrawChart()* an *xmCHART* übergeben. Erfolgt die Erstellung des Diagramms fehlerfrei, so wird abschließend das Diagramm aus der Zwischenablage in das Medienfeld *gDiagramm* kopiert.

```

Feld angeben ['gFehler', 'Extern("xmCH-DrawChart", gFunktionen)']
Wenn ['IstLeer(gFehler)']
  Einsetzen ['gDiagramm']
Ende-Wenn

```

Abb. 17

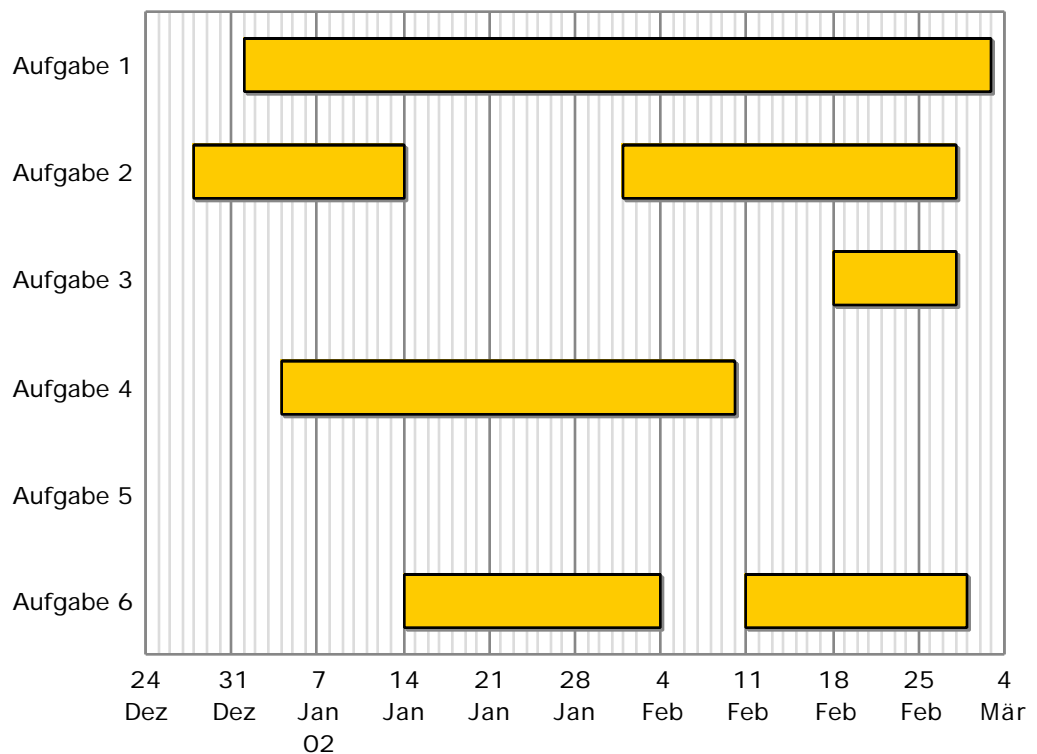
**Ergebnis:**

Abb. 18

**Varianten:**

Die vier in diesem Beispiel verwendeten Skalen können noch weiter verfeinert werden; das gleiche gilt für die Skalenbeschriftungen. Zusätzlich können noch die Balken beschriftet oder um 90 Grad gedreht werden, so dass die Zeitachse nicht von links nach rechts, sondern von oben nach unten verläuft. Beispiele dazu finden sich in *xmTutorial*, Abschnitt *Diagramme*.

# Beispiel 3

## Aufgabenstellung:

Im dritten Beispiel sollen zwei Diagramme, konkret ein Balkendiagramm und ein Liniendiagramm, übereinander platziert werden (siehe Abb. 27). Die dazugehörige FileMaker Pro Datei ist xmBEISPIEL3.FP3 bzw. xmBEISPIEL3.FP5. Bei der Überlagerung von Diagrammen sind folgende Punkte zu beachten:

- Es muß sichergestellt sein, dass die Diagramme, genauer gesagt, die vom Raster umschlossenen Flächen, sich exakt decken. Dies erfolgt durch die beiden Funktionen `OpenChart()` und `CloseChart()`. Dabei ist zu beachten, dass *istDiagrammfläche=on* gesetzt wird. Der allgemeine Aufbau ist:

```
OpenChart(links;oben;breite;höhe;istDiagrammfläche)
// Diagramm 1 definieren
// (Diagramm 1 liegt hinter Diagramm 2)
CloseChart()

OpenChart(links;oben;breite;höhe;istDiagrammfläche)
// Diagramm 2 definieren
// (Diagramm 2 liegt vor Diagramm 1)
CloseChart()
```

- Es muß garantiert sein, dass die Skalierungen der beiden Diagramme ident sind — ausgenommen sind Diagramme mit getrennt angeordneten y-Achsen. Da die einzelnen überlagerten Diagramme völlig unabhängig sind und "voneinander nichts wissen", müssen die Skalierungen für die beiden Diagramme entweder manuell durch die Funktion `Scaling()` oder mit einem kleinen Kniff gleich gesetzt werden: In diesem konkreten Beispiel werden im Zuge der Erstellung der Diagrammdaten, auch der minimale und maximale Wert bestimmt. Diese werden dann als zweite Datenserie bei beiden Diagrammen hinzugefügt. Dadurch ist sichergestellt, dass beide Diagramme die gleiche y-Skalierung besitzen. Diese zweite Datenserie wird unter Verwendung entsprechender Stilfunktionen unsichtbar gesetzt. Bei nicht-gestapelten Balkendiagrammen kann die



unsichtbare zweite Serie zusätzlich noch durch *serienabstand=-100* hinter die erste Serie verschoben werden, so dass die Balken und die Beschriftungen unterhalb der x-Achse exakt zusammenpassen.

- Weiters ist es zweckmäßig, nur die Achsen und Raster des ersten, hinteren Diagramms darzustellen und beim zweiten, vorderen Diagramm auszublenden.

### Lösung:

#### FileMaker Pro Felder: (Abb. 19)

Neben den in der Einleitung beschriebenen Variablenfeldern *gFunktionen*, *gDiagramm*, *gFehler* und *gFunktionenMitPlatzhalter* werden noch die folgenden Felder benötigt:

- Das Textfeld *Name*, welches die Beschriftungen der x-Achse enthält.
- Die Zahlenfelder *Balkenwert* und *Linienwert*, welche die jeweiligen Diagrammdaten enthalten.
- Die Variablenfelder *gBalkendaten*, *gLiniendaten*, *gXAchsenbeschriftungen* und *gMinWert*. Diese korrespondieren mit den entsprechenden Platzhaltern *<gBalkendaten>*, *<gLiniendaten>*, *<gXAchsenbeschriftungen>* und *<gMinWert>* und enthalten die tatsächlichen Werte. Zusätzlich noch das Variablenfeld *gMaxWert*.
- Optional die Textvariablen *gIstAktiv* und *gVersion*. Falls xmCHART aktiv ist, enthält *gIstAktiv* den Wert "1", ansonst ist *gIstAktiv* leer. *gVersion* enthält die genaue Bezeichnung der momentan aktiven xmCHART Version, zum Beispiel "xmCHART 2.2".

<u>Feldname</u>	<u>Feldtyp</u>	<u>Optionen</u>
[1]	Formel	Indiziert, = 1
Name	Text	
Balkenwert	Zahl	
Linienwert	Zahl	
gFunktionenMitPlatzhal...	Variable	Text
gFunktionen	Variable	Text
gDiagramm	Variable	Medien
gFehler	Variable	Text
gVersion	Variable	Text
gIstAktiv	Variable	Text
gBalkendaten	Variable	Text
gLiniendaten	Variable	Text
gXAchsenbeschriftungen	Variable	Text
gMinWert	Variable	Zahl
gMaxWert	Variable	Zahl

Abb. 19

**FileMaker Pro Scripts:**

- FileMaker Pro Script *DiagrammErstellen*: (Abb. 20)

Das Script *DiagrammErstellen* erzeugt das gewünschte Diagramm und ist aus mehreren Teilscrippts aufgebaut. Zuerst kann optional abgefragt werden, ob die passende FileMaker Pro Version vorhanden ist (Script: *\_IstFMVersion4.0+*) und ob xmCHART aktiv ist (Script: *\_IstPluginAktiv*).

```
Fenster fixieren
Script ausführen [Teilscrippts, '_IstFMVersion4.0+']
Script ausführen [Teilscrippts, '_IstPluginAktiv']
Script ausführen [Teilscrippts, '_gMinMaxBestimmen']
Script ausführen [Teilscrippts, '_gDatenErstellen']
Script ausführen [Teilscrippts, '_gXAchsenbeschriftungenErstellen']
Script ausführen [Teilscrippts, '_PlatzhalterAustauschen']
Script ausführen [Teilscrippts, '_DiagrammZeichnen']
```

Abb. 20

*(1) \_gMinMaxBestimmen*: (Abb. 21)

Das Teilsript *\_gMinMaxBestimmen* sucht in *Balkenwert* und *Linienwert* den kleinsten und größten Wert und speichert diese in *gMinWert* bzw. *gMaxWert*.

```
Feld angeben ['gMinWert', '0']
Feld angeben ['gMaxWert', '0']
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Wenn [' NICHT IstLeer(gBalkendaten)']
    Feld angeben ['gMinWert', 'Wenn(Balkenwert < gMinWert; Balkenwert; gMinWert)']
    Feld angeben ['gMaxWert', 'Wenn(Balkenwert > gMaxWert; Balkenwert; gMaxWert)']
  Ende-Wenn
  Wenn [' NICHT IstLeer(Linienwert)']
    Feld angeben ['gMinWert', 'Wenn(Linienwert < gMinWert; Linienwert; gMinWert)']
    Feld angeben ['gMaxWert', 'Wenn(Linienwert > gMaxWert; Linienwert; gMaxWert)']
  Ende-Wenn
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
```

Abb. 21

*(2) \_gDatenErstellen*: (Abb. 22)

Das Teilsript *\_gDatenErstellen* entnimmt aus der Datenbank die zur Darstellung des Balkendiagramms bzw. Liniendiagramms benötigten Werte. Die Werte werden in Form einer Zeichenkette gespeichert — die einzelnen Zahlenwerte durch Leerzeichen getrennt. Dies geschieht mittels einer Schleife über alle Datensätze. Dabei werden eventuell fehlende Werte beim Balkendiagramm durch Null ersetzt, beim Liniendiagramm durch einen Wert kleiner *gMinWert*, zum Beispiel *gMinWert-1*. Wird nun mittels *ChartDataLowerLimits(gMinWert)* ein unterer Grenzwert vorgegeben — das heißt, alle Werte kleiner als *gMinWert* werden igno-

riert — so wird der Linienzug an den fehlenden Punkten unterbrochen dargestellt. Weitere Informationen und Beispiele dazu finden sich in *xmTutorial*, Abschnitt *Daten*.

Abschließend werden im Script *\_gDatenErstellen* noch die Extremwerte *gMinWert* und *gMaxWert* als zweite Datenserie hinzugefügt. Dadurch wird garantiert, dass beide Diagramme die gleiche y-Skalierung verwenden.

```
Feld angeben ['gBalkendaten', '']
Feld angeben ['gLiniendaten', '']
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Wenn ['IstLeer(Balkenwert)']
    Feld angeben ['gBalkendaten', 'gBalkendaten & "0 "' ]
  Sonst
    Feld angeben ['gBalkendaten', 'gBalkendaten & Balkenwert & " "' ]
  Ende-Wenn
  Wenn ['IstLeer(Linienwert)']
    Feld angeben ['gLiniendaten', 'gLiniendaten & ZahlZuText(gMinWert-1) & " "' ]
  Sonst
    Feld angeben ['gLiniendaten', 'gLiniendaten & Linienwert & " "' ]
  Ende-Wenn
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
Feld angeben ['gBalkendaten', 'gBalkendaten & ";" & gMinWert & " " & gMaxWert']
Feld angeben ['gLiniendaten', 'gLiniendaten & ";" & gMinWert & " " & gMaxWert']
```

Abb. 22

### (3) *\_gXAchsenbeschriftungenErstellen*: (Abb. 23)

Das Teilsript *\_gXAchsenbeschriftungenErstellen* entnimmt aus der Datenbank die Beschriftungstexte für die x-Achse und speichert diese in Form einer Zeichenkette in *gXAchsenbeschriftungen*. Die einzelnen Namen sind unter Hochkomma zu setzen und durch Strichpunkte zu trennen. Dabei ist zu beachten, dass die bei der Eingabe von Texten erforderlichen Hochkommas im Script Editor immer doppelt eingegeben werden müssen.

```
Feld angeben ['gXAchsenbeschriftungen', '']
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Feld angeben ['gXAchsenbeschriftungen', 'gXAchsenbeschriftungen & "" & Name & "" ;"]
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
Feld angeben ['gXAchsenbeschriftungen', 'Links(gXAchsenbeschriftungen; Länge(gXAchsenbe...
```

Abb. 23

### (4) *\_PlatzhalterAustauschen*: (Abb. 24)

Das Teilsript *\_PlatzhalterAustauschen* ersetzt die Platzhalter in *gFunktionenMitPlatzhalter* (Abb. 25) durch die tatsächlichen Werte.

```

Feld angeben ['gFunktionen', 'Austauschen(gFunktionenMitPlatzhalter; "<gMinWert>"; gMinWert)']
Feld angeben ['gFunktionen', 'Austauschen(gFunktionen; "<gBalkendaten>"; gBalkendaten)']
Feld angeben ['gFunktionen', 'Austauschen(gFunktionen; "<gLiniendaten>"; gLiniendaten)']
Feld angeben ['gFunktionen', 'Austauschen(gFunktionen; "<gXAchsenbeschriftungen>"; gXAchsenbe...

```

Abb. 24

```

OpenDrawing(400;300)
// BALKENDIAGRAMM
OpenChart(40;20;340;250;on)
    ChartData(<gBalkendaten>)
    BarChart(label;;-100) // -100 verschiebt 2. Serie
    AxisMajorTicks(all;0) // keine Skalenmarkierungen
    AxisMajorTickLabelTexts(x;<gXAchsenbeschriftungen>)
    // Raster
    MajorGridLineWidths(y;x;0) // kein vertikales Raster
    MajorGridLineColors(all;all;lightGray)
    MajorGridStripeColors(x;y;240 240 240;250 250 250)
    GridFrame()
    // 2.Serie unsichtbar setzen
    BorderStyle(2;none)
    FillStyle(2;;transparent)
    LabelTexts(2;" ")
CloseChart()

// LINIENDIAGRAMM
OpenChart(40;20;340;250;on)
    ChartData(<gLiniendaten>)
    ChartDataLowerLimits(<gMinWert>)
    LineChart(label;on)
    LineStyle(1;poly;2;darkYellow)
    LabelBackground(1;lightYellow)
    LabelOptions(1;centerCenter)
    AxisOptions(all;none) // Achsen ausblenden
    GridLocation(;none) // Raster ausblenden
    // 2.Serie unsichtbar setzen
    LineStyle(2;none)
    LabelTexts(2;" ")
CloseChart()
SetDecimalPoint(",")
CloseDrawing()

```

Abb. 25

(5) *\_DiagrammZeichnen*: (Abb. 26)

Im Teilsript *\_DiagrammZeichnen* werden alle in *gFunktionen* zusammengefaßten Befehle durch die externe Funktion `xmCH-DrawChart()` an `xmCHART` übergeben. Erfolgt die Erstellung des Diagramms fehlerfrei, so wird abschließend das Diagramm aus der Zwischenablage in das Medienfeld *gDiagramm* kopiert.

```
Feld angeben ['gFehler', 'Extern("xmCH-DrawChart", gFunktionen)']
Wenn ['IstLeer(gFehler)']
  Einsetzen ['gDiagramm']
Ende-Wenn
```

Abb. 26

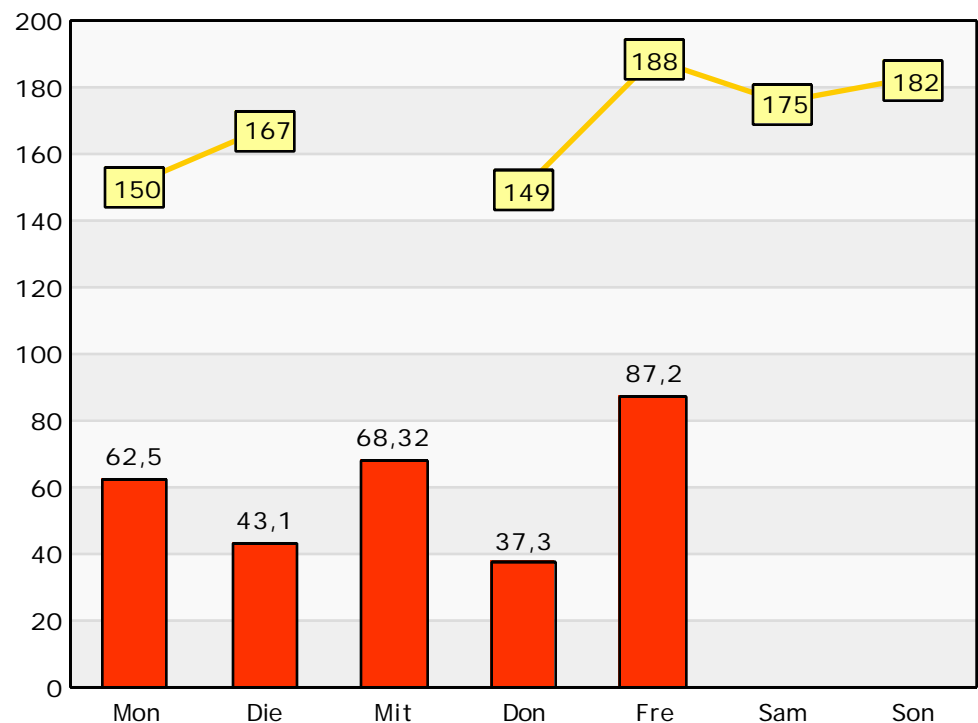
**Ergebnis:**

Abb. 27

**Varianten:**

Optional kann sowohl für das Balken- als auch für das Liniendiagramm eine eigene Skalierung verwendet werden. Dabei wird zum Beispiel die y-Achse des Balkendiagramms links dargestellt (default) und die y-Achse des Liniendiagramms auf der rechten Seite — die y-Achse wird durch die Funktion `AxisOptions(y;ion)` nach rechts verschoben. Ein Beispiel dazu findet sich in `xmGALERIE.FP3` bzw. `xmGALERIE.FP5`.

## Beispiel 4

Im vierten Beispiel sollen Feldinhalte und deren Summen grafisch ausgewertet werden. Die dazugehörige FileMaker Pro Datei ist xmBEISPIEL4.FP3 bzw. xmBEISPIEL4.FP5.

### Aufgabenstellung:

Angenommen Sie wollen die Umsätze Ihrer Firma grafisch aufbereiten. Dabei sollen die Umsätze einerseits quartalsmäßig ausgewertet werden, andererseits auch nach den Mitarbeitern aufgeschlüsselt werden und in Form eines gestapelten Balkendiagramms dargestellt werden.

Zu diesem Zweck haben Sie eine Mitarbeiterdatenbank angelegt, in der die Umsätze der einzelnen Mitarbeiter pro Quartal eingetragen werden. Dabei wird für jeden Mitarbeiter ein Datensatz angelegt mit den Feldern *Mitarbeiter\_Name*, *Umsatz\_Q1*, *Umsatz\_Q2*, *Umsatz\_Q3*, *Umsatz\_Q4*, sowie den Auswertungsfeldern *Summe\_Umsatz\_Q1*, *Summe\_Umsatz\_Q2*, *Summe\_Umsatz\_Q3*, *Summe\_Umsatz\_Q4*, weiters das Formelfeld *Mitarbeiter\_Umsatz*, welches den Gesamtumsatz eines Mitarbeiters innerhalb eines Jahres berechnet und abschließend das Auswertungsfeld *Gesamtumsatz*, welches die Summe aller Umsätze innerhalb eines Jahres enthält.

### Lösung:

#### FileMaker Pro Felder: (Abb. 28)

Neben den zuvor beschriebenen Datenfeldern werden noch die folgenden Felder benötigt:

- Die drei "xmCHART-Felder" *gFunktionen*, *gDiagramm* und *gFehler*.
- Optional die Textvariablen *gIstAktiv* und *gVersion*. Falls xmCHART aktiv ist, enthält *gIstAktiv* den Wert "1", ansonst ist *gIstAktiv* leer. *gVersion* enthält die genaue Bezeichnung der momentan aktiven xmCHART Version, zum Beispiel "xmCHART 2.2".

<u>Feldname</u>	<u>Feldtyp</u>	<u>Optionen</u>
Mitarbeiter_Name	Text	
Umsatz_Q1	Zahl	
Umsatz_Q2	Zahl	
Umsatz_Q3	Zahl	
Umsatz_Q4	Zahl	
Summe_Umsatz_Q1	Auswertung	= Summe von Umsatz_Q1
Summe_Umsatz_Q2	Auswertung	= Summe von Umsatz_Q2
Summe_Umsatz_Q3	Auswertung	= Summe von Umsatz_Q3
Summe_Umsatz_Q4	Auswertung	= Summe von Umsatz_Q4
Mitarbeiter_Umsatz	Formel	= Umsatz_Q1 + Umsatz_Q2 + ...
Gesamtumsatz	Auswertung	= Summe von Mitarbeiter_Umsatz
gFunktionen	Variable	Text
gDiagramm	Variable	Medien
gFehler	Variable	Text
gVersion	Variable	Text
glstAktiv	Variable	Text

Abb. 28

**Umsatz pro Quartal:**

- FileMaker Pro Script *FunktionenErstellen\_UmsatzProQuartal*:  
Das Script *FunktionenErstellen\_UmsatzProQuartal* ist aus drei Teilscripts aufgebaut: (Abb. 29)

```

Fenster fixieren
Feld angeben ['gFunktionen', '"OpenDrawing(420;300)¶"']
Script ausführen [Teilscripts, '_DiagrammDaten_UmsatzProQuartal']
Script ausführen [Teilscripts, '_Funktionen_UmsatzProQuartal']
Script ausführen [Teilscripts, '_Legendentexte_UmsatzProQuartal']

```

Abb. 29

**(1) \_DiagrammDaten\_UmsatzProQuartal:**

Dieses Teilscripts baut die Funktion `ChartData()` auf. Um die Umsätze pro Quartal zu bestimmen, werden mittels einer Schleife über alle Datensätze die Mitarbeiterumsätze zu `ChartData()` hinzugefügt. (siehe Abb. 30)

```

Feld angeben ['gFunktionen', 'gFunktionen & "ChartData("']
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
    Feld angeben ['gFunktionen', 'gFunktionen & Umsatz_Q1 & " " & Umsatz_Q2 & ...']
    Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
Feld angeben ['gFunktionen', 'Links(gFunktionen;Länge(gFunktionen) - 1) & ")"']

```

Abb. 30

**(2) *\_Funktionen\_UmsatzProQuartal:***

Die zusätzlich zu `ChartData()` notwendigen Funktionsaufrufe werden wie in Abb. 31 dargestellt zu *gFunktionen* hinzugefügt.

```
gFunktionen &
"BarChart(stacked)" &
"AxisMajorTickLabelTexts(x;"Q1"";"Q2"";"Q3"";"Q4"")" &
"TitleText("Umsatz pro Quartal"")" &
"AddText(320;250;"Gesamtumsatz:\n" & Gesamtumsatz & """";;bold)" &
"SetThousandsSep("".")"
```

Abb. 31

**(3) *\_Legendentexte\_UmsatzProQuartal:***

Da die Legende sowohl die Mitarbeiternamen als auch die dazugehörenden Umsätze enthalten soll, wird die Funktion `LegendTexts()` dynamisch aufgebaut. Das heißt, mittels einer Schleife über alle Datensätze werden die Mitarbeiternamen samt den dazugehörenden Umsätzen zur Funktion `LegendTexts()` hinzugefügt. (Abb. 32)

```
Feld angeben ['gFunktionen', 'gFunktionen & "LegendTexts("]
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Feld angeben ['gFunktionen', 'gFunktionen & """" & Mitarbeiter_Name & ...]
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
Feld angeben ['gFunktionen', 'Links(gFunktionen;Länge(gFunktionen) - 1) & ")"]
```

Abb. 32

- **FileMaker Pro Feld *gFunktionen*:**

Nach Ausführen des Scripts *FunktionenErstellen\_UmsatzProQuartal* enthält das Variablenfeld *gFunktionen* die folgenden Funktionsaufrufe für xmCHART. (Abb. 33)

```
OpenDrawing(420;300)
ChartData(32500 62400 49700 42100; 12300 32500 53200;
23900 52300 53400 45200;30100 58200 56100 0;
0 0 22000 0)
BarChart(stacked)
AxisMajorTickLabelTexts(x;"Q1";"Q2";"Q3";"Q4")
TitleText("Umsatz pro Quartal")
AddText(320;250;"Gesamtumsatz:\n625900";;bold)
SetThousandsSep("".")
LegendTexts("MA-1 (186700)";"MA-2 (98000)";
"MA-3 (174800)";"MA-4 (144400)";"MA-5 (22000)")
```

Abb. 33



- FileMaker Pro Feld *gDiagramm*:  
Das von xmCHART erstellte und anschließend in *gDiagramm* gespeicherte Diagramm ist in Abb. 34 dargestellt.

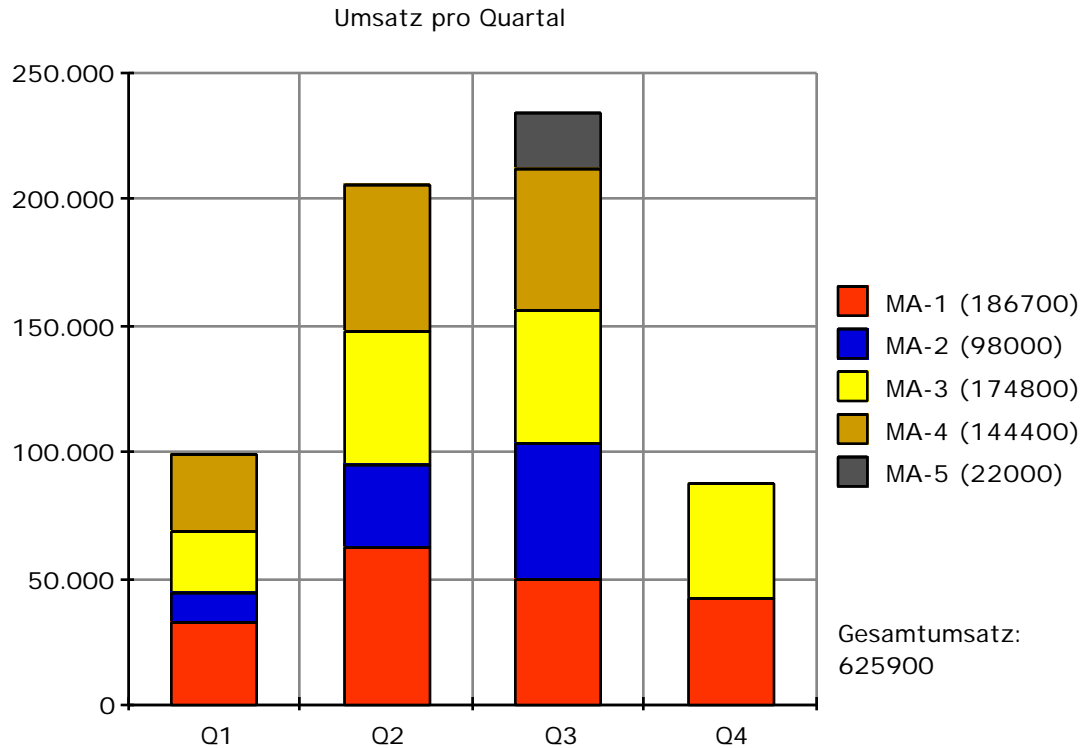


Abb. 34

#### Umsatz pro Mitarbeiter:

- FileMaker Pro Script *FunktionenErstellen\_UmsatzProMitarbeiter*:  
Das Script *FunktionenErstellen\_UmsatzProMitarbeiter* setzt sich aus drei Teilscripts zusammen. (Abb. 35)

```

Fenster fixieren
Feld angeben ['gFunktionen', '"OpenDrawing(420;300)¶"']
Script ausführen [Teilscripts, '_DiagrammDaten_UmsatzProMitarbe...']
Script ausführen [Teilscripts, '_Funktionen_UmsatzProMitarbeiter']
Script ausführen [Teilscripts, '_Legendentexte_UmsatzProMitarbe...']
Script ausführen [Teilscripts, '_Achsenbeschriftung_UmsatzProMi...']

```

Abb. 35

*(1) \_DiagrammDaten\_UmsatzProMitarbeiter:*

Durch dieses Teilsript wird die Funktion `ChartData()` aufgebaut. Um die Umsätze pro Mitarbeiter zu bestimmen, werden quartalsweise mittels einer Schleife über alle Mitarbeiter (= Datensätze) die Umsätze zu `ChartData()` hinzugefügt. Das heißt, es ergeben sich somit insgesamt vier Datenserien (pro Quartal eine Serie) (Abb. 36)

```
Feld angeben ['gFunktionen', 'gFunktionen & "ChartData("']
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Feld angeben ['gFunktionen', 'gFunktionen & Umsatz_Q1 & " "']
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
Feld angeben ['gFunktionen', 'gFunktionen & "; "']
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Feld angeben ['gFunktionen', 'gFunktionen & Umsatz_Q2 & " "']
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
Feld angeben ['gFunktionen', 'gFunktionen & "; "']
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Feld angeben ['gFunktionen', 'gFunktionen & Umsatz_Q3 & " "']
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
Feld angeben ['gFunktionen', 'gFunktionen & "; "']
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Feld angeben ['gFunktionen', 'gFunktionen & Umsatz_Q4 & " "']
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
Feld angeben ['gFunktionen', 'Links(gFunktionen,Länge(gFunktionen) - 1) & ")¶¶"]
```

Abb. 36

*(2) \_Funktionen\_UmsatzProMitarbeiter:*

Die zusätzlich zu `ChartData()` notwendigen Funktionsaufrufe werden wie in Abb. 37 dargestellt zu *gFunktionen* hinzugefügt

```
gFunktionen &
"BarChart(stacked)¶¶" &
"TitleText("Umsatz pro Mitarbeiter")¶¶" &
"AddText(320;250;"Gesamtumsatz:\n" & Gesamtumsatz & """";;;bold)¶¶" &
"SetThousandsSep("".")¶¶"
```

Abb. 37

**(3) *\_Legendentexte\_UmsatzProMitarbeiter:***

Die Funktion `LegendTexts()` wird wie in Abb. 38 dargestellt, durch Verknüpfen von Textkonstanten und Feldinhalten zusammengesetzt.

```
gFunktionen & "LegendTexts(
""1.Quartal (" & Summe_Umsatz_Q1 & ")"";
""2.Quartal (" & Summe_Umsatz_Q2 & ")"";
""3.Quartal (" & Summe_Umsatz_Q3 & ")"";
""4.Quartal (" & Summe_Umsatz_Q4 & ")""
)"
```

Abb. 38

**(4) *\_Achsenbeschriftung\_UmsatzProMitarbeiter:***

Da die X-Achsenbeschriftung die Mitarbeiternamen enthalten soll, wird die Funktion `AxisMajorTickLabelTexts()` dynamisch aufgebaut. Das heißt, mittels einer Schleife über alle Datensätze werden die Mitarbeiternamen zur Funktion `AxisMajorTickLabelTexts()` hinzugefügt. (siehe Abb. 39)

```
Feld angeben ['gFunktionen', 'gFunktionen & "AxisMajorTickLabelTexts(x;")']
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Feld angeben ['gFunktionen', 'gFunktionen & "" & Mitarbeiter_Name & "";"']
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
Feld angeben ['gFunktionen', 'Links(gFunktionen,Länge(gFunktionen) - 1) & ")"]
```

Abb. 39

- **FileMaker Pro Feld *gFunktionen*:**

Nach Ausführen des Scripts *FunktionenErstellen\_UmsatzProMitarbeiter* enthält das Variablenfeld *gFunktionen* die folgenden Funktionsaufrufe für xmCHART. (Abb. 40)

```
OpenDrawing(420;300)
ChartData(32500 23900 30100 0 ;
62400 12300 52300 58200 0 ;
49700 32500 53400 56100 22000 ;
42100 53200 45200 0 0)
BarChart(stacked)
TitleText("Umsatz pro Mitarbeiter")
AddText(320;250;"Gesamtumsatz:\n625900";;bold)
SetThousandsSep(".")
LegendTexts( "1.Quartal (86500)"; "2.Quartal (185200)";
"3.Quartal (213700)"; "4.Quartal (140500)" )
AxisMajorTickLabelTexts(x;"MA-1";"MA-2";"MA-3";
"MA-4";"MA-5")
```

Abb. 40

- FileMaker Pro Feld *gDiagramm*:  
Die von xmCHART erstellte und anschließend in *gDiagramm* gespeicherte Grafik ist in Abb. 41 dargestellt.

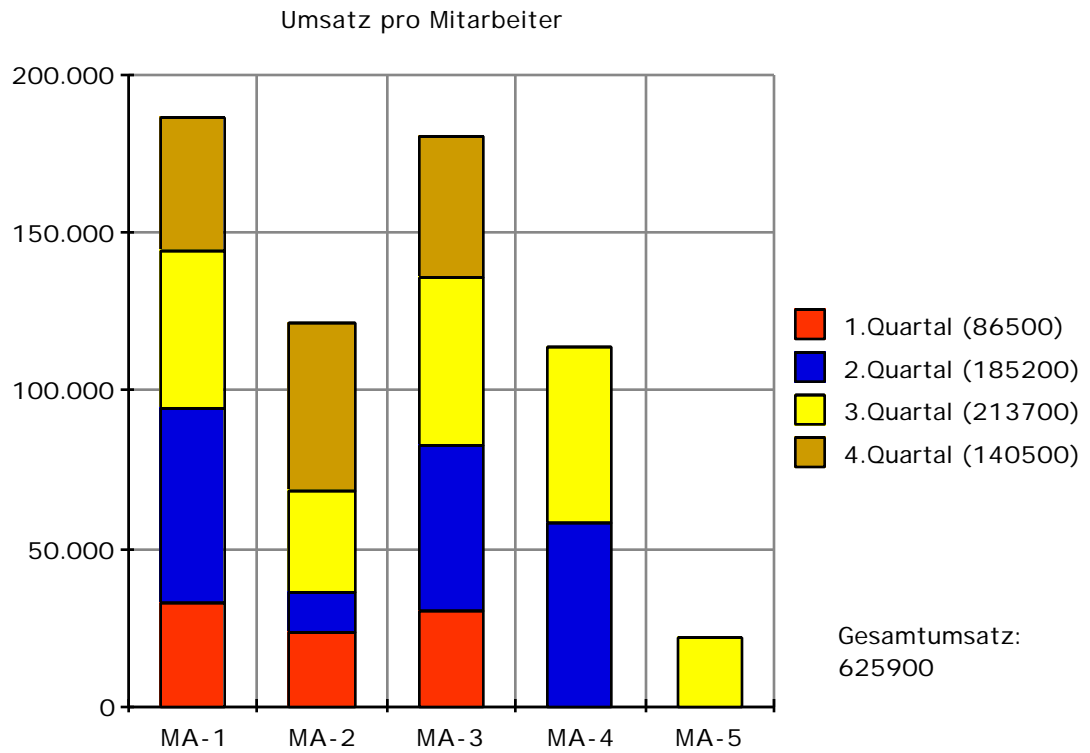


Abb. 41

# Beispiel 5

Im fünften Beispiel sollen Feldinhalte nach der Häufigkeit ihres Auftretens grafisch ausgewertet werden. Die dazu korrespondierende FileMaker Pro Datei ist xmBEISPIEL5.FP3 bzw. xmBEISPIEL5.FP5.

## Aufgabenstellung:

Angenommen Sie betreiben eine Web-Site und wollen sich über die Verteilung der Besucherfrequenz innerhalb eines Tages bzw. innerhalb einer Woche einen Überblick verschaffen und wollen weiters noch die Besucher Ihrer Web-Site nach deren Herkunftsländern aufschlüsseln.

Zu diesem Zweck haben Sie eine FileMaker Pro Datei mit drei Feldern, *Stunde*, *Wochentag* und *Land* angelegt. Die entsprechenden Daten haben Sie aus der Zugriffsprotokoll-Datei Ihrer Web-Site übernommen. Das Zahlenfeld *Stunde* enthält einen Wert zwischen 0 und 23 — detailliertere Informationen über Minuten und Sekunden scheinen nicht auf. Das Zahlenfeld *Wochentag* enthält eine Zahl zwischen 1 (Sonntag) und 7 (Samstag) und das Textfeld *Land* enthält ein zwei Buchstaben langes Länderkürzel.

## Lösung:

### FileMaker Pro Felder: (Abb. 42)

Neben den drei zuvor beschriebenen Datenfeldern *Stunde*, *Wochentag* und *Land* werden noch die folgenden Felder benötigt:

- Die Variablenfelder *gLand*, *gBeschriftung* und *gZaehler*.  
Diese Hilfsvariablen werden zum Aufbau der Funktionen `ChartData()` und `AxisMajorTickLabelTexts()` benötigt.
- Die drei "xmCHART-Felder" *gFunktionen*, *gDiagramm* und *gFehler*.
- Optional die Textvariablen *gIstAktiv* und *gVersion*. Falls xmCHART aktiv ist, enthält *gIstAktiv* den Wert "1", ansonst ist *gIstAktiv* leer. *gVersion* enthält die genaue Bezeichnung der momentan aktiven xmCHART Version, zum Beispiel "xmCHART 2.2".

<u>Feldname</u>	<u>Feldtyp</u>	<u>Optionen</u>
Stunde	Zahl	Indiziert
Wochentag	Zahl	Indiziert
Land	Text	
gLand	Variable	Text
gBeschriftung	Variable	Text
gZaehler	Variable	Zahl
gFunktionen	Variable	Text
gDiagramm	Variable	Medien
gFehler	Variable	Text
gIstAktiv	Variable	Text
gVersion	Variable	Text

Abb. 42

### Häufigkeitsverteilung über 24 Stunden:

- FileMaker Pro Script *FunktionenErstellen\_24Stunden*:

Das Script *FunktionenErstellen\_24Stunden* setzt sich aus zwei Teilscripts zusammen: (siehe Abb. 43)

```

Fenster fixieren
Feld angeben ['gFunktionen', '"OpenDrawing(420;300)¶"']
Script ausführen [Teilscripts, '_DiagrammDaten_24Stunden']
Script ausführen [Teilscripts, '_Funktionen_24Stunden']

```

Abb. 43

#### (1) *\_DiagrammDaten\_24Stunden*:

Durch dieses Teilscrip wird die Funktion `ChartData()` aufgebaut. Um die Anzahl der Besucher pro Stunde zu ermitteln, wird mittels einer Schleife nach Datensätzen mit *Stunde=0*, *Stunde=1*, *Stunde=2* bis *Stunde=23* gesucht und die Anzahl der gefundenen Datensätze zu `ChartData()` hinzugefügt. Falls innerhalb eines bestimmten Stundenintervalls keine Besucher gefunden werden, so ist zu beachten, dass der daraus resultierende Fehler (Fehlernummer 401 – keine Datensätze vorhanden) mittels des FileMaker Pro Scriptbefehls *Fehlerrückmeldung* `setzen[Ein]` abgefangen werden muß und anschließend der Wert 0 zu `ChartData()` hinzuzufügen ist. (Abb. 44)

```

Fehlerrückmeldung setzen [Ein]
Feld angeben ['gFunktionen', 'gFunktionen & "ChartData("")']
Feld angeben ['gZaehler', '0']
Schleife
  Suchen aktivieren []
  Feld angeben ['Stunde', 'gZaehler']
  Suchen []
  Wenn ['Status(AktuelleFehler) = 401']
    Feld angeben ['gFunktionen', 'gFunktionen & "0 "'']
  Sonst
    Feld angeben ['gFunktionen', 'gFunktionen & Status(AktuelleDatensatzanzahl) & " "'']
  Ende-Wenn
  Schleife-Verlassen-Wenn ['gZaehler = 23']
    Feld angeben ['gZaehler', 'gZaehler + 1']
  Ende-Schleife
Feld angeben ['gFunktionen', 'Links(gFunktionen;Länge(gFunktionen) - 1) & "␣"']
Fehlerrückmeldung setzen [Aus]

```

Abb. 44

### (2) *Funktionen\_24Stunden:*

Die zusätzlich zu `ChartData()` notwendigen Funktionsaufrufe werden wie in Abb. 45 dargestellt zu *gFunktionen* hinzugefügt.

```

gFunktionen &
"BarChart()␣" &
"PictureStyle(1;resource;"8")␣" &
"TitleText("Besuchfrequenz über 24 Stunden")␣" &
"TitleBackground()␣" &
"GridLocation(all;none)␣" &
"AxisMajorTicks(y;3;;;out)␣" &
"AxisMajorTicks(x;0)␣"

```

Abb. 45

- FileMaker Pro Feld *gFunktionen*:

Nach Ausführen des Scripts *FunktionenErstellen\_24Stunden* enthält das Variablenfeld *gFunktionen* die folgenden Funktionsaufrufe: (Abb. 46)

```

OpenDrawing(420;300)
ChartData(65 46 37 33 24 33 36 47 59 42 41 47 55 61 62 76
63 68 70 57 62 73 65 66)
BarChart()
PictureStyle(1;resource;"8")
TitleText("Besucherfrequenz über 24 Stunden")
TitleBackground()
GridLocation(all;none)
AxisMajorTicks(y;3;;;out)
AxisMajorTicks(x;0)

```

Abb. 46

- FileMaker Pro Feld *gDiagramm*:  
Das von xmCHART erstellte und anschließend in *gDiagramm* gespeicherte Diagramm ist in Abb. 47 dargestellt.

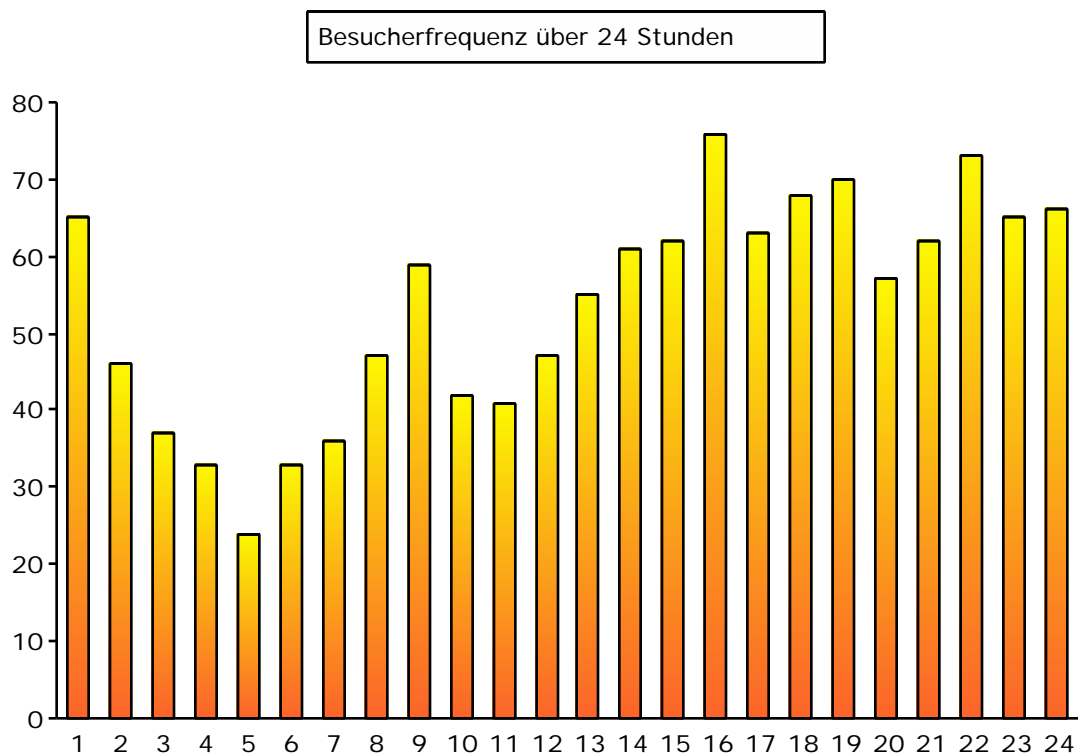


Abb. 47



**Häufigkeitsverteilung über Wochentage:**

- FileMaker Pro Script *FunktionenErstellen\_Wochentage*:

Das Script *FunktionenErstellen\_Wochentage* ist ebenfalls aus zwei Teilscripts aufgebaut: (Abb. 48)

```
Fenster fixieren
Feld angeben ['gFunktionen', 'OpenDrawing(420;300)¶']
Script ausführen [Teilscripts, '_DiagrammDaten_Wochentage']
Script ausführen [Teilscripts, '_Funktionen_Wochentage']
```

Abb. 48

**(1) \_DiagrammDaten\_Wochentage:**

Durch dieses Teilscrip wird die Funktion `ChartData()` aufgebaut. Um die Anzahl der Besucher pro Wochentag zu ermitteln, wird mittels einer Schleife nach Datensätzen mit *Wochentag=1*, *Wochentag=2* bis *Wochentag=7* gesucht und die Anzahl der gefundenen Datensätze zu `ChartData()` hinzugefügt. Falls innerhalb eines Tages keine Besucher gefunden werden, so ist zu beachten, dass der daraus resultierende Fehler (Fehlernummer 401 – keine Datensätze vorhanden) mittels des FileMaker Pro Scriptbefehls *Fehleraufzeichnung setzen[Ein]* abgefangen werden muß und anschließend der Wert 0 zu `ChartData()` hinzuzufügen ist. (Abb. 49)

```
Fehleraufzeichnung setzen [Ein]
Feld angeben ['gFunktionen', 'gFunktionen & "ChartData("']
Feld angeben ['gZaehler', '1']
Schleife
  Suchen aktivieren []
  Feld angeben ['Wochentag', 'gZaehler']
  Suchen []
  Wenn ['Status(AktuellFehler) = 401']
    Feld angeben ['gFunktionen', 'gFunktionen & "0 "']
  Sonst
    Feld angeben ['gFunktionen', 'gFunktionen & Status(AktuellDatensatzanzahl) & " "']
  Ende-Wenn
  Schleife-Verlassen-Wenn ['gZaehler = 7']
  Feld angeben ['gZaehler', 'gZaehler + 1']
Ende-Schleife
Feld angeben ['gFunktionen', 'Links(gFunktionen;Länge(gFunktionen) - 1) & "¶"']
Fehleraufzeichnung setzen [Aus]
```

Abb. 49

**(2) \_Funktionen\_Wochentage:**

Die zusätzlich zu `ChartData()` notwendigen Funktionsaufrufe werden wie in Abb. 50 dargestellt zu *gFunktionen* hinzugefügt.

```

gFunktionen &
"BarChart(label)" &
"TitleText("Besucherfrequenz pro Wochentag")" &
"TitleBackground(;transparent)" &
"Background(;119;0)" &
"AxisMajorTicks(y;3;;;out)" &
"AxisMajorTicks(x;0)" &
"AxisMajorTickLabelTexts(x;"Son";"Mon";"Die";
"Mit";"Don";"Fre";"Sam")"

```

Abb. 50

- FileMaker Pro Feld *gFunktionen*:

Nach Ausführen des Scripts *FunktionenErstellen\_Wochentage* enthält das Variablenfeld *gFunktionen* die folgenden Funktionsaufrufe: (siehe Abb. 51)

```

OpenDrawing(420;300)
ChartData(118 108 222 236 209 214 181)
BarChart(label)
TitleText("Besucherfrequenz pro Wochentag")
TitleBackground(;transparent)
Background(;119;0)
AxisMajorTicks(y;3;;;out)
AxisMajorTicks(x;0)
AxisMajorTickLabelTexts(x;"Son";"Mon";"Die";
"Mit";"Don";"Fre";"Sam")

```

Abb. 51

- FileMaker Pro Feld *gDiagramm*:

Das von xmCHART erstellte und anschließend in *gDiagramm* gespeicherte Diagramm ist in Abb. 52 dargestellt.

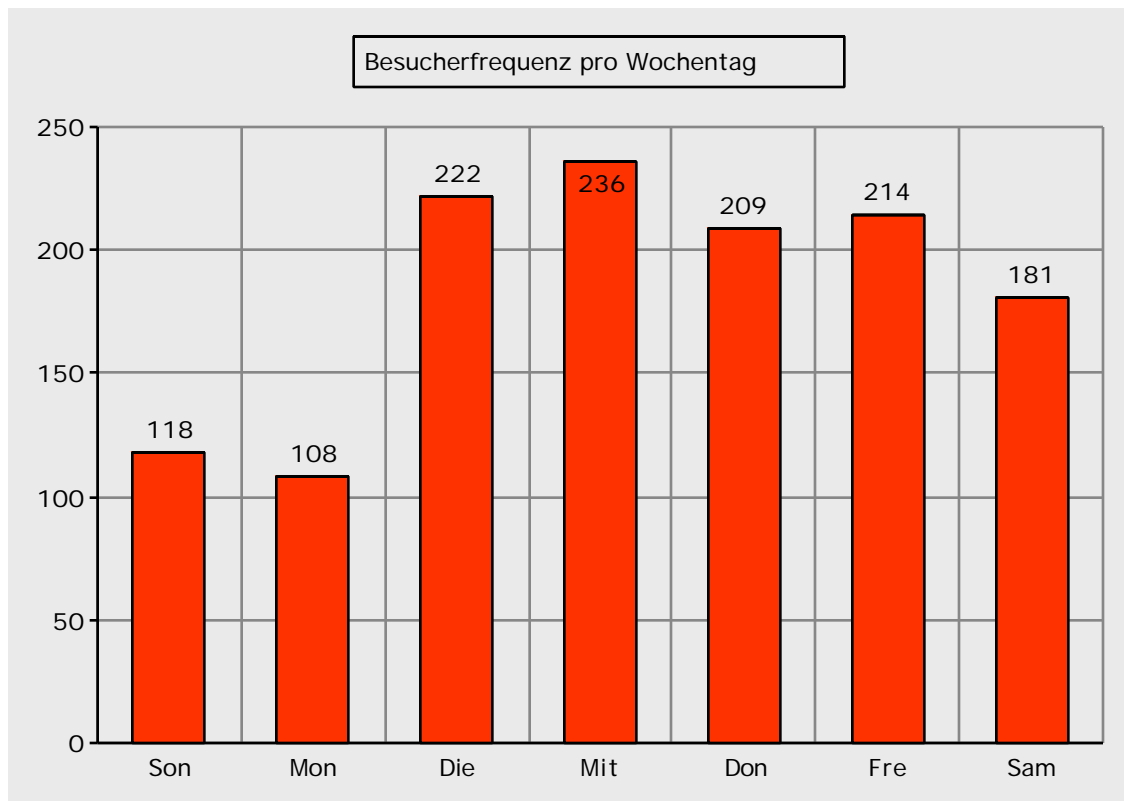


Abb. 52

**Aufteilung nach Ländern:**

- FileMaker Pro Script *FunktionenErstellen\_Länder*:

Das Script *FunktionenErstellen\_Länder* setzt sich ebenfalls aus zwei Teilscripts zusammen: (Abb. 53)

```
Fenster fixieren
Feld angeben ['gFunktionen', 'OpenDrawing(420;300)¶']
Script ausführen [Teilscripts, '_DiagrammDaten_Länder']
Script ausführen [Teilscripts, '_Funktionen_Länder']
```

Abb. 53

**(1) \_DiagrammDaten\_Länder:**

Durch dieses Teilsript werden die Funktionen `ChartData()` und `Axis-MajorTickLabelTexts()` aufgebaut. Um die Besucher nach ihren Herkunftsländern aufzuschlüsseln, werden die Datensätze zuerst nach den Länderkürzeln sortiert, und anschließend wird mittels einer Schleife über alle Datensätze, die Häufigkeit der einzelnen Länderkürzel hochgezählt und zu `ChartData()` hinzugefügt. Zusätzlich werden noch die Länderkürzel und deren Häufigkeiten für die Beschriftung der Y-Achse auf die Hilfsvariable *gBeschriftung* zwischengespeichert. (Abb. 54)

```

Feld angeben ['gFunktionen', 'gFunktionen & "ChartData('']
Alle aufrufen
Sortieren [Übernehmen, Ohne Dialogfenster]
Gehe zu Datens./Abfrage/Seite [Erster]
Feld angeben ['gBeschriftung', ""AxisMajorTickLabelTexts(y;""']
Feld angeben ['gLand', 'Land']
Feld angeben ['gZaehler', '1']
Schleife
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
  Wenn ['Land = gLand']
    Feld angeben ['gZaehler', 'gZaehler + 1']
  Sonst
    Feld angeben ['gFunktionen', 'gFunktionen & gZaehler & " "]
    Feld angeben ['gBeschriftung', 'gBeschriftung & """"(" & gZaehler & ") " & gLand & """";""']
    Feld angeben ['gLand', 'Land']
    Feld angeben ['gZaehler', '1']
  Ende-Wenn
Ende-Schleife
Feld angeben ['gFunktionen', 'gFunktionen & gZaehler & " "]
Feld angeben ['gBeschriftung', 'gBeschriftung & """"(" & gZaehler & ") " & gLand & """"']
Feld angeben ['gLand', 'Land']
Feld angeben ['gZaehler', '1']
Feld angeben ['gFunktionen', 'Links(gFunktionen, Länge(gFunktionen) - 1) & ")¶"]
Feld angeben ['gFunktionen', 'gFunktionen & gBeschriftung & ")¶"]

```

Abb. 54

### (2) Funktionen\_Länder:

Die zusätzlich zu ChartData() und AxisMajorTickLabelTexts() notwendigen Funktionsaufrufe werden wie in Abb. 55 dargestellt zu *gFunktionen* hinzugefügt.

```

gFunktionen &
"BarChart(horizontal)¶" &
"TitleText("Besucher pro Land")¶" &
"TitleBackground(;transparent)¶" &
"BackgroundPict(resource;""29""")¶" &
"AxisMajorTicks(x;3;;;out)¶" &
"AxisMajorTicks(y;0)¶" &
"GridLocation(all;none)¶"

```

Abb. 55

### • FileMaker Pro Feld *gFunktionen*:

Nach Ausführen des Scripts *FunktionenErstellen\_Länder* enthält das Variablenfeld *gFunktionen* die folgenden Funktionsaufrufe: (Abb. 56)

```

OpenDrawing(420;300)
ChartData(6 34 76 23 117 54 148 16 23 6 46 64 75 44
8 48 8 7 47 438)
AxisMajorTickLabelTexts(y;"(6) AR";"(34) AT";"(76) AU";
"(23) BE";"(117) CA";"(54) CH";"(148) DE";"(16) DK";
"(23) ES";"(6) FI";"(46) FR";"(64) GB";"(75) IT";
"(44) JP";"(8) MX";"(48) NL";"(8) NO";"(7) NZ";
"(47) SE";"(438) US")
BarChart(horizontal)
TitleText("Besucher pro Land")
TitleBackground(;transparent)
BackgroundPict(resource;"29")
AxisMajorTicks(x;3;;;out)
AxisMajorTicks(y;0)
GridLocation(all;none)

```

Abb. 56

- **FileMaker Pro Feld *gDiagramm*:**  
Die von xmCHART erstellte und anschließend in *gDiagramm* gespeicherte Grafik ist in Abb. 57 dargestellt.

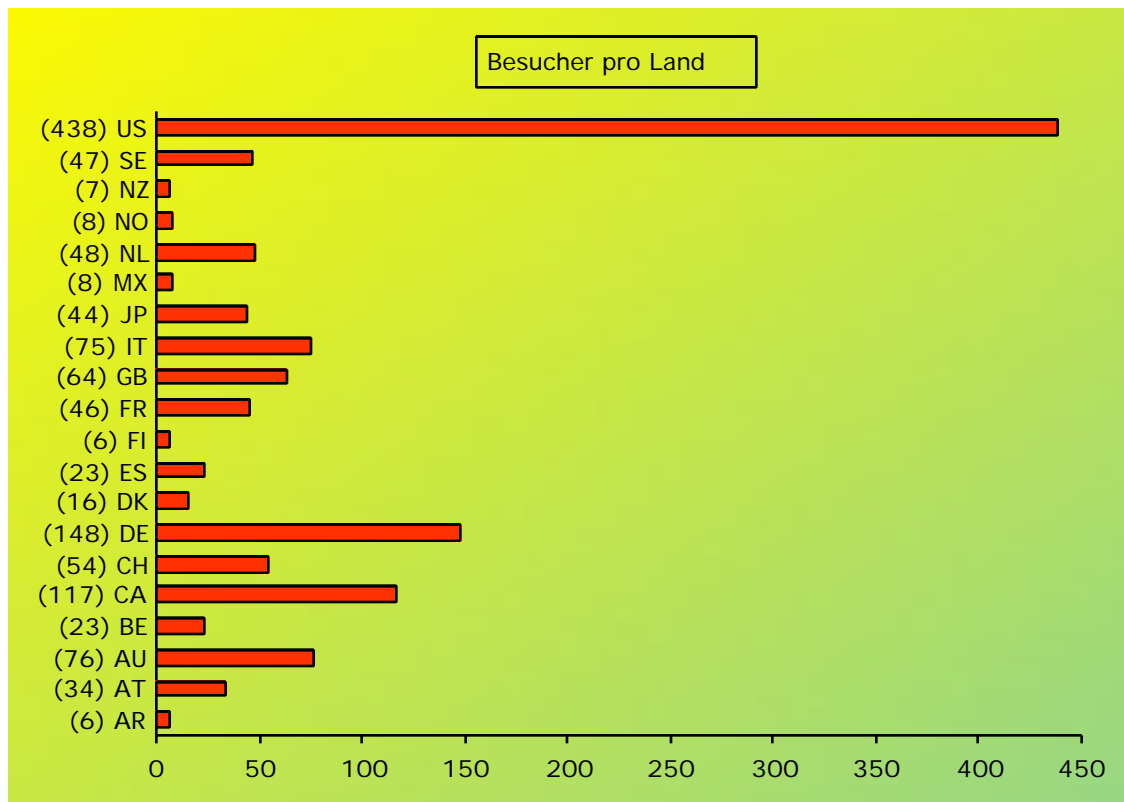


Abb. 57

# Beispiel 6

Im sechsten Beispiel werden zeitabhängige Daten grafisch ausgewertet. Die dafür benötigte FileMaker Pro Datenbank ist xmBEISPIEL6.FP3 bzw. xmBEISPIEL6.FP5.

## Aufgabenstellung:

Angenommen Sie erfassen alle 14 Tage drei Messwerte, welche durch Linienzüge grafisch dargestellt werden sollen. Weiters sollen die Messwertserien zu unterschiedlichen Zeitpunkten beginnen.

Zu diesem Zweck haben Sie eine FileMaker Pro Datei mit vier Feldern, *Datum*, *Serie1*, *Serie2* und *Serie3* angelegt. Für jeden Messtag wird ein neuer Datensatz angelegt. In diesem Beispiel wird ein 2-dimensionales Liniendiagramm verwendet, welches sehr vielseitig einsetzbar ist. So können zum Beispiel Datenserien zu unterschiedlichen Zeitpunkten beginnen, oder weiters wäre es möglich, unterschiedlich lange Messintervalle zu verwenden.

## FileMaker Pro Felder: (Abb. 58)

Neben den zuvor beschriebenen Datenfeldern werden noch die folgenden Felder benötigt:

- Die drei "xmCHART-Felder" *gFunktionen*, *gDiagramm* und *gFehler*.
- Optional die Textvariablen *gIstAktiv* und *gVersion*. Falls xmCHART aktiv ist, enthält *gIstAktiv* den Wert "1", ansonst ist *gIstAktiv* leer. *gVersion* enthält die genaue Bezeichnung der momentan aktiven xmCHART Version, zum Beispiel "xmCHART 2.2".

<u>Feldname</u>	<u>Feldtyp</u>	<u>Optionen</u>
Datum	Datum	
Serie1	Zahl	
Serie2	Zahl	
Serie3	Zahl	
gFunktionen	Variable	Text
gDiagramm	Variable	Medien
gFehler	Variable	Text
gVersion	Variable	Text
gIstAktiv	Variable	Text

Abb. 58

**FileMaker Pro Scripts:**• FileMaker Pro Script *FunktionenErstellen*:

Das Script *FunktionenErstellen* setzt sich aus zwei Teilscripits zusammen:  
(Abb. 59)

```
Fenster fixieren
Feld angeben ['gFunktionen', '"OpenDrawing(420;300)¶"']
Script ausführen [Teilscripits, '_DiagrammDaten']
Script ausführen [Teilscripits, '_Funktionen']
```

Abb. 59

(1) *\_DiagrammDaten*:

Um den Ablauf übersichtlicher zu gestalten, ruft das Script *\_DiagrammDaten* drei weitere Teilscripits auf, welche die Funktion *ChartData()* aufbauen. (Abb. 60)

```
Feld angeben ['gFunktionen', 'gFunktionen & "ChartData("']
Script ausführen [Teilscripits, '__Serie1']
Script ausführen [Teilscripits, '__Serie2']
Script ausführen [Teilscripits, '__Serie3']
Feld angeben ['gFunktionen', 'Links(gFunktionen; Länge(gFunktionen) - 2) & ")¶"']
```

Abb. 60

Die zur Darstellung der ersten Messserie notwendigen Werte werden durch das in Abb. 61 dargestellte Script *\_\_Serie1* aufbereitet. Zuerst werden mittels einer Schleife über alle Datensätze, die Messzeitpunkte bzw. genauer gesagt, die daraus ermittelten laufenden Tage innerhalb eines Jahres an *ChartData()* übergeben. Anschließend werden in einer weiteren Schleife über alle Datensätze, die dazugehörigen Messwerte zu *ChartData()* hinzugefügt.

```
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Wenn ['NICHT IstLeer(Serie1)']
    Feld angeben ['gFunktionen', 'gFunktionen & TagDesJahres(Datum) & " "']
  Ende-Wenn
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
Feld angeben ['gFunktionen', 'gFunktionen & "; "']
Gehe zu Datens./Abfrage/Seite [Erster]
Schleife
  Wenn ['NICHT IstLeer(Serie1)']
    Feld angeben ['gFunktionen', 'gFunktionen & Serie1 & " "']
  Ende-Wenn
  Gehe zu Datens./Abfrage/Seite [Nach letztem beenden, Nächster]
Ende-Schleife
Feld angeben ['gFunktionen', 'gFunktionen & "; "']
```

Abb. 61

**(2) \_Funktionen:**

Die zusätzlich zu `ChartData()` notwendigen Funktionsaufrufe werden wie in Abb. 62 dargestellt zu *gFunktionen* hinzugefügt

```
gFunktionen &
"LineChart2D()" &
"LineStyle(all;poly;2)" &
"LegendTexts("Serie A";"Serie B";"Serie C")" &
"Background (240 240 255;;0)" &
"Scaling(y;linear;400;600;4;5)" &
"AxisMinorTicks(y;3;;;out)" &
"Scaling(x;linear;1;365;12;1)" &
"AxisMajorTickLabelTexts(x;"1.Jan";
"\n1.Feb";"1.Mär";
"\n1.Apr";"1.Mai";
"\n1.Jun";"1.Jul";
"\n1.Aug";"1.Sep";
"\n1.Okt";"1.Nov";
"\n1.Dez";"31.Dez")"
```

Abb. 62

- FileMaker Pro Feld *gFunktionen*:

Nach Ausführen des Scripts *FunktionenErstellen* enthält das Variablenfeld *gFunktionen* die folgenden Funktionsaufrufe: (Abb. 63)

```
OpenDrawing(420;300)
ChartData(1 15 29 43 57 71 85 99 113 127 141 155 169
183 197 211 225 239 253 267 281 295 ; 428 531 493 473
472 473 442 443 467 454 442 475 485 472 489 499 492
512 510 501 519 532 ; 71 85 99 113 127 141 155 169 183
197 211 225 239 253 267 281 295 ; 540 572 567 557 554
563 561 571 572 573 564 542 555 567 572 575 577 ; 113
127 141 155 169 183 197 211 225 239 253 267 281 295 ;
500 505 510 520 525 521 510 512 496 490 489 483 489
479 )
LineChart2D()
LineStyle(all;poly;2)
LegendTexts("Serie A";"Serie B";"Serie C")
Background (240 240 255;;0)
Scaling(y;linear;400;600;4;5)
AxisMinorTicks(y;3;;;out)
Scaling(x;linear;1;365;12;1)
AxisMajorTickLabelTexts(x;"1.Jan";"\n1.Feb"; "1.Mär";
"\n1.Apr";"1.Mai";"\n1.Jun"; "1.Jul";"\n1.Aug";"1.Sep";
"\n1.Okt"; "1.Nov";"\n1.Dez";"31.Dez")
```

Abb. 63



- FileMaker Pro Feld *gDiagramm*:  
Das von xmCHART erstellte und anschließend in *gDiagramm* gespeicherte Diagramm ist in Abb. 64 dargestellt.

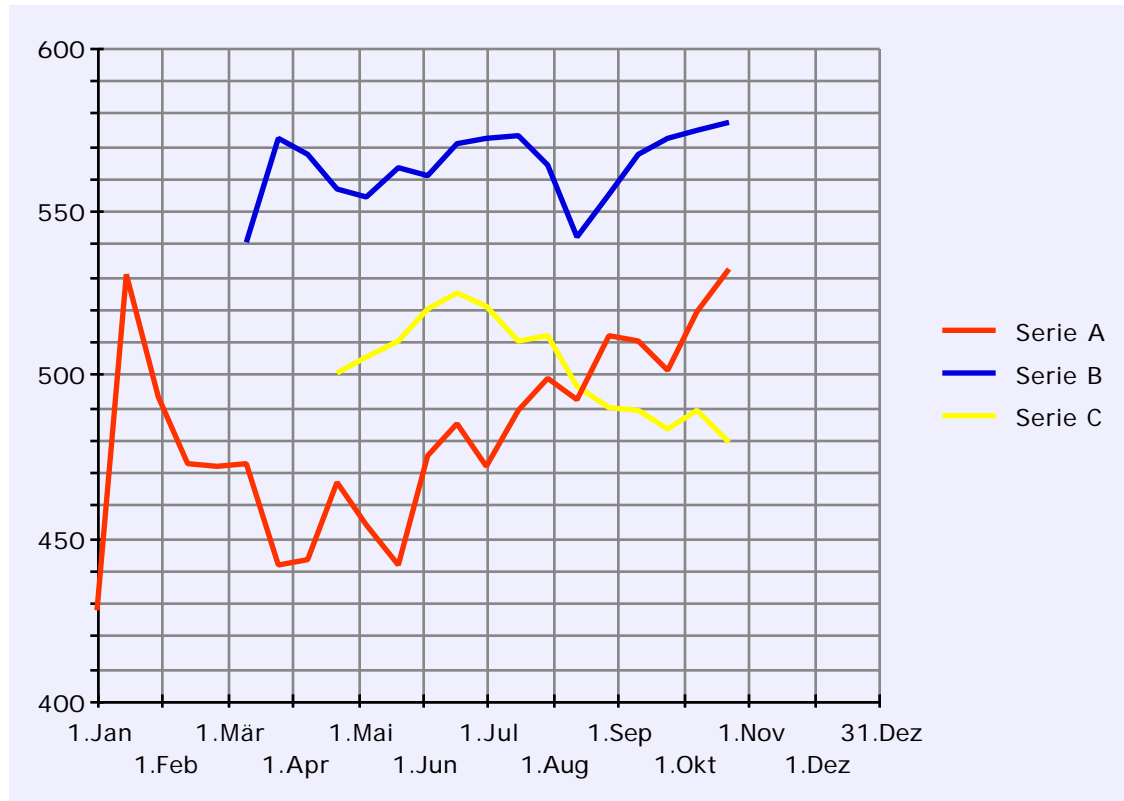


Abb. 64