

**MicroBase**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> MicroBase	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		August 23, 2022
		<i>SIGNATURE</i>

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>MicroBase</b>	<b>1</b>
1.1	MicroBase 3.0.2 © 1992-1997 by FR-Softworks	1
1.2	Weiterführende Themen	1
1.3	Datenbankdateien im- und exportieren	2
1.4	MicroBase installieren	4
1.5	Was bietet MicroBase?	5
1.6	Die Menüs von MicroBase	7
1.7	Das 'Datei'-Menü	8
1.8	Das 'Bearbeiten'-Menü	13
1.9	Das 'Ausgabe'-Menü	18
1.10	Das 'Spezial'-Menü	20
1.11	Das 'Extras'-Menü	20
1.12	Das 'Optionen'-Menü	22
1.13	Tastaturkürzel (Shortcuts)	25
1.14	Interna: Programmtechnische Informationen	27
1.15	Interner Aufbau der verwendeten Dateiformate	27
1.16	Dateinamen-Erweiterungen (Extensionen)	28
1.17	Einschränkungen dieser MicroBase-Version	29
1.18	Die MicroBase-Fehlermeldungen	30
1.19	Copyright-Bestimmungen und Registrierung	36
1.20	Weitere Produkte von FR-SoftWorks	37
1.21	Steuerzeichen in Listen und Etiketten	39
1.22	Suchkriterien	40
1.23	Operatoren für das Aktualisieren	40
1.24	MicroBase-Datenbanken in das dBASE-Format konvertieren	41
1.25	SuperBase-Datenbanken konvertieren	42
1.26	dBASE- und FoxPro-Datenbanken konvertieren	43
1.27	dBASE- und FoxPro-Datenbanken anzeigen	44
1.28	Tips & Tricks	45
1.29	Datenfelder hinzufügen oder löschen	45

---

---

1.30	Mehr als 65 535 Datensätze verwenden	46
1.31	Datenaustausch mit MicroWrite	47
1.32	Änderung des Passwortes der Datenbank	48
1.33	Datentypen und Eingabeformate	49
1.34	Der 'virtuelle Modus'	52
1.35	Datenverluste	52
1.36	Tips zum Importieren / Exportieren einer sequentiellen Datei	54
1.37	Tutorial	55
1.38	Die »rexxmathlib.library«	57
1.39	cbsi	58
1.40	Die mitgelieferten Viewer	61
1.41	Berechnungsformeln verwenden	61
1.42	Einführung in ARexx	62
1.43	Zugriff auf Felder und Konstanten	63
1.44	Verschlüsselte Datenbankdateien des alten Formats konvertieren	64
1.45	Die mitgelieferten Datenbanken	65
1.46	Relationen definieren	67
1.47	History	70
1.48	MicroBase im Mehrbenutzerbetrieb	72
1.49	Die Verwendung von ARexx-Skripten	73
1.50	Beispiel einer Problemstellung	74
1.51	Ablauf einer Skriptausführung	74
1.52	Übergebene Variablen	75
1.53	Ein Beispiel für ein Skript	75
1.54	Anwendungsbeispiele	90
1.55	Die Erstellung interaktiver Skripte	91
1.56	Die Entwicklung eines ARexx-Skriptes	96
1.57	Datenbankentwicklung	97
1.58	Zahlenformate	99
1.59	Problemlösungen	100

---

## Chapter 1

# MicroBase

### 1.1 MicroBase 3.0.2 © 1992-1997 by FR-Softworks

Installation

Übersicht

Tutorial

Datenbankentwicklung

Menüs

Tastaturkürzel

Weiterführende Themen

Tips & Tricks

Datenbankdateien im- und exportieren

Die mitgelieferten Datenbanken

History

Interna

Fehlermeldungen

Problemlösungen

Copyright und Registrierung

Werbung

### 1.2 Weiterführende Themen

---

Zu folgenden Themen sind weiterführende Informationen verfügbar:

Steuerzeichen in Listen / Etiketten

Such-Kriterien

Aktualisieren-Operatoren

Datentypen und Eingabeformate

Berechnungsformeln verwenden

Die mitgelieferten Viewer

Definition von Relationen

Der 'Virtuelle Modus'

MicroBase im Mehrbenutzerbetrieb

dBASE- und FoxPro-Datenbanken konvertieren

dBASE- und FoxPro-Datenbanken anzeigen

MicroBase-Datenbanken in das dBASE-Format konvertieren

SuperBase-Datenbanken konvertieren

Die Verwendung von ARexx-Skripten

### 1.3 Datenbankdateien im- und exportieren

MicroBase kann Datenbankdateien in folgenden Formaten im- und exportieren, indem die mitgelieferten Konvertierungsmodule benutzt werden: ↔

#### 1. Sequentielle Textdateien

Sequentielle Dateien können von jedem Programm gelesen und geschrieben werden und haben folgende Eigenschaften:

- jeder Datensatz wird in eine eigene Zeile geschrieben
  - die Datenfelder werden durch ein Trennzeichen voneinander getrennt (meist ein Komma oder ein Tabulatorzeichen) und können optional in Anführungszeichen eingefasst werden
  - die erste Zeile der Datei kann optional die Feldnamen enthalten
  - alle Eigenschaften der Felder wie Länge, Optionen und Feldtyp gehen während des Bearbeitens einer sequentiellen Textdatei verloren
-

- Textdateien können sehr schnell bearbeitet werden
- sequentielle Dateien sind kürzer als MicroBase-Datenbankdateien

Weitere Informationen finden Sie unter:

Tips zum Importieren / Exportieren einer sequentiellen Datei  
2. dBASE-Datenbankdateien

Datenbankdateien im dBASE-Format haben diese Eigenschaften:

- dBASE-Datenbankdateien können nicht von jedem Programm bearbeitet werden; oft gibt es spezielle Importfilter (Exportfilter) oder Konvertierungsprogramme
- dBASE-Datenbankdateien sind im Allgemeinen etwas länger als MicroBase-Dateien, da sie zusätzlich die Felddefinitionen enthalten
- beim Importieren von dBASE-Dateien bleibt der Feldtyp erhalten; beim Exportieren muss dies nicht immer der Fall sein
- beim Bearbeiten einer dBASE-Datenbankdatei bleiben die Felddaten erhalten, die Optionen und Standardwerte gehen verloren. Feldnamen werden beim Export abgekürzt
- das Konvertieren dauert meist länger als bei Textdateien

Weitere Informationen finden Sie unter:

dBASE- und FoxPro-Datenbanken konvertieren

MicroBase-Datenbanken in das dBASE-Format konvertieren

dBASE- und FoxPro-Datenbanken anzeigen

### 3. SuperBase

MicroBase unterstützt nur den Import von SuperBase-Datenbankdateien, nicht jedoch den Export.

SuperBase-Datenbanken haben folgende Eigenschaften:

- SuperBase-Datenbankdateien sind meist kürzer als dBASE- und MicroBase-Dateien, jedoch etwas länger als sequentielle Textdateien
- das Konvertieren benötigt auf Grund des speziellen Datenbankformates mehr Zeit als bei dBASE-Dateien.
- es können nur Informationen aus Textfeldern übernommen werden

Weitere Informationen finden Sie unter:

SuperBase-Datenbanken konvertieren  
Fazit

---

Wenn es auf die Geschwindigkeit des Ex- bzw. Importvorgangs ankommt, sind sequentielle Textdateien die beste Wahl. (Dies gilt nicht für den Import in eine neue Datenbankdatei.)

Falls die Qualität der erzeugten Datenbankdatei wichtiger ist, sollten dBASE-Datenbankdateien verwendet werden.

## 1.4 MicroBase installieren

Um MicroBase zu installieren, rufen Sie einfach das Installerskript auf.

Folgende Dateien werden installiert:

Datei	Verzeichnis
-----	-----
Programmdateien	Programmverzeichnis
CBSI	C:
Bibliotheken	LIBS:

Anforderungen an die Hard- und Software

MicroBase benötigt AmigaDOS 2.04 oder höher und mindestens 1 MB Speicher.

Da die Bildschirmmasken des Editors auf den Standardzeichensatz »Topaz 8« ausgelegt sind, sollte dieser oder ein anderer nicht proportionaler Zeichensatz mit 8 Punkten in den Voreinstellungen aktiviert werden.

Zum Verwalten großer Dateien kann mehr Speicher erforderlich sein, welchen Sie dann über das ToolType MEMORY (siehe unten) festlegen müssen. Der Speicherbedarf wird wie folgt berechnet:

Matrix :	1 Bit	
Anzahl an Datensätzen:	65.535 Sätze	
-----	-----	
Zwischensumme:	8.192 Bytes	= 65 535 / 8
Variablen:	160.000 Bytes	
-----	-----	
Speicherbedarf:	168.192 Bytes	
=====	=====	

Der Eintrag des ToolTypes kann demnach MEMORY=170 lauten.

Im sog. "virtuellen Modus" benötigt MicroBase immer 180 KB Speicher.

ToolType MEMORY=KKKKKKKKK

MicroBase unterstützt das Merkmal MEMORY=KKKKKKKKK, das den zu reservierenden Speicher in KByte angibt (Vorgabe: 180 KB).



## 1.5 Was bietet MicroBase?

MicroBase bietet Ihnen folgende Möglichkeiten:

- Import und Export von ASCII-Dateien in den Formaten SuperBase Professional 4, BECKERtext III, Documentum 2, ProWrite, TransWrite, Personal Write; Anpassung an andere Formate ist möglich
  - Flexible Funktionen zum Suchen und Ersetzen von Feldinhalten, z. B. Multiplikation von Feldinhalten mit einer Konstanten
  - Feldnamen können 20 Zeichen lang sein und Leerzeichen enthalten
  - direkter Zugriff über Datensatznummern
  - Filterfunktion
  - Ausgabe von Etiketten und Listen auf Drucker bzw. in Datei
  - Vorlage für Serienbriefe
  - Statistik-Funktionen: Durchschnitte, Summen, Minima, Maxima errechnen und anzeigen
  - Extrakt selektierter Datensätze in eine neue Datenbankdatei
  - Import und Konvertierung von dBASE-kompatiblen Datenbanken
  - dBASE-Viewer
  - Sortieren einer Datenbank direkt auf dem Datenträger (benutzt virtuellen Speicher)
  - Verschlüsseln von Datenbankdateien (verbessert in Version 2.1.3)
  - Vorgabewerte für die Feldinhalte, z. B. das aktuelle Datum
  - fortlaufende automatische Seriennummer mit definierbarem Startwert und Inkrement
  - Datentypen Text, Nummer (Zahl), Zeit, Datum, Extern (z. B. für Texte und Bilder), Berechnung und Logisch
  - Eingabeformate für Datum und Zeit
  - "virtueller Modus": Auslagern der Selektionsmatrix auf den Datenträger
  - Unterstützung dezimaler, binärer, hexadezimaler und oktaler Zahleneingaben im Bereich  $\pm 10^{309}$
  - Sortieren nach Text, Datum, Zeit sowie numerisch
  - die Datenbankstruktur kann jederzeit geändert werden
  - Hilfefunktion während der Dateneingabe (abhängig vom Feldtyp)
-

- für externe Felder sind beliebige Anzeigeprogramme wählbar
- Reorganisation beschädigter Datenbankdateien
- Berechnungen innerhalb der Datensätze
- Feldoptionen wie erlaubte Zahlenbereiche und zwingende Eingabe
- für OS 2.X werden Viewer mitgeliefert
- fehlerhafte Datenbankdateien werden beim Öffnen reorganisiert
- Ausdruck von Listen bzw. Etiketten in Dateien ist möglich
- überprüft Datenbankdateien beim Öffnen auf Beschädigungen
- Definition globaler Konstanten ist möglich
- viele Beispieldateien
- Relationen (Verknüpfungen mit anderen Datenbankdateien)
- Caches für Relationen
- einfaches Löschen von Datensätzen
- Sperrung von Datenbanken bzw. -sätzen
- Auffinden fehlerhafter Relationen
- Ausführung von ARexx-Skripten
- Schnittstelle zu "MicroChart" für einfache Diagrammerstellung
- Lernprogramm
- Zahlenformate
- phonetische Suche / Sortieren
- partielles Ersetzen
- dBASE-Export
- z. Zt. 234 Felder pro Datensatz, bis zu 2 048 Zeichen pro Datenfeld, 65 535 Datensätze pro Datei (max. 2 Milliarden im 'virtuellen Modus')
- Import in eine neue Datenbankdatei
- Listengenerator
- SuperBase-Import

Funktionsweise des Programms

MicroBase teilt eine Datenbankdatei in gleich lange Datensätze ein, die aus

---

Datenfeldern bestehen. Auf jeden beliebigen Datensatz kann über die Satznummer zugegriffen werden. Das Programm benötigt sehr wenig Speicher, weil nur jeweils ein Datensatz im Speicher gehalten wird.

Datensätze werden direkt auf dem Datenträger sortiert, indem virtueller Speicher verwendet wird.

Die Datei, die die Anzahl der vorhandenen Datensätze enthält (".PTR"), wird bei jeder Änderung eines Datensatzes aktualisiert, um das Risiko eines Datenverlustes zu minimieren.

Falls erforderlich, werden Datenbankdateien bzw. Datensätze gesperrt, damit sie nicht von mehreren Benutzern gleichzeitig verändert werden können.

Programmtechnische Informationen finden Sie weiter unten in dem Abschnitt

Intern

.

## 1.6 Die Menüs von MicroBase

MicroBase hat folgende Menüs:

```

Datei
    Dateioperationen wie Öffnen, Sichern, Löschen

Bearbeiten
    Eingeben, Ansehen, Suchen, Sortieren ...

Ausgabe
    Import, Export, Listen und Etiketten drucken

Spezial
    Statistiken, CLI, Hilfe

Extras
    Aufruf externer Module, Datensicherheit

Optionen
    Einstellungen vornehmen, öffnen und sichern
```

Es gelten folgende Vereinbarungen:

Drei Punkte (...) nach einem Menüpunkt bedeuten, dass ein Requester (eine Dialogbox) folgt. Funktionsgruppen werden durch eine "Wellenlinie" (~~~~~) voneinander getrennt.

Eine Übersicht über die Menüpunkte und ihre Shortcuts finden Sie in dem Abschnitt

Tastaturkürzel

.

## 1.7 Das 'Datei'-Menü

Neu ... Mit diesem Befehl wird eine neue Datenbankdatei ←  
erstellt. Zuvor  
wird die aktuelle Datenbankdatei geschlossen.

Folgende Schritte sind dazu erforderlich:

- 1 Wählen Sie einen Dateinamen aus. MicroBase hängt an diesen die Erweiterung für MicroBase-Datenbankdateien ".DAT" an.
- 2 Geben Sie nun bitte die erforderliche Anzahl von Datenfeldern ein.
- 3 Für jedes Datenfeld Feldbezeichnung,  
Feldtyp  
, Feldlänge und  
ggf. einen Standardwert sowie Optionen definieren.

Hinweis: Bitte beachten Sie, dass die Länge eines Datensatzes 64 KB (65 536 Bytes) nicht überschreiten darf. Wenn diese Datensatzlänge erreicht ist, können keine weiteren Datenfelder definiert werden.

### Feldtypen

Kennung	Typ	Bemerkung
T	Text	beliebige alphanumerische sowie Sonderzeichen
N	Zahl	Der Dezimalpunkt ('.') zählt mit
D	Datum	Feldlänge bitte nicht ändern!
Z	Zeit	Feldlänge bitte nicht ändern!
X	Extern	speichert Dateipfade ab
B	Berechnung	Enthält Formeln und entspricht Typ "Zahl"
L	Logisch	Enthält Wahrheitswerte. Feldlänge ist fest
R	Relation	Verknüpfung mit einem Feld einer anderen Datenbank

### Standardwerte

\$D	aktuelles Datum einfügen
\$N	Seriennummer einfügen Für die Seriennummer können Startwert und Schrittweite (Inkrement) angegeben werden. Voreingestellt ist jeweils "1". Beachten Sie, dass pro Datensatz nur eine Seriennummer möglich ist, da diese als eindeutiger Schlüssel verwendet werden soll. Die eingegebenen Zahlen werden auf die nächste ganze Zahl gerundet.
\$Z	Aktuelle Uhrzeit einfügen
\$	öffnet bei "externen" Feldern eine ASL-Auswahlbox

zur einfacheren Dateiauswahl.

Alle anderen Eingaben werden unverändert übernommen. Die maximal erlaubte Länge des Vorgabewertes beträgt 20 Zeichen.

#### Optionen

K	Eingabe wird in Kleinbuchstaben umgewandelt.
G	Umwandlung in Großbuchstaben.
N	Namensschreibweise: am Anfang und nach Leerzeichen erfolgt eine Umwandlung in Großbuchstaben.
<Zahl	Maximalwert; z. B. sind bei '<99' nur Zahlen kleiner als 99 erlaubt.
>Zahl	Minimalwert.
E	Eingabe ist zwingend erforderlich, die Eingabemaske kann sonst nicht verlassen werden.

#### Optionen

Bei den Optionen wird die Groß-/Kleinschreibung unterschieden; max. 12 Zeichen können eingegeben werden. Die Optionen können miteinander kombiniert werden (bei K, G, N ist dies natürlich nicht sinnvoll.) Bei Namensfeldern in Adressdatenbanken kann man z. B. 'EN' verwenden.

#### Zahlenformate

Bei numerischen und Berechnungsfeldern können Sie noch ein

Zahlenformat angeben, das beim Ausdrucken von Listen bzw. Etiketten verwendet wird. Dies gilt entsprechend für Relationsfelder, die einen numerischen Wert beinhalten.

#### Standardfelder

MicroBase erstellt standardmäßig Textfelder mit 20 Zeichen Länge, die die Bezeichnung "Feld " zuzüglich der Feldnummer haben - also z. B. "Feld 4". Unbekannte Feldtypen werden durch den Typ Text ersetzt.

#### Feldlängen

Bitte beachten Sie, dass die Feldlängen der Typen "Datum", "Relation", "Zeit" und "Logisch" fest sind. Sie sollten im Hinblick auf die Kompatibilität mit zukünftigen Programmversionen nicht geändert werden.

Die Länge von Feldern der Typen "Nummer" und "Berechnung" darf 19 Stellen nicht überschreiten, da dies in dBASE die Beschränkung für Zahlenfelder ist.

#### Hilfefunktion

Drücken Sie während der Eingabe der Feldtypen und Wertvorgaben auf <Help>, um die Hilfefunktion zu aktivieren.

## Externe Felder und Berechnungen

Bei Feldern des Typs "Extern" müssen Sie zusätzlich einen Viewer eingeben, bei "Berechnungs"-Feldern eine Formel. Beide Angaben können bis zu 2 048 Zeichen lang sein.

**Viewer:** Unter OS 3.X wird standardmäßig »MultiView« als Anzeigeprogramm benutzt. Sie können jedoch jedes andere Programm angeben (wenn Sie z. B. »C:Ed« benutzen, haben Sie ein Äquivalent zu einem MEMO-Feld auf dem PC.) Für OS 2.X werden einige Viewer mitgeliefert.

**Formel:** Sie können alle Funktionen von ARexx benutzen; die mitgelieferte »rexxmathlib.library« bietet zusätzliche mathematische Funktionen.

## Relationen

Die Erstellung von Relationen entnehmen Sie bitte dem gesonderten Abschnitt.

- 4 Danach werden Sie nach einem optionalen Passwort gefragt, mit dem die Datenbank verschlüsselt werden soll. Geben Sie <Escape> und <Return> ein, wenn eine Verschlüsselung nicht gewünscht ist.

Alle Leerzeichen am Anfang bzw. Ende des eingegebenen Passwortes werden entfernt. Groß- und Kleinschreibung werden unterschieden. Sie sollten möglichst Passwörter verwenden, die aus mehreren durch Leerzeichen getrennten Begriffen bestehen.

Es wird empfohlen, mindestens sechs Zeichen lange Passwörter zu benutzen, um eine maximale Datensicherheit gewährleisten zu können (bis zu 50 Zeichen lange Passwörter sind erlaubt).

- 5 Nach einer letzten Sicherheitsabfrage wird die Datenbankdatei erstellt; sie besteht aus folgenden Dateien (der Dateiname wird mit 'Beispiel' angenommen):

'Beispiel.DAT': Die eigentliche Datenbankdatei.  
'Beispiel.DEF': Die Definitionsdatei für die Datenfelder  
'Beispiel.PTR': Die Zeigerdatei, die die maximale und tatsächliche Anzahl der Datensätze angibt.

Falls die Datenbankdatei schon existieren sollte, wird eine entsprechende Fehlermeldung angezeigt.

öffnen ... Sie können in einer Dateiauswahlbox eine Datei auswählen, die Sie öffnen möchten.

## Demoversion

Ohne Keyfile können nur Datenbankdateien mit bis zu 20 Datensätzen

geladen werden; der 'virtuelle Modus' ist nicht möglich.

#### Passwörter

Sie müssen immer ein Passwort eingeben; es kann jedoch "leer" sein, falls die Datei unverschlüsselt ist.

Falls das Passwort falsch ist, erscheinen auf dem Bildschirm nur "wirre Zeichen" anstatt der Datensatzinhalte. In einigen Fällen kann sogar die Eingabemaske nicht mehr korrekt angezeigt werden.

Sie müssen es dann UNBEDINGT vermeiden, Datensätze zu edieren oder hinzuzufügen.

Andernfalls müssen Sie mit DATENVERLUSTEN rechnen.

Falls das Passwort nicht validiert werden kann, wird die Datenbankdatei geschlossen.

#### Automatische Datenbankreparatur

Fehlende Zeigerdateien (.PTR) werden automatisch erzeugt. Die Anzahl der Datensätze wird errechnet, die maximal mögliche Anzahl der Datensätze auf den Standardwert gesetzt. Dieser Wert kann jedoch von Ihnen geändert werden.

Fehlende Definitionsdateien werden ebenfalls automatisch erzeugt. Da die Datenbankstruktur unbekannt ist, erstellt MicroBase nur ein einziges Text-Datenfeld "Lazarus", das so lang ist wie ein Datensatz. Auf diese Weise ist ein Zugriff auf die Datenbankdatei möglich. Sie sollten jedoch die vorhandenen Datensätze exportieren, um einen Datenverlust zu vermeiden.

Falls eine Datenbankdatei zwar definiert, aber noch nicht vorhanden ist, wird sie von MicroBase neu erstellt. Sie müssen sie dann erneut öffnen, um sie verwenden zu können.

Fehlerhafte Datenbankdateien werden vor dem Öffnen automatisch reorganisiert. Eine Datenbankdatei wird als fehlerhaft erkannt, wenn die in der Zeigerdatei angegebene Anzahl von Datensätzen nicht zu der Dateilänge der Datenbankdatei passt.

#### Prüfung auf Lesbarkeit

Falls der erste Datensatz der Datenbank nicht korrekt gelesen werden kann, erscheint eine Fehlermeldung. Die Ursache des Fehlers ist entweder ein falsches Passwort oder eine Beschädigung der Datenbank. Sie sollten sich den ersten Datensatz ansehen und ggf. die Datenbankdatei reorganisieren lassen, falls der überwiegende Teil des Datensatzes doch noch lesbar ist.

#### 'Virtueller Modus'

Falls die zu öffnende Datenbank schon von einem anderen Benutzer bearbeitet wird, ist der 'virtuelle Modus' nicht verfügbar. Das Programm aktiviert dann den Standardmodus.

Im 'virtuellen Modus' wird nach dem Öffnen einer Datei zuerst die Auslagerungsdatei aufgebaut. Dieser Vorgang nimmt im Allgemeinen nur sehr wenig Zeit in Anspruch.

#### Relationscaches

Sollten die Cachedateien für Relationsfelder noch nicht existieren, werden sie jetzt erzeugt.

**schließen** Die aktuelle Datei wird geschlossen, die ".PTR"-Datei gespeichert und im 'virtuellen Modus' die Auslagerungsdatei gelöscht. Die aktuellen Einstellungen für Viewer, Formeln usw. werden auf ihre Standardwerte zurückgesetzt (d. h., sie werden gelöscht).

Alle Datenbankdateien werden vor dem Öffnen einer anderen oder am Programmende automatisch geschlossen.

**Hinweis:** Das Schließen der Datenbankdatei benötigt mehrere Sekunden, da auf den Abschluss aller erforderlichen Datenträgerzugriffe gewartet wird.

**löschen ...** Nach einer Sicherheitsabfrage wird die ausgewählte Datei gelöscht, außerdem die gleichnamigen Dateien mit den Extensionen .CCY, .CLC, .CMD, .DEF, .NUM, .OPT, .PRE, .PTR, .REL, .SYS, .VM und .VM2, falls diese ebenfalls vorhanden sind. Vorhandene Relationscaches (.1, .2 usw.) werden ebenfalls gelöscht.

**Info** Zeigt die Versionsnummer und das Revisionsdatum an und bei einer geöffneten Datenbankdatei zusätzlich folgende Informationen:

1 Namen der Datenbankdatei, der Abfrage und der Selektionsmatrix (jeweils nur, falls vorhanden)

2 Anzahl der in der Datenbank vorhandenen Datensätze

3 Anzahl der selektierten Datensätze (absolut und im Verhältnis zur Gesamtanzahl)

4 Status der Selektion: aktiv oder inaktiv

5 Freie Ressourcen: Dies ist der Anteil des freien RAM am reservierten Speicher. Reservieren Sie mehr Speicher, wenn der Wert unter ca. 20 Prozent sinkt (bzw. aktivieren Sie den 'virtuellen Modus', falls dies noch nicht geschehen ist.)

6 Anzahl der Bytes, die im 'virtuellen Modus' auf den Datenträger ausgelagert wurden (0 = deaktiviert.)

Bei der Demo-Version erscheint ein entsprechender Hinweis: die Anzahl der Datensätze ist begrenzt, der 'virtuelle Modus' nicht möglich.

**Ende** Nach einer Sicherheitsabfrage wird das Programm beendet. Zuvor wird die Datenbankdatei geschlossen.



## 1.8 Das 'Bearbeiten'-Menü

Eingeben ... Die eingegebenen Datensätze werden an die Datei ←  
angehängt,  
wenn Sie den Menüpunkt "Bearbeiten/Eingeben" aufrufen. Das  
Edieren bestehender Datensätze ist hier nicht möglich.

Die Eingabe der Datensätze geschieht seitenweise; es  
werden bis zu 24 Datenfelder auf dem Bildschirm angezeigt.

In der Datenmaske werden die definierten Vorgabewerte au-  
tomatisch angezeigt und können so übernommen werden.  
Optional werden zusätzlich die Eingabeformate angezeigt.

Bei der Eingabe ist grundsätzlich der Einfügemodus akti-  
viert. Wenn Sie jedoch etwas in Felder der Formate "Datum"  
oder "Zeit" eingeben, wird der Überschreibmodus aktiviert,  
falls Sie die Anzeige von Eingabeformaten eingeschaltet  
haben.

MicroBase entfernt Leerzeichen am Anfang sowie Ende des  
Feldes automatisch. Doppelte Anführungszeichen (") werden  
in einfache (') umgewandelt, da doppelte Anführungszei-  
chen in ARexx-Skripten als Stringbegrenzer benutzt werden.

### Tastaturbefehle

<Return>	Eingabe übernehmen.
<Buchstabe>	Im Einfügemodus wird der Buchstabe vor dem aktuellen Zeichen eingefügt, im Überschreib- modus wird das Zeichen an der Cursorposition überschrieben. Der Cursor bewegt sich in jedem Fall nach um eine Position nach rechts.
<links>	Bewegt den Cursor um eine Position nach links.
<rechts>	Bewegt den Cursor eine Position nach rechts.
<Backspace>	Löscht das Zeichen vor dem Cursor und bewegt diesen um eine Position nach links.
<Del>	Löscht das Zeichen an der Cursorposition.
<CTRL-V>	Schaltet zwischen Einfüge- und Überschreib- modus um.
<Help>	Ein vom Feldtyp abhängiger Hilfetext wird ein- geblendet. Dieser gibt Ihnen Hinweise auf die akzeptierten Feldinhalte.
<Escape>	Löscht das Eingabefeld und setzt den Cursor an dessen Beginn.

### Lange Datenfelder

Beachten Sie bitte, dass bei Datenfeldern, die mehr als 53  
Zeichen aufnehmen können, der Cursor immer am Anfang des  
Eingabefeldes steht. Sie können sich jedoch trotzdem wie  
beschrieben im Feld bewegen; auch die anderen Tastatur-  
befehle stehen wie gewohnt zur Verfügung.

### Konsistenzprüfung

Bei falschen Eingaben erscheint eine Fehlermeldung, die die Art des Fehlers anzeigt (z. B. 'Stunde muss im Bereich 00..23 sein.'). Bei Zahlen erscheint außerdem ein Fragezeichen('?') an der vermuteten Fehlerstelle.

#### Zahleneingabe

Zahlen können Sie in MicroBase im dezimalen, hexadezimalen, binären und oktalen Format eingeben. Die Wertebereiche sind dabei wie folgt:

dezimal, wissenschaftlich:  $-10^{309}..+10^{309}$   
binär, oktal, hexadezimal:  $-2^{31}..+2^{31}-1$

Kommazahlen sind nur im dezimalen bzw. wissenschaftlichen Format erlaubt; dort sind sie mit Dezimalpunkt ('.') einzugeben. Das Dezimalkomma wird von MicroBase nicht unterstützt und automatisch in einen Dezimalpunkt umgewandelt. (Das Dezimalkomma wird nur in Zahlenformaten unterstützt.)

#### Externe Felder

Wenn Sie ein Dollarzeichen ('\$') eingeben, öffnet MicroBase einen Dateirequester.

#### Abgeleitete Datenfelder

Felder des Typs "Berechnung" können nicht ediert werden, da ihr Inhalt erst nach der Eingabe aller Felder errechnet wird. Dies gilt entsprechend auch bei "Relationen".

Für Relationsfelder wird ein Eintrag in der entsprechenden Cache-Datei erzeugt.

#### Mehrbenutzerbetrieb

Wenn der Datensatz schon von einem anderen Benutzer bearbeitet wird, erhalten Sie eine Fehlermeldung. Sie müssen dann warten, bis der Datensatz wieder verfügbar ist.

#### Referenzintegrität

Wenn die Option "Strikte Referenzintegrität" aktiviert wurde, können Sie den Datensatz erst speichern, wenn er die referenzielle Integrität erfüllt.

#### Systemdatei

Falls die Systemdatei (".SYS") mit den Versionsangaben noch nicht existiert, wird sie erstellt. Sie wird für die Kompatibilität mit zukünftigen Programmversionen benötigt.

#### Automatische Selektion

Wenn Sie zuvor den Menüpunkt "Bearbeiten/Alles selekt."

---

aufgerufen hatten, wird der neu eingegebene Datensatz automatisch selektiert.

Demoversion

Ohne Keyfile können maximal 20 Datensätze eingegeben werden.

Ansehen

Mit diesem Befehl werden die (selektierten) Sätze angezeigt und ediert. Folgende Tasten werden erkannt:

```

Cursor_rechts: 1 Satz weiter
Cursor_links:  1 Satz zurück
Cursor_runter: 10 Sätze vor
Cursor_hoch:   10 Sätze zurück
e              Aktuellen Satz bearbeiten (siehe oben)
s              "          "      selektieren   = in
              "          "      Matrix aufnehmen
d              "          "      de-selektieren = aus
              "          "      Matrix entfernen
b              Begriff suchen in selektierten Sätzen (ab
              nächstem Satz). Die Optionen zur Groß- und
              Kleinschreibung sowie zur phonetischen
              Suche werden berücksichtigt.
g              'Gehe zu' = Sprung zu einem anderen
              Datensatz
v              Erlaubt Ihnen, die Nummer des anzuzeigen-
              den Datenfeldes einzugeben.
CTRL-A bis    übergibt den Inhalt des Datenfeldes an
CTRL-Y        den Viewer (CTRL-A ist für Feld 1 usw.)
              (Die Nummer der aktiven Seite wird dabei
              berücksichtigt.)
ESC           'Abbruch' = Das Edieren der Sätze wird
              beendet
0 bis 9       schaltet auf die entsprechende Seite um;
              jede Seite besteht aus 24 Datenfeldern.
              Dieses Verfahren ermöglicht das Anzeigen
              von mehr als 25 Datenfeldern.

```

Falls auf dem Bildschirm nur wirre Zeichen erscheinen, haben Sie entweder das Passwort falsch eingegeben oder die Datenbankdatei muss reorganisiert (d. h. repariert) werden.

Je nach den gewählten Optionen werden Relationsfelder normal, kursiv formatiert oder gar nicht angezeigt.

Löschen ...

Nach einer Sicherheitsabfrage werden alle selektierten Datensätze gelöscht. Die Selektion wird entfernt; der Satzzeiger auf Datensatz 1 positioniert.

Das Löschen von Datensätzen verändert nicht die Länge der Datenbankdatei. Die überflüssigen Daten am Ende der Datei werden mit Leerzeichen überschrieben.

Im Gegensatz zu einigen anderen Datenbankverwaltungssystemen - z. B. dBASE - werden die Datensätze nicht nur

logisch, sondern auch physikalisch gelöscht.

Die Sortierung der Datenbank geht durch diesen Vorgang verloren.

Falls Sie nicht die "strikte Referenzintegrität" aktiviert haben, verlieren die Relationscaches ihre Gültigkeit.

Sortieren ...

Hiermit wird die Datenbankdatei nach dem Inhalt eines Datenfeldes sortiert.

Folgende Sortiermethoden stehen zur Verfügung:

- [T]Text: Die Feldinhalte werden nach ihrem ASCII-Wert sortiert. Umlaute werden zuvor in A, O, U bzw. SS gewandelt, um korrekt sortiert werden zu können (Bis Vers. 1.4.8 wurden sie in AE usw. gewandelt. Dies entsprach jedoch nicht der DIN.)
- [N]umerisch: Es wird nach dem Zahlenwert sortiert.
- [D]atum: Die Feldinhalte werden als Datumsangaben interpretiert. Das Datum wird in die Tageszahl des julianischen Datums umgewandelt und als ganze Zahl (Integer) behandelt.
- [Z]eit: Es wird nach der Uhrzeit sortiert, indem die Zeitangabe in die Sekundenzahl des julianischen Datums umgewandelt wird.
- E[X]tern: Es wird nach dem Dateinamen sortiert.
- [B]erechnung: Das Ergebnis wird wie eine Zahl sortiert.
- [L]ogisch: Die Sortierung erfolgt wie bei Texten.
- [R]elation: Sortierung wie bei Texten.

MicroBase erstellt zuerst die Auslagerungsdatei für den virtuellen Speicher und beginnt dann erst zu sortieren. Nur der Start des Sortiervorgangs wird mitgeteilt; eine Fortlaufsanzeige ist wegen des verwendeten Sortierverfahrens (Quicksort) leider nicht möglich. Nach dem Sortiervorgang löscht MicroBase automatisch die Auslagerungsdatei.

Sie können Daten auch absteigend sortieren lassen; dies nimmt bei Texten jedoch ein wenig mehr Zeit in Anspruch als die Standardsortierrichtung "aufwärts", da alle Zeichen vor dem Sortieren geändert werden müssen. Bei den anderen Sortiermethoden wird die resultierende Zahl (z. B. das julianische Datum) einfach mit (-1) multipliziert.

Groß/Kleinschreibung wird normalerweise nicht unterschieden. Dies kann jedoch über die Optionen geändert werden; dann wird streng nach ASCII-Codewerten sortiert und Umlaute werden am Ende der Datenbankdatei einsortiert.

Wurde das phonetische Suchen aktiviert, werden die Textfelder nach ihrem Lautwert sortiert. Der Sortiervorgang wird dadurch verlangsamt.

Falls Sie nicht die "strikte Referenzintegrität" akti-

viert haben, verlieren die Relationscaches ihre Gültigkeit.

Zum Sortieren wird virtueller Speicher verwendet. Die erzeugte temporäre Datei hat den gleichen Namen wie die Datenbankdatei, jedoch die Extension ".VM" statt ".DAT".

Beachten Sie bitte, dass diese Datei unter Umständen sehr lang werden kann: Bei einer Datenbank mit 65 535 Datensätzen z. B. bis zu 128 MB!

Versuchen Sie unter keinen Umständen, die temporäre Datei auf der RAM-Disk zu erzeugen!

Dies könnte das System zum Absturz bringen.

Suchen ...

Es kann nach verschiedenen Kriterien gesucht werden. (Die Suche nach einem leeren Feld ist immer erfolgreich.) Hier ist es möglich, einen Wert für die Suchgenauigkeit einzugeben: dies ist die Anzahl der Bedingungen, die mindestens erfüllt sein müssen.

Die Eingabe geschieht seitenweise, indem jeweils 24 Datenfelder auf dem Bildschirm angezeigt werden.

Standardmäßig wird dabei die Groß/Kleinschreibung ignoriert und die phonetische Suche deaktiviert.

Klicken Sie hier für eine Übersicht über die verfügbaren

Suchkriterien

.

Aktualisieren ...

Die gefundenen Sätze werden geändert. Dies ist in etwa eine Ersetzen-Funktion, aber wesentlich leistungsfähiger.

Sie können wählen, ob eine Neuberechnung der Formeln erfolgen soll oder nicht.

MicroBase ersetzt standardmäßig den kompletten Feldinhalt durch den Ersatzbegriff. Wenn Sie die Option "partielles Ersetzen" aktiviert haben, wird nur der mit dem Suchbegriff übereinstimmende Teil des Feldinhaltes ersetzt. (Das "partielle Ersetzen" wird nur dann durchgeführt, wenn Sie keine Suchkriterien angegeben haben.)

Falls Sie nicht die "strikte Referenzintegrität" aktiviert haben, verlieren die Relationscaches ihre Gültigkeit.

Klicken Sie hier für eine Übersicht über die möglichen

Aktualisieren-Operatoren

.

sichern ...

Die aktuelle Abfrage wird gespeichert (unter einer Abfrage versteht MicroBase die Einträge in den Such- und Aktuali-

sierungsmasken). Die Datei bekommt die Extension ".MBA".

Es wird empfohlen, die Abfragedateien fortlaufend zu nummerieren (z. B. Videos1.MBA).

öffnen ...	Lädt eine zuvor gespeicherte Abfragendefinition.  Beachten Sie bitte, dass die Aktualisierungsangaben ebenfalls neu geladen werden.
zurücksetzen ...	Löscht - falls gewünscht - die aktuelle Abfrage.
Selektion	Ist die Selektion aktiv, werden nur noch die Sätze beachtet, die in der Matrix stehen. Dies gilt nicht für Befehle, die die Selektionsmatrix verändern (Suchen und Aktualisieren) und für das Sortieren.
Alles selekt.	Alle Datensätze werden selektiert.
Sel.-Matrix	Selektionsmatrizen können auf einem Datenträger archiviert bzw. aus dem Speicher entfernt werden.

## 1.9 Das 'Ausgabe'-Menü

Import ... Sie können wählen, ob die Datei in die aktuelle ←  
oder in  
eine neue Datenbankdatei importiert werden soll.

Das Trennzeichen, das die Datenfelder voneinander trennt, kann eingegeben werden. Vorgabe ist ein Tabulatorzeichen.

Näheres zu den Trennzeichen siehe "Export ..." weiter unten.

Fall 1: Import in die aktuelle Datenbank

Es wird nur die maximal mögliche Anzahl von Datensätzen importiert. Die überzähligen werden ggf. ignoriert.

Die zu importierende Datei darf nicht mehr Felder haben als die aktuelle Datenbankdatei. Der erste Datensatz kann überlesen werden, falls er Feldnamen enthält.

Falls Sie nicht die "strikte Referenzintegrität" aktiviert haben, verlieren die Relationscaches ihre Gültigkeit.

Fall 2: Import in eine neue Datenbank

Aus der sequentiellen Datei wird eine neue Datenbankdatei erstellt.

Im ersten Schritt werden die Feldnamen und -längen ermittelt (falls der erste Datensatz keine Feldnamen enthält, werden Standardnamen wie "Feld 1" usw.

benutzt).

Im zweiten Schritt werden die Datensätze in die neue Datenbankdatei kopiert.

Bei aktiver Datenbanksperre wird das Importieren einer Datei verweigert.

Export ...

Mit diesen Unterprogrammen können Daten mit sequentiellen Dateien ausgetauscht werden, z. B. mit ProWrite und Super-Base Professional 4. Die Dateien müssen hierzu im ASCII-Format vorliegen. Das Trennzeichen für die Datenfelder ist wählbar. Außerdem kann festgelegt werden, ob die Feldbezeichnungen auch gespeichert werden sollen.

Bei MicroBase dürfen die Trennzeichen nicht in den eigentlichen Daten vorkommen. Dies gilt auch, wenn die Datenfelder zusätzlich durch Anführungszeichen begrenzt werden.

Geben Sie den ASCII-Code des Trennzeichens ein oder stellen Sie das Zeichen ' ' vor das gewünschte Zeichen.

Beispiele

-----

'9' Das Tabulatorzeichen (ASCII 9) wird benutzt. Dies ist Standard, da so die besten Resultate erzielt werden können.

'\',' Das Trennzeichen ist das Komma.

Weitere Informationen finden Sie unter:

Datenbankdateien im- und exportieren

Extrakt ... Die selektierten Datensätze werden in eine neue ←

Datenbankdatei geschrieben. Eine evtl. schon bestehende Datei wird dabei überschrieben. Die Datenbank erhält das Passwort der aktiven Datenbank, auch die gleichen Standardwerte und Seriennummern usw.

Dies ist natürlich auch eine hervorragende Methode, Sicherungskopien Ihrer Datenbankdateien anzufertigen.

Caches für Relationsdateien werden nicht gesichert. Sie werden neu aufgebaut, wenn Sie die Datensätze der erzeugten Datei das erste Mal betrachten oder bearbeiten.

Listen /  
Etiketten

Hier können Listen oder Etiketten erstellt werden. Etiketten sind immer neun Zeilen lang, die Zeilanzahl von Listen kann von 1 bis 200 frei gewählt werden.

Beide Listenarten werden normalerweise auf dem Drucker ausgegeben. Geben Sie '\$' ein, um über einen ASL-Requester eine Ausgabedatei wählen zu können. Wenn Sie 'W' eingeben, wird die Ausgabe in ein Fenster umgeleitet. Dies ist z. B.

für eine Druckvorschau nützlich.

Es wird empfohlen, die Listen-, Etiketten- und Ausgabe-dateien analog zu den Abfragedateien fortlaufend zu nummerieren - etwa Spielfilmel.MBO.

Im Verzeichnis "Utils" wird ein Listengenerator zur Verfügung gestellt.

Klicken Sie hier für eine Übersicht über die zur Verfügung stehenden Steuerzeichen in Listen .

In den Listen bzw. Etiketten werden ggf. die von Ihnen definierten Zahlenformate verwendet.

In der Liste »Beispiel.SAL« sehen Sie, wie mit MicroBase ein Serienbrief erstellt wird.

## 1.10 Das 'Spezial'-Menü

Neues CLI ...	Ein neues CLI wird geöffnet.  Durch Eingabe von "EndCLi" kehren Sie zu MicroBase zurück.
Statistiken	Von den selektierten Datensätzen wird einer der folgenden Werte ermittelt: Gesamtsumme, Minimum, Maximum oder Durchschnitt.  Die Grenzwerte sind hierbei $-10^{309}$ und $+10^{309}$ . Werte außerhalb dieser Grenzen werden nicht berücksichtigt.
Hilfe	Das Programm »SYS:Utilities/MultiView« aufgerufen, um dieses Dokument anzuzeigen.

## 1.11 Das 'Extras'-Menü

dBASE-Datenbanken Erlaubt das Ansehen  
,  
Importieren  
und  
Exportieren  
von zu dBASE kompatiblen Datenbankdateien.

Dies geschieht über den Aufruf des entsprechenden Tools im "Module"-Verzeichnis.

Systeminformation Ruft das Programm  
CBSI  
im Verzeichnis SYS:C auf.



Es werden unter anderem Informationen zum Prozessor, dem verwendeten Chipsatz (z. B. AA) und der Geschwindigkeit des Computers angezeigt. Um korrekte Geschwindigkeitswerte zu erreichen, sollte das Programm verständlicherweise allein laufen, um nicht durch andere Anwendungen beeinflusst zu werden.

Reorganisieren ... Versucht, eine beschädigte Datenbankdatei zu reparieren (zum Beispiel nach einem Stromausfall). Nicht lesbare Zeichen werden dabei entfernt.

Dieser Vorgang nimmt einige Zeit in Anspruch, da alle Datensätze komplett bearbeitet werden müssen.

Bei den folgenden Felddtypen wird ggf. das erforderliche Format wieder hergestellt:

Nummer	Ab dem ersten nicht numerischen Zeichen wird der Rest des Feldinhaltes ignoriert, überflüssige Leerzeichen werden entfernt.
Datum	Die erforderlichen Punkte ('.') werden eingefügt.
Zeit	Wie oben, es werden jedoch Doppelpunkte eingefügt.
Berechnung	Neuberechnung aller Formeln.

Wertefehler in Feldern der Typen "Datum" und "Zeit" wie z. B. »31.02.1996« werden nicht untersucht. Dies bedeutet, dass sie erst bei der Änderung von Datensätzen darauf aufmerksam gemacht werden.

Doppelte Anführungszeichen werden in einfache umgewandelt.

Alle Formeln werden neu berechnet. Relationsfelder werden ignoriert, da sie nicht in der Datenbank gespeichert sind.

Falls Sie die "strikte Referenzintegrität" aktiviert haben, werden die Relationscaches neu aufgebaut.

Caches aktualis. MicroBase verwendet Caches, um den Zugriff auf verbundene Datenbanken zu beschleunigen. Diese Cachedateien werden bei jedem Zugriff auf einen Datensatz aktualisiert.

Sie sollten diesen Menüpunkt vor zeitaufwendigen Operationen (z. B. Suchen; Exportieren) aufrufen, um die Caches auf den aktuellen Stand zu bringen.

Beim Löschen und Sortieren von Datensätzen werden die Cacheeinträge ungültig und müssen neu aufgebaut werden.

Die Aktualisierung der Caches ist die Grundlage der Option "strikte Referenzintegrität" und wird nach jeder Operation ausgeführt, die Daten in der Datenbankdatei ändert. Die "strikte Referenzintegrität" kann zwar deaktiviert werden, aber dies wird aus Gründen der

Datenintegrität nicht empfohlen.

Während der Aktualisierung der Verknüpfungsdaten werden keine Meldungen über fehlerhafte Relationen angezeigt. Im Anschluss an die Überarbeitung der Caches wird die Anzahl ungültiger Cacheeinträge mitgeteilt.

Beim Neuaufbau der Relationscaches werden die Datensätze mit fehlerhaften Verknüpfungen automatisch selektiert. Sie können sie dann ganz einfach an die gegebenen Daten in der zweiten Datenbank anpassen oder sie löschen, falls keine Entsprechungen mehr vorhanden sind.

Skript ausführen Sie können ein ARexx-Skript ausführen lassen.

Diagramm erst. ... Mit diesem Menüpunkt erhalten Sie die Möglichkeit, Projektdateien für den Diagrammgenerator "MicroChart" erzeugen zu lassen.

MicroBase fordert Sie nach der Auswahl des Dateinamens auf, zwei Felder anzugeben, aus denen das Diagramm aufgebaut werden soll. Ein Feld muss die darzustellenden Werte enthalten, das andere die zugehörige Beschriftung. (Wenn Sie als Beschriftungsfeld "0" eingeben, werden die Daten fortlaufend nummeriert.)

Anmerkung: Die erzeugte Projektdatei enthält nur die Daten und keine Einstellungen wie z. B. das Format des Diagrammbildschirms oder den zu verwendenden Zeichensatz.

Diese Einstellungen können Sie nachträglich in MicroChart vornehmen.

Es werden nur die selektierten Datensätze berücksichtigt.

Das Programm "MicroChart" finden Sie in einem separaten Archiv.

## 1.12 Das 'Optionen'-Menü

festlegen ... Folgende Eigenschaften können festgelegt werden:

- Eingabeformate anzeigen (J|N): zeigt bei der Eingabe in Zeit- oder Datumsfelder eine Schablone an (' : :' bzw. ' . .') Standard ist: [N]ein.
- Groß/Kleinschreibung beim Suchen ignorieren (J|N): Die Vorgabe ist [J]a. Dies verlangsamt den Suchvorgang nur unwesentlich. Kleinbuchstaben werden dann wie Großbuchstaben behandelt.
- Virtueller Modus (J|N): Standard ist [N]ein. Die Einstel-

lung wirkt sich nicht auf die aktuelle Datenbankdatei aus; sie wird erst beim Öffnen bzw. Erstellen der nächsten Datenbank beachtet.

- **Relationsfelder hervorheben:** Wenn Sie diese Option aktivieren, werden Relationsfelder in der Ansicht kursiv dargestellt (Vorgabe). Dies wirkt sich natürlich nicht auf den Ausdruck aus.
- **Relationsfelder ignorieren:** Diese Option ist normalerweise deaktiviert. Wenn Sie die Option einschalten, werden die Relationsfelder beim Blättern durch die Datenbank nicht mehr angezeigt (dies beschleunigt das Blättern durch die Datenbankdatei). Beim Suchen, Drucken usw. werden die Relationsfelder weiterhin aktualisiert und ausgewertet.
- **Zugriffssperre aktiv:** Standardmäßig sperrt MicroBase Datenbanken bzw. -sätze, wenn eine Datenbankdatei von mehreren Benutzern geöffnet wurde. Die Deaktivierung dieser Option beschleunigt den Beginn der sich auf die gesamte Datenbank beziehenden Operationen (z. B. das Sortieren). Standard ist [J].
- **Kompatibilitätsmodus:** Wenn Sie den Kompatibilitätsmodus aktivieren, wird das alte Zugriffsverfahren aus den Programmversionen vor 2.2.2 verwendet. Dies ist nur notwendig, wenn am Anfang oder Ende eines Datenfeldes Zeichen wie 'ÿ' erscheinen oder Fehler beim Sortieren von Datenbankdateien auftreten. Der Kompatibilitätsmodus verlangsamt den Datenbankzugriff nur unwesentlich. Diese Option gilt auch für Relationen. Vorgabe ist [N].
- **Strikte Referenzintegrität:** Wenn Sie diese Option aktivieren, akzeptiert MicroBase bei der Dateneingabe nur Datensätze, die die referenzielle Integrität erfüllen (verlangsamt die Dateneingabe). Datensätze, die die Referenzintegrität verletzen, müssen erneut eingegeben werden. Nach dem Sortieren, Löschen, Aktualisieren und Importieren von Datensätzen werden die Relationscaches aktualisiert. Diese Option ist standardmäßig aktiviert.
- **Phonetische Suche:** Wenn Sie diese Option aktivieren, wandelt MicroBase beim Suchen und Sortieren alle Daten in die entsprechenden Lautwerte um. Eine Suche nach "Meier" findet dann auch "Mayr" und "Maier". Diese Option ist standardmäßig deaktiviert.

Hinweis: Die Aktivierung dieser Option kann bei dem Sortieren einer Datenbank zu unerwarteten Ergebnissen führen – so kann z. B. der Name "Müller" vor "Miller" einsortiert werden, da beide denselben Lautwert besitzen!

Die phonetische Suche verlangsamt den Suchvorgang, da alle betroffenen Datenfelder zeichenweise umgewandelt

werden müssen.

- **Partielles Ersetzen:** Wenn Sie diese Option aktivieren, ersetzt MicroBase nur den mit dem Suchbegriff übereinstimmenden Teil des Feldinhaltes durch den Ersatzbegriff. Standardmäßig wird der komplette Feldinhalt ersetzt. (Das "partielle Ersetzen" wird nur dann durchgeführt, wenn Sie keine Suchkriterien angegeben haben.)

(Wenn Sie die Standardeinstellungen wieder herstellen möchten, können Sie auch vor dem Start von MicroBase die Datei »Standard.MBP« im Programmverzeichnis löschen.)

öffnen ...	Lädt eine Einstellungsdatei nach. Diese muss nicht unbedingt alle Optionen definieren.																
sichern	Speichert die Datei »Standard.MBP« im Programmverzeichnis, die bei jedem Programmstart automatisch geladen wird.																
sichern als ...	Speichert die Einstellungen in einer Datei mit der Endung ".MBP" ab.																
Struktur ändern ...	Hier haben Sie die Möglichkeit, die Struktur der Datenbank in begrenztem Maße nachträglich zu ändern. Drücken Sie auf <Help>, um zu einem Eingabefeld eine Hilfestellung zu erhalten.																
Feldname	Sie können eine bis zu 20 Zeichen lange Feldbezeichnung eingeben.																
Feldtyp	Folgende Feldtypen sind möglich: <table> <tr> <td>Text</td> <td>beliebiger Inhalt</td> </tr> <tr> <td>Nummer</td> <td>eine Zahl in einem der unterstützten Formate; ggf. mit Dezimalpunkt (die Feldlänge darf maximal 19 Zeichen betragen)</td> </tr> <tr> <td>Datum</td> <td>im Format 'TT.MM.JJJJ' (Der Feldtyp wird nur geändert, wenn die Feldlänge 10 ist)</td> </tr> <tr> <td>Zeit</td> <td>im Format 'HH:MM:SS' (s. o.; das Feld muss jedoch 8 Zeichen lang sein)</td> </tr> <tr> <td>Extern</td> <td>Dateipfad</td> </tr> <tr> <td>Berechnung</td> <td>= Formel; es gilt dasselbe wie bei Nummern</td> </tr> <tr> <td>Logisch</td> <td>das Feld darf nur ein Zeichen lang sein</td> </tr> <tr> <td>Relation</td> <td>Verknüpfung mit einer anderen Datenbankdatei</td> </tr> </table>	Text	beliebiger Inhalt	Nummer	eine Zahl in einem der unterstützten Formate; ggf. mit Dezimalpunkt (die Feldlänge darf maximal 19 Zeichen betragen)	Datum	im Format 'TT.MM.JJJJ' (Der Feldtyp wird nur geändert, wenn die Feldlänge 10 ist)	Zeit	im Format 'HH:MM:SS' (s. o.; das Feld muss jedoch 8 Zeichen lang sein)	Extern	Dateipfad	Berechnung	= Formel; es gilt dasselbe wie bei Nummern	Logisch	das Feld darf nur ein Zeichen lang sein	Relation	Verknüpfung mit einer anderen Datenbankdatei
Text	beliebiger Inhalt																
Nummer	eine Zahl in einem der unterstützten Formate; ggf. mit Dezimalpunkt (die Feldlänge darf maximal 19 Zeichen betragen)																
Datum	im Format 'TT.MM.JJJJ' (Der Feldtyp wird nur geändert, wenn die Feldlänge 10 ist)																
Zeit	im Format 'HH:MM:SS' (s. o.; das Feld muss jedoch 8 Zeichen lang sein)																
Extern	Dateipfad																
Berechnung	= Formel; es gilt dasselbe wie bei Nummern																
Logisch	das Feld darf nur ein Zeichen lang sein																
Relation	Verknüpfung mit einer anderen Datenbankdatei																
Feldlänge	Eine Änderung der Feldlänge ist zur Zeit noch nicht möglich; sie wird nur angezeigt.																
Vorgabewert	Sie können neue Vorgabewerte definieren.																
Optionen	Die Eingabe von Feldoptionen ist möglich.																

Bei der Beschreibung des Befehls "Projekt/Neu" können Sie mehr über die Definition von Vorgabewerten und Optionen erfahren.

Zusätzlich besteht die Möglichkeit, die Einstellungen für Viewer, Zahlenformate sowie Berechnungsformeln zu ändern.

Falls Sie Feldtypen geändert haben und die Datenbank schon Datensätze enthält, erscheint eine Sicherheitsabfrage, ob die Datenbank wirklich konvertiert werden soll.

Eine Konvertierung hat auf die einzelnen Feldtypen folgende Auswirkungen:

Text	(Keine!)
Nummer	Ab dem ersten nicht numerischen Zeichen wird der Rest des Feldinhaltes ignoriert, überflüssige Leerzeichen werden entfernt.
Datum	Die erforderlichen Punkte ('.') werden eingefügt.
Zeit	Wie oben, es werden jedoch Doppelpunkte eingefügt.
Extern	(Keine!)
Berechnung	Neuberechnung aller Formeln.
Logisch	(Keine!)
Relationen	Konvertierung nicht möglich.

Es wird nur beim Feldtyp »Nummer« eine Überprüfung des Feldinhaltes vorgenommen. Wertefehler in Feldern der Typen "Datum" und "Zeit" wie z. B. »31.02.1996« werden nicht berichtigt.

Alle Formeln werden neu berechnet. Da der Feldtyp der Relationsfelder nicht geändert werden kann, werden sie bei der Strukturänderung ignoriert.

Zum Abschluss werden alle Änderungen an den Dateien wieder auf den Datenträger geschrieben.

## 1.13 Tastaturkürzel (Shortcuts)

Für die Menüs gelten in Verbindung mit der rechten Amiga-Taste folgende Tastaturkürzel (sog. "Shortcuts"):

Datei	
Neu ...	N
öffnen ...	O
schließen	C
löschen ...	
~~~~~	
Info	I
~~~~~	
Ende	Q

```

Bearbeiten
Eingeben ... D
Ansehen A
Löschen ...
~~~~~
Sortieren ... S
~~~~~
Suchen ... ?
Aktualisieren ... !
Öffnen ...
Sichern ...
Zurücksetzen ...
~~~~~
Selektion >>
        aktiv 1
        inaktiv 0
        ^
Alles selekt.
Selektionsmatrix >>
        öffnen ...
        sichern ...
        löschen

Ausgabe
Import ... <
Export ... >
Extrakt ... ^
~~~~~
Listen >>
        öffnen ... ,
        sichern ...
        edieren (
        drucken L

Etiketten >>
        öffnen ... .
        sichern ...
        edieren )
        drucken E

Spezial
Neues CLI \
~~~~~
Statistiken >>
        Gesamtsumme +
        Durchschnitt -
        Maximum G
        Minimum K
~~~~~
Hilfe ... H

Extras
dBASE-Datenbank >>
        Anzeigen ...
        Importieren ...
        Exportieren ...

Systeminformation
Reorganisieren ...
Caches aktualisieren

```

Diagramm erstellen ...

Optionen

festlegen ...

öffnen ...

sichern

sichern als ...

~~~~~

Struktur ändern ...

## 1.14 Interna: Programmtechnische Informationen

Folgende Themen sind verfügbar:

Einschränkungen dieser MicroBase-Version

Dateinamen-Erweiterungen (Extensionen)

Interner Aufbau der verwendeten Dateiformate

## 1.15 Interner Aufbau der verwendeten Dateiformate

Folgende Abkürzungen werden verwendet:

|       |                              |
|-------|------------------------------|
| NUM   | numerischer Wert in Textform |
| TXT   | Text                         |
| "TXT" | Text in Anführungszeichen    |
| <CR>  | Zeilenvorschub               |

MicroBase-Definitionsdatei (.DEF)

NUM Feldanzahl, NUM Satzlänge <CR>

"TXT" Feldname <CR> -----

NUM Feldlänge <CR> | für jedes Datenfeld zu wiederholen

"TXT" Feldtyp <CR> -----

MicroBase-Zeigerdatei (.PTR)

NUM Maximalanzahl <CR>

NUM Datensatzanzahl <CR>

MicroBase-Datenbankdatei (.DAT)

MicroBase speichert alle Datensätze mit derselben Satzlänge ab.

Um dies zu erreichen, wird jedes Datenfeld rechts so lange mit Leerzeichen aufgefüllt, bis es die definierte Feldlänge hat. Danach werden alle Datenfelder in der definierten Reihenfolge gesichert.

## Beispiel

-----

Wenn eine Datenbank zwei Felder (20 Zeichen / Text, 10 Zeichen / Nummer) hat und der erste Datensatz die Daten "Test" und "1,234" enthält, wird der erste Datensatz wie folgt gespeichert:

```
0000000000111111111122222222223 Position
123456789012345678901234567890
Test           1.234           Inhalt
```

Dezimalkommata werden als Dezimalpunkt gespeichert, doppelte Anführungszeichen als einfache.

## Virtuelle Selektionsmatrix (.VM2)

Die Selektionsmatrix für den 'virtuellen Modus' enthält für jeden Datensatz genau ein Zeichen - '1' für selektierte bzw. '0' für nicht selektierte Datensätze.

MicroBase löscht die .VM2-Datei während des Schließens der Datenbankdatei.

## Sperrdatei (.LCK)

Die Sperrdatei für den Mehrbenutzerbetrieb enthält entweder die Nummer des zu sperrenden Datensatzes oder '-1', wenn die gesamte Datenbankdatei gesperrt werden soll.

## 1.16 Dateinamen-Erweiterungen (Extensionen)

Die Tabelle listet die verwendeten Erweiterungen auf. Es wird jeweils angegeben, ob es sich um eine ASCII-Datei handelt - die von jedem Editor gelesen werden kann - oder nicht.

"Feldabhängig" bedeutet, dass die Datei zu jedem Datenfeld eine Zeile enthält, die jedoch auch leer sein kann. Diese Dateien werden zur Definition einer Datenbank benötigt.

In der Spalte "Bemerkungen" finden Sie zusätzliche Informationen.

| Erw. | ASCII | Feldabh. | Bemerkungen                                                                                                                                  |
|------|-------|----------|----------------------------------------------------------------------------------------------------------------------------------------------|
| .APR | Binär | Nein     | Datei, die von Lotus Approach zusätzlich zu den dBASE-Dateien erzeugt wird. Kann von MicroBase leider nicht interpretiert werden.            |
| .ASC | ASCII | Nein     | dBASE-Datenbank, die mit DBF2ASC2 in eine sequentielle Datei konvertiert wurde. Die erste Zeile kann zusätzlich Feldbezeichnungen enthalten. |
| .CCY | ASCII | Ja       | Diese Datei enthält die definierten Zahlenformate.                                                                                           |
| .CLC | ASCII | Ja       | Diese Datei enthält die Berechnungs-Formeln.                                                                                                 |
| .CMD | ASCII | Ja       | Diese Datei enthält die Pfade zu den Viewern.                                                                                                |



|      |       |        |                                                                                                                                                                                                         |
|------|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| .CVD | ASCII | Nein   | Definition der #Konstanten.                                                                                                                                                                             |
| .DAT | Binär | Nein   | Dies ist die Datenbankdatei (ggf. verschlüsselt). MicroBase erstellt ggf. beim Öffnen der Datenbank eine neue Datendatei, falls die bei der Datenbankdefinition erstellte versehentlich gelöscht wurde. |
| .DEF | ASCII | (Nein) | Dies ist die Definitionsdatei. Sie enthält die Feldnamen und -längen. Ohne Diese Datei kann die Datenbank nicht korrekt geöffnet werden.                                                                |
| .LCK | ASCII | Nein   | Sperrdatei für den Netzwerkbetrieb.                                                                                                                                                                     |
| .MBA | ASCII | Nein   | Diese Datei enthält eine Abfragedefinition.                                                                                                                                                             |
| .MBO | ANSI  | Nein   | Gedruckte Liste bzw. gedrucktes Etikett.                                                                                                                                                                |
| .MBP | Binär | Nein   | Enthält die festgelegten Optionen.                                                                                                                                                                      |
| .NUM | ASCII | Nein   | Aktueller Wert von Seriennummer und Schrittweite.                                                                                                                                                       |
| .OPT | ASCII | Ja     | Feldoptionen wie z. B. Wertebereich.                                                                                                                                                                    |
| .PRE | ASCII | Ja     | Vorgabewerte für die Datenfelder.                                                                                                                                                                       |
| .PTR | ASCII | Nein   | Diese Datei enthält die maximale und die tatsächliche Anzahl von Datensätzen. Sie wird ggf. beim Öffnen der Datenbankdatei neu erstellt, wenn sie versehentlich gelöscht wurde.                         |
| .REL | ASCII | Ja     | Definition der Relationen.                                                                                                                                                                              |
| .SAL | ASCII | Nein   | Diese Datei enthält eine Listendefinition.                                                                                                                                                              |
| .SAE | ASCII | Nein   | Diese Datei enthält ein Etikettdefinition.                                                                                                                                                              |
| .SMX | ASCII | Nein   | Diese Datei enthält eine Selektionsmatrix, d. h. die Nummern der selektierten Sätze.                                                                                                                    |
| .SYS | Binär | Nein   | Dies ist eine Systemdatei, die Informationen über die verwendete MicroBase-Version enthält. Sie wird ggf. bei der nächsten Dateneingabe neu erstellt, falls sie versehentlich gelöscht wird.            |
| .VM  | Binär | Nein   | Virtueller Speicher (nur während des Sortierens.)                                                                                                                                                       |
| .VM1 | Binär | Nein   | Früher Hilfsdatei für den 'virtuellen Modus'; wird nicht mehr benötigt und kann gelöscht werden.                                                                                                        |
| .VM2 | Binär | Nein   | Selektionsmatrix im 'virtuellen Modus'.                                                                                                                                                                 |
| .nn  | Binär | Nein   | Cache-Dateien für Relationsfelder (nn = Feldnr.)                                                                                                                                                        |

Anmerkung: Die meisten feldabhängigen Dateien können repariert werden, indem die Datenbankstruktur bearbeitet wird. Bei diesem Vorgang werden die Dateien automatisch aktualisiert.

|      |       |      |                                                    |
|------|-------|------|----------------------------------------------------|
| .DBF | Binär | Nein | dBASE- bzw. FoxPro-Datenbankdatei (auch Approach.) |
| .DBT | Binär | Nein | dBASE-Memotexte-Datei (auch Approach.)             |
| .FPT | Binär | Nein | FoxPro-Memotexte-Datei.                            |

Anmerkung: Die Memotexte-Dateien können unter Windows auch Grafiken oder OLE-Objekte enthalten.

## 1.17 Einschränkungen dieser MicroBase-Version

Die vorliegende Version von MicroBase ist folgenden Einschränkungen unterworfen:

Alle Modi  
-----

Max. Länge für Feldnamen:                    20 Zeichen  
Max. Feldlänge:                                2 048 Zeichen

|                                 |                                                                      |
|---------------------------------|----------------------------------------------------------------------|
| Anzahl Datentypen:              | 8 (Text, Datum, Nummer, Zeit, Extern, Berechnung, Logisch, Relation) |
| Max. Anzahl von Feldern:        | 234 Felder                                                           |
| Max. Datensatzlänge:            | 65 536 Zeichen                                                       |
| Max. Formellänge:               | 2 048 Zeichen                                                        |
| Max. Viewer-Pfadlänge:          | 2 048 Zeichen                                                        |
| Max. Länge Standardvorgaben:    | 20 Zeichen                                                           |
| Max. Länge Feldoptionen:        | 12 Zeichen                                                           |
| Max. Anzahl Verknüpfungen:      | 25 Datenbankdateien                                                  |
| Max. Anzahl geöffneter Dateien: | 26 Datenbankdateien                                                  |

#### Standard-Modus

-----

|                              |                                |
|------------------------------|--------------------------------|
| Max. Anzahl von Datensätzen: | 65 535 Sätze                   |
| Max. Dateilänge:             | 4 294 901 760 Zeichen (~ 4 GB) |

#### Virtueller Modus

-----

|                              |                                      |
|------------------------------|--------------------------------------|
| Max. Anzahl von Datensätzen: | 2 147 483 547 Sätze (~ 2 Milliarden) |
| Max. Dateilänge:             | 4 294 967 295 Zeichen (4 GB)         |

#### Demo-Version

-----

|                              |                              |
|------------------------------|------------------------------|
| Max. Anzahl von Datensätzen: | 20 Sätze                     |
| Max. Dateilänge:             | 1 310 720 Zeichen (~ 1,3 MB) |

Die Länge eines Datensatzes ist programmintern auf maximal 64 KB begrenzt.

## 1.18 Die MicroBase-Fehlermeldungen

|                                  |                                                                                                                                                                                                                                                                                                 |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XXX Bytes Speicher erforderlich. | Mit dem Tooltype MEMORY (siehe Abschnitt Installation) wurde Speicher angefordert, der jedoch auf Ihrem Rechner nicht verfügbar ist. Hierbei ist zu beachten, dass MicroBase 70 KB Speicher für das Betriebssystem frei lassen möchte und bei MEMORY=400 also 470 KB Speicher frei sein müssen. |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Lösung: Verringern Sie den Wert, den Sie bei MEMORY= angegeben haben. (Kaufen Sie sich eine Speichererweiterung, falls Sie nur 512 KB RAM haben.)

Im 'virtuellen Modus' müssen immer 180 KB frei sein.

|                                                                         |                                                                                                                                                                                                     |
|-------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Die Ressourcen werden knapp.<br>Bitte Programm beenden und neu starten. | Diese Warnung wird nach der Ausführung eines Menübefehls angezeigt, wenn weniger als 20 Prozent des reservierten Speichers frei sind. (Dies ist kein Fehler im eigentlichen Sinne, sondern nur eine |
|-------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Warnung.) Normalerweise sollte dieser Wert bei über 70 Prozent liegen.

Lösung: Versuchen Sie, Speicher frei zu machen. Sie können z. B. leere Listen und Suchmasken erstellen.

Wenn diese Maßnahmen keinen Erfolg zeigen, müssen Sie dem angezeigten Ratschlag folgen und das Programm verlassen. Beim nächsten Programmstart sollten Sie mehr Speicher reservieren oder den virtuellen Modus aktivieren.

Benötige Kickstart 2.0 oder höher. Sie versuchen, MicroBase unter Kickstart 1.3 oder niedriger laufen zu lassen.

Lösung: Kaufen Sie sich Kickstart 2.0 oder besser 3.1.

RexxMaster-Prozess läuft nicht. Berechnungen und Skripte nicht ausführbar. Dieser Hinweis erscheint beim Programmstart, falls der ARexx-Interpreter "RexxMast" noch nicht läuft.

Lösung: Starten Sie "RexxMast", am besten in der Datei "S:User-Startup". Sie können auch das Icon in die "WBStartup"-Schublade ziehen ("RexxMast" befindet sich im "System"-Verzeichnis des Startlaufwerkes).

Falls ARexx noch nicht installiert ist, sollten Sie dies nachholen. Sie können sonst keine Berechnungen oder Skripte ausführen.

»Dateiname«  
Datei ist schon  
vorhanden!

Sie möchten eine Datenbankdatei erstellen. Diese existiert jedoch schon.

Lösung: Verwenden Sie einen anderen Namen für Ihre Datenbankdatei.

»Dateiname«  
Demo-Limit überschritten

Sie möchten eine Datenbank mit mehr als 20 Datensätzen öffnen, besitzen jedoch kein Keyfile.

Lösung: Lassen Sie sich registrieren!

(Manchmal lässt sich eine Datei erst beim zweiten Versuch öffnen, nachdem MicroBase in das entsprechende Verzeichnis gewechselt ist.)

Falsches Passwort.  
Zugriff verweigert!

Sie haben beim Öffnen einer Datenbank das falsche Passwort eingegeben.

Lösung: Geben Sie beim Öffnen der Datenbankdatei ein anderes Passwort an.

Anmerkung: Bei Passwörtern wird die Groß- bzw. Kleinschreibung unterschieden. Leerzeichen sind erlaubt.

»Dateiname«  
Reorganisierung notwendig!

Die gewählte Datenbankdatei enthält fehlerhafte Daten und kann deshalb nicht geöffnet werden. Es sind weniger Datensätze vorhanden, als in der Definitionsdatei der Datenbank angegeben sind.

Lösung: MicroBase reorganisiert die Datei automatisch. Bei diesem Vorgang können jedoch möglicherweise Datensätze verloren gehen.

»Dateiname«  
Datenb./Passw. fehlerh.

Dieser Fehler tritt direkt beim Öffnen einer Datenbankdatei auf, wenn der erste Datensatz nicht korrekt gelesen werden kann.

Lösung: Je nach Ursache sind zwei Vorgehensweisen denkbar.

1. Wenn der erste Datensatz auf dem Bildschirm korrekt aussieht, müssen Sie die Datenbankdatei reorganisieren lassen, damit fehlerhafte Zeichen ("Datenmüll") entfernt wird.
2. Beim Ansehen des ersten Datensatz erscheinen auf dem Bildschirm nur wirre Zeichen. Dies bedeutet, dass die Datei verschlüsselt ist und Sie das Passwort falsch eingegeben haben. Öffnen Sie die Datenbankdatei mit dem korrekten Passwort.

Die Datenbank wird von einem anderen Benutzer bearbeitet. 'Virtueller Modus' ist nicht verfügbar. werden.

Sie arbeiten mit MicroBase in einem Netzwerk und möchten eine Datenbankdatei im 'virtuellen Modus' öffnen. Die Datenbankdatei wird jedoch schon von einem anderen Benutzer bearbeitet; die Verwaltungsdatei für den 'virtuellen Modus' existiert also schon.

Lösung: MicroBase aktiviert automatisch den Standardmodus, damit Sie die Datei öffnen können.

Wenn Sie absolut sicher sind, dass nur Sie auf die Datenbankdatei zugreifen, dann können Sie die zu der Datenbankdatei gehörende Verwaltungsdatei löschen. Diese hat den gleichen Namen wie die Datenbank, jedoch die Dateiendung ".VM2". Danach können

Sie die Datei im 'virtuellen Modus' öffnen.

Referenzintegrität verletzt.  
Bitte Dateneingabe korrigieren.

Der von Ihnen eingegebene Datensatz verletzt die referenzielle Integrität, d. h. in der verbundenen Datenbankdatei konnte kein passender Datensatz gefunden werden. (Dieser Fehler tritt nur auf, wenn Sie die "strikte Referenzintegrität" aktiviert haben.)

Lösung: Sie müssen korrekte Daten eingeben, da der Datensatz sonst nicht gespeichert werden kann.

Ungültige Datensatzzahl: Maximal »Anzahl«

Sie haben mehr Datensätze eingegeben als möglich sind.

- Lösung:
- 1 Erhöhen Sie die Anzahl der möglichen Datensätze.
  - 2 Erhöhen Sie dafür ggf. den zu reservierenden Speicher.
  - 3 Aktivieren Sie notfalls den 'virtuellen Modus', der mehr als 2 Milliarden Datensätze ermöglicht.

Ungültiger Feldinhalt Fehlerbeschreibung

Sie haben etwas eingegeben, das nicht zum Typ des Datenfeldes passt.

Lösung: Die Fehlerbeschreibung gibt Ihnen Hinweise, wie Sie korrekte Eingaben vornehmen. Sie können für weitere Hinweise auf <Help> drücken.

Der Fehler hängt vom Feldtyp sowie den Feldoptionen ab.

»Dateiname«  
Relation nicht möglich

Folgende Ursachen sind möglich: falsche Datei, falsche Schlüssel- oder Datenfelder, falscher Schlüsselbegriff. Es wird der Name der zweiten (verknüpften) Datenbank angegeben. (Bei Verwendung von Nebenschlüsseln kann möglicherweise ein Teil der Relationen trotzdem angezeigt werden.)

Diese Fehlermeldung weist darauf hin, dass die referenzielle Integrität der Datenbank verletzt worden ist.

Lösung: Ändern Sie die Relationsdefinition oder den Inhalt des Schlüsselfeldes.

»Dateiname«

Sie wollen eine Datei importieren, die mehr Felder

---

- Zu viele Felder. als die aktuelle Datei hat. Dies ist in MicroBase nicht möglich.
- Lösung: Erstellen Sie eine neue Datei mit der korrekten Anzahl von Feldern oder importieren Sie die Datei in eine neue Datenbankdatei.
- »Dateiname«  
Mehr als XXX Felder. Sie wollten eine Datei importieren, die mehr als die in MicroBase maximal erlaubten Felder hat.
- Lösung: Teilen Sie die Datei in mehrere Dateien auf, um die Anzahl der Felder zu verringern.
- »Dateiname«  
Datei ist leer! Sie wollen eine Datei importieren oder öffnen, die leer ist.
- Lösung: Öffnen Sie eine Datei, die nicht leer ist.
- Suchbegriff wurde nicht gefunden. Sie haben im Ansichtsbildschirm die Begriffssuche aktiviert und die Suche blieb erfolglos.
- Lösung: Es gibt – je nach Ursache – zwei Lösungsmöglichkeiten.
- 1 Beachten Sie, dass immer ab dem nächsten Datensatz gesucht wird. (Sie können sich so ganz einfach per Tastendruck durch die gesamte Datenbank bewegen.)
  - 2 Es wird nur in selektierten Datensätzen gesucht.
- Hinweis: Falls Sie den genauen Suchbegriff nicht kennen, können Sie die phonetische Suche aktivieren.
- »Satznummer«  
Satznummer zu hoch! Sie wollten im Ansichtsbildschirm zu einem Datensatz springen, der nicht existiert.
- Lösung: Geben Sie die Nummer eines existierenden Datensatzes ein. In der Titelleiste zeigt MicroBase die Anzahl der vorhandenen Datensätze an.
- »Dateiname«  
Falsches Dateiformat! Sie wollen eine Datei laden, die nicht von MicroBase erstellt wurde.
- Lösung: Laden Sie eine MicroBase-Datei (die erforderliche Extension für Datenbankdateien ist z. B. '.DAT', die für Etiketten '.SAE').
-

- »Name der temp. Datei«  
Fehler im virtuellen  
Speicher.
- Das Verzeichnis, das Ihre Datenbank enthält, verfügt nicht über genügend freien Speicher für die zum Sortieren benötigte Auslagerungsdatei.
- Lösung: Kopieren Sie alle zu der Datenbank gehörenden Dateien auf ein anderes Laufwerk (nicht auf die RAM-Disk; RAD: kann jedoch benutzt werden).
- Datei nicht gefunden.
- Sie haben bei einem externen Feld den Pfad einer nicht existierenden Datei angegeben.
- Lösung: Geben Sie eine andere Datei an. MicroBase änderte den Feldinhalt automatisch in '\$' ab, der einen ASL-Requester zur einfacheren Dateiauswahl öffnet.
- Datenbank / -satz wird  
von einem anderen  
Benutzer bearbeitet.  
Gewünschte Operation  
kann nicht ausgeführt  
werden.
- Sie arbeiten mit MicroBase in einem Netzwerk und möchten eine Datenbankdatei bearbeiten. Die Datei wird jedoch schon von einem anderen Benutzer bearbeitet.
- Lösung: Sie müssen warten, bis die Datenbank bzw. der Datensatz wieder verfügbar ist.
- Wenn Sie absolut sicher sind, dass nur Sie auf die Datenbankdatei zugreifen, dann können Sie die zu der Datenbankdatei gehörende Sperrdatei löschen. Diese hat den gleichen Namen wie die Datenbank, jedoch die Dateiendung ".LCK".
- Fehler im ARexx-Progr.  
Skript / Berechnung  
schlag fehl.
- Sie haben ein Skript oder eine Berechnung ausgeführt. Diese konnte von ARexx nicht bearbeitet werden.
- Lösung: Sie sollten zunächst überprüfen, ob die erforderlichen Bibliotheken vorhanden sind. MicroBase benötigt die "rexsupport.library" und die "rexxmathlib.library". Möglicherweise haben Sie auch ARexx nicht korrekt installiert.
- Eine weitere Fehlerquelle sind Syntaxfehler in Ihrem Skript. Sie sollten versuchen, das Skript einmal außerhalb von MicroBase zu starten. Möglicherweise hat sich bei den Variablennamen (field001 bis field234) ein Fehler eingeschlichen.
- Die letzte Fehlerquelle könnte Speichermangel sein. Skript und Ausgabedatei können während der Laufzeit doppelt so groß wie

ein Datensatz werden (zusammen maximal 150 KB). Eine Berechnung benötigt halb so viel Speicher, und zwar im aktuellen Verzeichnis.

## 1.19 Copyright-Bestimmungen und Registrierung

Copyright (C) 1992-1997 by FR-SoftWorks.

\*\*\* »rexxmathlib.library« © 1989 by Willy Langeveld

\*\*\* »hbasic2.library« © 1994 by MAXON Computer GmbH, HiSoft

Für eventuelle Schäden, die durch das Programm entstehen, wird nicht gehaftet.

Das Programm darf ohne Erlaubnis des Autors weder weitergegeben bzw. verkauft noch übersetzt, disassembliert oder in eine andere Programmiersprache übersetzt werden.

Verbesserungsvorschläge, Fragen und Fehlerberichte senden Sie bitte an:

Frank Reibold  
Ottberger Weg 13  
31737 Rinteln

eMail: Peter.Reibold@T-Online.de

Bei Fehlerberichten notieren Sie bitte die genaue Fehlermeldung und schildern auch die Situation, die den Fehler verursachte. Geben Sie auch Ihre Hardware-Konfiguration (benutzter Computer, Betriebssystemversion usw. ) an.

### Registrierung

MicroBase ist ein Shareware-Programm und muss deshalb registriert werden (falls Sie es nicht als Bestandteil von MicroWrite erworben haben.) Eine Registrierung ist erforderlich, wenn Sie MicroBase nach Ablauf der Testphase (30 Tage) weiter benutzen möchten.

Um sich registrieren zu lassen, drucken Sie bitte die Liste »Registrierung.SAL«, füllen sie aus und senden sie an den Autor.

Nach der Registrierung erhalten Sie ein Keyfile, das auch alle folgenden Versionen freischaltet.

### DBF2ASC und DBVIEW

Die Programme DBF2ASC und DBVIEW sind Public Domain und dürfen frei kopiert werden. Die Quellcodes in MaxonBasic 3 sind beim Autor erhältlich und Bestandteil des MicroWrite-Paketes.

---



Viel Spaß mit MicroBase!

## 1.20 Weitere Produkte von FR-SoftWorks

Von FR-SoftWorks sind folgende Produkte erhältlich:

MicroWrite      MicroWrite ist ein Textverarbeitungssystem mit folgenden Merkmalen:

- integrierte Adressverwaltung; Erstellung von Serienbriefen
  - integrierter Datei-Manager: Dateien löschen, umbenennen, kopieren und anzeigen (ab Kickstart 2.0)
  - Import und Export von ASCII-Texten; Export von ANSI-Dokumenten zum Ausdrucken über Preferences
  - Import und Export von Adressen in folgenden Formaten (alle nur "ASCII"): SuperBase Professional 4, TransWrite, Documentum 2, BECKERtext III, ProWrite, Personal Write. Anpassung an andere Formate ist möglich
  - Verschlüsseln von Texten
  - Unterstützung der Druckerzeichensätze; schneller Druck; Anpassung der Umlaute möglich; Druck in Dateien möglich
  - Stichwortlisten, Seitennummerierung, Textstatistik und -analyse
  - Druckbildvorschau
  - Umfangreiche Funktion zum Sortieren von Texten; z. B. nach dem dritten Wort
  - Addieren und Multiplizieren von Textzeilen und -spalten
  - Berechnung von Gleichungen (über ARexx)
  - Lösung einer Gleichung (über ARexx)
  - Installationsprogramm für die Festplatte
  - externe Rechtschreibprüfung (z. Zt. nur eingeschränkte Testversion)
  - Hilfe-Funktion und Lernprogramm
  - Häufigkeit von Begriffen ermitteln
  - Übersetzungslisten
  - Erstellung von Autorenvermerken, die z. B. Thema, Datum und Autor enthalten
-

- Texte mit Hilfe von Autorenvermerken suchen
- und vieles mehr.

MicroWrite benötigt mindestens Kickstart 2.0

Für die Berechnung von Gleichungen wird ARexx benötigt (liegt Kickstart 2.0 oder höher bei).

Das Programm druckt direkt über die parallele Schnittstelle; die Einstellungen in den Preferences werden übergangen. Hierdurch wird eine bessere Kontrolle über den Drucker möglich.

MicroWrite kommt auf 1 Diskette und kostet z. Zt. 10,00 DM.

MicroBase ist die Dateiverwaltung, die in diesem Dokument beschrieben wird.

MicroPaint ist ein Malprogramm, das das AA Chipset unterstützt.

- bis zu 256 Farben auf Rechnern mit AA Chipset
  - Unterstützung von ECS und AA Chipset
  - Unterstützung von Intellifonts, Schriftstilen, Drawmodes (z. B. invers)
  - Unterstützung aller Grafikmodi, die Ihr Computer unterstützt; z. B. SuperHighres Interlace
  - Flexible Spray-Funktion
  - Multi-Spray: mit Zufallsfarben sprühen
  - Helligkeits-Effekt: Ändern Sie ganz einfach die Helligkeit des ganzen Bildes oder nur einiger Farben bzw. Farbkomponenten. (Sie können z. B. den Rotanteil der Zeichnung um 50 % erhöhen.)
  - Grafiken invertieren und umfärben
  - Grafiken in Graustufen umrechnen
  - Grafiken drucken
  - Pinsel
  - Bildverarbeitung
  - Screen-Grabber
  - Datatypes
  - Sichern im BMP-Format
  - definierbare Zeichenabstände
-

- Linienstile
- Tutorial
- lokalisiert
- Werkzeugleiste
- Animationen können abgespielt werden
- ARexx-Schnittstelle

MicroPaint benötigt mindestens Kickstart 2.0. Es kommt auf einer Diskette und kostet z. Zt. 25,00 DM.

## 1.21 Steuerzeichen in Listen und Etiketten

Textformatierung

|     |                              |
|-----|------------------------------|
| \$f | Fettschrift ein              |
| \$u | Unterstreichen ein           |
| \$i | Kursivschrift ein            |
| \$n | Alles aus; Normalschrift ein |
| \$h | Schrift hochsetzen           |
| \$t | Schrift tiefsetzen           |
| \$m | Hoch-/Tiefsetzen ausschalten |
| \$# | Gibt Satznummer aus.         |
| \$s | Seitenvorschub               |

Ausgabe von Feldinhalten

|      |                                                                                                       |
|------|-------------------------------------------------------------------------------------------------------|
| &nnn | Gibt das Feld Nummer nnn in seiner ganzen Länge aus, indem es mit rechts Leerzeichen aufgefüllt wird. |
| %nnn | Gibt nur das Feld Nummer nnn aus.                                                                     |
| #nnn | Gibt den Namen des Feldes Nummer nnn aus.                                                             |

nnn ist eine dreistellige Zahl zwischen 001 und 234.

Listengenerator "makelist"

Der Listengenerator erzeugt aus Datenbankdefinitionsdateien (.DEF) automatisch eine Standardliste in Masken- oder Tabellenform.

Als zweites Argument kann optional das Wort 'table' angegeben werden, um eine Liste in Tabellenform zu erhalten.

Sie finden den Listengenerator im Verzeichnis "Utils". Das Utility muss aus dem CLI gestartet werden.

Hinweis

Listen und Etiketten der Version 1.X/2.X müssen angepasst werden, da sich die

Steuerzeichen für die Feldinhalte geändert haben.

Hierzu finden Sie im Verzeichnis "Utils" das Utility "ConvertList", das Listen und Etiketten in das neue Format konvertiert. Das Utility läuft nur in der Shell.

## 1.22 Suchkriterien

Der Aufbau des Suchkriteriums ist '\$\$KOO<Zeichenfolge>'. \$\$ bedeutet, dass ein erweitertes Suchkriterium folgt. Normalerweise reicht jedoch die Angabe einer Zeichenfolge aus.

K ist ein Kennzeichen, ob nach Zeichen oder Zahlen gesucht werden soll. ('\$' steht für Zeichen, '%' für Zahlen.) OO steht für den sogenannten Operator.

Wird nur einfach ein Suchbegriff angegeben, so wird nicht nach Groß- und Kleinschreibung unterschieden, jedoch in allen anderen Fällen. Wollen Sie doch eine Unterscheidung nach Groß/Kleinschreibung, so ändern Sie bitte die Einstellungen im Menü "Optionen/Festlegen..."

Beispiele

|               |                                                                   |
|---------------|-------------------------------------------------------------------|
| Meier         | sucht nach 'MeIeR', 'Meier-Lehmann' usw.                          |
| \$\$\$==Meier | sucht exakt nach 'Meier'<br>(leere Suchbegriffe sind immer wahr!) |
| \$\$\$==      | sucht leeres Datenfeld!                                           |
| \$\$%<>200    | sucht Zahlen, aber nicht 200                                      |
| \$\$%<<10000  | sucht Zahlen die kleiner sind als 10 000                          |
| \$\$%>>6100   | sucht Zahlen, die größer sind als 6 100                           |
| \$\$%<=0      | sucht Zahlen, die kleiner oder gleich 0 sind                      |
| \$\$%>=0      | sucht Zahlen, die größer/gleich Null sind                         |

Alle diese 'Operatoren' gelten sowohl für Zahlen als auch für Zeichenfolgen. (Zeichenfolgen ergeben den Wert 0, falls sie keine gültige Zahl darstellen.)

Phonetische Suche

Wenn Sie die phonetische Suche aktiviert haben, sucht MicroBase die Daten über ihre jeweiligen Lautwerte. Dies bedeutet, dass Sie z. B. auch die Namen "Mayr" und "Maier" finden, wenn Sie nach "Meier" suchen.

## 1.23 Operatoren für das Aktualisieren

Beim Aktualisieren kann angegeben werden, was mit den gefundenen Datenfeldern geschehen soll. Dies können Zeichenfolgen oder 'Aktualisierungsbefehle' sein. Aktualisierungsbefehle haben dasselbe Format wie Suchkriterien, aber andere Operatoren.

Beispiele

```

$$%++<Wert>      addiert <Wert> zum Wert des Datenfeldes
$$%--<Wert>      subtrahiert <Wert> vom Wert des Feldes
$$%//<Wert>      dividiert Wert des Feldes durch <Wert>
$$%**<Wert>      multipliziert Wert des Feldes mit <Wert>
2000              neuer Inhalt des Feldes ist: '2000'
hallo             neuer Inhalt des Feldes ist 'hallo'

```

### Zeichenfolgen ersetzen

Es gibt keine speziellen Aktualisierungsbefehle für Zeichenfolgen. Falls die Option "partielles Ersetzen" nicht aktiviert ist, ersetzt MicroBase das komplette Feld. (Das "partielle Ersetzen" wird nur dann durchgeführt, wenn Sie keine Suchkriterien angegeben haben.)

Hinweis: Das "partielle Ersetzen" kann zu Problemen führen, wenn Sie das phonetische Suchen aktiviert haben. In diesem Fall kann es vorkommen, dass ein Begriff zwar über seinen Lautwert gefunden, aber wegen fehlender exakter Übereinstimmung nicht ersetzt werden kann.

## 1.24 MicroBase-Datenbanken in das dBASE-Format konvertieren

Mit dem Programm »DAT2DBF« können Sie Datenbanken aus dem MicroBase- in das dBASE-Format wandeln.

Gehen Sie dazu in folgenden Schritten vor:

- 1 Auswahl einer MicroBase-Datei (.DAT). Die MicroBase-Datei darf nicht verschlüsselt sein, da das Verschlüsselungsverfahren nicht zu dBASE kompatibel ist.
- 2 Das Programm erzeugt einen sogenannten "Header", der die Felddefinitionen enthält, und schreibt ihn in die dBASE-Datei (.DBF). Dazu werden die Feldnamen auf 10 Buchstaben begrenzt und in Großbuchstaben gewandelt. Die Feldlängen werden nicht begrenzt; alle Felder werden als Textfelder behandelt.
- 3 Sie können wählen, ob der Zeichensatz in das ASCII-Format gewandelt werden soll. Dies ist notwendig, falls Sie noch dBASE für DOS benutzen, da MS-DOS einen anderen Zeichensatz als Amiga bzw. Windows benutzt. Auch einige Windows-Datenbanksysteme verwenden aus Kompatibilitätsgründen diesen Zeichensatz.
- 4 Die Datensätze werden kopiert bzw. konvertiert.

Hinweise: Die "hbasic2.library" wird im Verzeichnis LIBS: benötigt.

Die erstellte Datenbankdatei hat das Format der dBASE-Version III+ für DOS (die Feldadressen werden auf 0 (null) gesetzt, da sie während der Bearbeitung einer dBASE-Datenbank auf die korrekten Werte gesetzt werden müssen).

Das korrekte Konvertieren kann mit dem

dBASE-Viewer  
überprüft werden.

Nach dem Konvertieren sollten Sie die Datenbankstruktur überprüfen und ggf. die Feldtypen anpassen.

Hinweise zum dBASE-Datenbankformat

Bei dBASE-Datenbanken sind folgende Besonderheiten zu beachten:

- numerische Felder haben wie in MicroBase bis zu 19 Stellen (die Anzahl der Nachkommastellen muss zusätzlich angegeben werden)
- Textfelder können lediglich bis zu 254 Zeichen lang sein
- Felder des Typs Datum werden ohne Punkte gespeichert; Zeitfelder werden z. T. mit Hundertstelsekunden gespeichert bzw. von einigen Programmen nicht akzeptiert
- Relationen, externe und Berechnungsfelder können nicht konvertiert werden
- die Datensatzgröße ist auf 4 KB beschränkt
- es sind max. 1 Mrd. Datensätze bei einer Dateigröße von 2 GB möglich.

Die Angaben gelten für Datenbanken des Formats dBASE IV, die mit Lotus Approach 3.0 für Windows bearbeitet werden. Bei anderen Programmen können die Einschränkungen größer sein.

## 1.25 SuperBase-Datenbanken konvertieren

Mit dem Programm »sbf2dat« können SuperBase-Datenbankdateien in das Format von MicroBase konvertiert werden.

Gehen Sie wie folgt vor:

- 1 Auswahl einer SuperBase-Datenbankdatei (.sbf).
- 2 Die Datenfelder werden konvertiert. Feldnamen und -längen werden übernommen. Bei anderen Feldtypen als "TXT" erscheint eine Warnung, da nur Textfelder korrekt konvertiert werden können.
- 3 Die Datensätze werden konvertiert. Dabei wird die Anzahl der bereits konvertierten Datensätze angezeigt.

Die erstellte Datenbankdatei ist unverschlüsselt.

Nach dem Konvertieren sollte die Datenbankstruktur überprüft und ggf. angepasst werden.

Hinweise

Das Konvertieren von SuperBase-Datenbankdateien wurde an Hand von SuperBase

---

Professional IV, Version 1.01, getestet.

Zur Zeit können nur Textfelder (SuperBase-Feldtyp "TXT") korrekt konvertiert werden. Vor der Konvertierung sollten deshalb alle Datenfelder in Textfelder umgewandelt werden.

Umlaute werden von SuperBase unter Umständen anders kodiert als von MicroBase.

## 1.26 dBASE- und FoxPro-Datenbanken konvertieren

Mit dem Programm »dbf2DAT« können Datenbanken folgender Systeme in das MicroBase-Format konvertiert werden:

- Ashton Tate dBASE III, III+, IV 1.x
- Borland dBASE IV 2.0, 5.0 und dBASE für Windows 5.0
- Lotus Approach 3.X und 96
- Microsoft FoxPro 2.X
- SoftMaker DataMaker

Visual dBASE 5.5 und Visual FoxPro 3.0 wurden noch nicht getestet.

Gehen Sie in folgenden Schritten vor:

- 1 Auswahl der Datei (Endung ".DBF")
- 2 Die Datenfelder werden konvertiert. Dabei bleiben die Feldnamen, -längen und -typen erhalten. Falls einige Felder nicht korrekt übernommen werden können, so wird dies angezeigt (z. B. Bilder, OLE, Memos; Feld länger als 2 048 Zeichen; mehr als 234 Felder.) Die Datensatzlänge vermindert sich mindestens um den Wert 1, da dBASE zu jedem Datensatz zusätzlich eine Löschmarkierung speichert. Diese wird von MicroBase nicht benötigt und deshalb ignoriert.
- 3 Sie können wählen, ob der Zeichensatz von IBM-ASCII nach ANSI gewandelt werden soll. Dies ist bei MS-DOS-Dateien der Fall, bei Windows-Dateien meist nicht (einige Windows-Datenbanken verwenden jedoch aus Kompatibilitätsgründen den ASCII-Zeichensatz.)
- 4 Die Datensätze werden konvertiert.
- 5 Zum Abschluss wird die Anzahl der gelesenen sowie der überlesenen (= von dBASE als gelöscht markierten) Datensätze angezeigt.

Die erstellte Datenbank ist unverschlüsselt und hat keine Vorgabewerte.

Das korrekte Konvertieren kann mit dem  
dBASE-Viewer  
überprüft werden.

Sie sollten nach der Konvertierung die Datenbankstruktur überprüfen. Es ist

---

ratsam, die Feldnamen zu ändern, da dBASE sie in Großbuchstaben speichert und auf 10 Zeichen begrenzt. Hierbei wird außerdem überprüft, ob die importierten Daten dem Format von MicroBase entsprechen.

Besonderheiten von Lotus Approach

- Zeitinformationen werden zusätzlich mit Hundertstelsekunden, jedoch ohne Doppelpunkte gespeichert. Sie werden in Felder des Typs "Zeichen" umgewandelt.
- Zusätzlich zum Standard vorhandene Feldtypen (Variable und Berechnung) werden in einem eigenen Format gespeichert und können nicht konvertiert werden.
- Die Informationen der .APR-Dateien sind für MicroBase nicht lesbar. Sie definieren u. a. die neuen Feldtypen und die Eingabemasken (Formulare).

\*

Das Programm »DBF2ASC« erzeugt aus der dBASE-Datenbank eine sequentielle Datei, die von jedem Programm gelesen werden kann. Sie müssen folgende Angaben machen:

- Feldtrennzeichen (Vorgabe ist ein Komma): geben Sie den ASCII-Code des gewünschten Zeichens ein. Dieses Zeichen steht nach jedem Datenfeld.
- Feldbegrenzer (Vorgabe: doppelte Anführungszeichen): siehe oben. Das Zeichen steht vor und nach jedem Datenfeld.

Am Ende eines Datensatzes wird immer ein Linefeed (ASCII 10) ausgegeben.

Einschränkungen

- die maximale Feldlänge beträgt z. Zt. 4 KByte
- die Anzahl der Datenfelder darf 2 048 nicht überschreiten
- die Anzahl der Datensätze wird nur durch AmigaDOS begrenzt

Beide Programme benötigen die "hbasic2.library".

## 1.27 dBASE- und FoxPro-Datenbanken anzeigen

Das Programm »dBView« liest eine Datenbank im dBASE- oder FoxPro-  
Format und zeigt sie auf dem Bildschirm an.

Zuvor werden einige Informationen über die Datenbankdatei ausgegeben: verwendete Programmversion, Erstellungsdatum, Anzahl und Art der Felder usw.

Tastaturbelegung während der Datensatzanzeige:

q Programm beenden!



```
+          1 Datensatz vorblättern
-          1 Datensatz zurückblättern

*          10 Datensätze vorblättern
-          10 Datensätze zurückblättern
```

Näheres siehe

dBASE- und FoxPro-Datenbanken konvertieren

.

Die "hbasic2.library" wird im Verzeichnis LIBS: benötigt.

## 1.28 Tips & Tricks

Folgende Themen stehen zur Auswahl:

Datenfelder hinzufügen oder löschen

Mehr als 65 535 Datensätze verwenden

Datenaustausch mit MicroWrite

Änderung des Passwortes einer Datenbank

Datenverluste

Tips zum Importieren / Exportieren einer sequentiellen Datei

## 1.29 Datenfelder hinzufügen oder löschen

Hinweis: der Name der Datenbank wird mit »Test.DAT« angenommen.

- 1 Duplizieren Sie die Dateien »Test.DEF« , »Test.NUM« unnd »Test.PTR« z. B. unter den Namen »Test1.DEF« usw.

(Lesen Sie unter "Dateiformate" nach, welche Dateien noch kopiert werden müssen. Diese sind dort als "feldabhängig" gekennzeichnet. In diesen Dateien müssen ggf. Änderungen vorgenommen werden; s. u.)

- 2 Edieren Sie die Datei »Test1.PTR«. Ändern Sie den Wert in der zweiten Zeile auf null (0).

- 3 Edieren Sie die Datei »Test1.DEF«. Fügen Sie am Dateiende drei Zeilen hinzu - eine, die den Feldnamen in Anführungszeichen (") enthält, eine mit der Feldlänge, eine mit dem Feldtyp (T,N,Z,D,B,X,L). Ändern Sie die beiden Zahlen am Anfang der Datei: erhöhen Sie die erste um eins (1), die zweite um die Länge des neuen Feldes.

Beispiel:

---

```

1,10          <-- vor der Aktualisierung
"Artikel"
10
"T"

2,15          <-- nach der Aktualisierung
"Artikel"
10
"T"
"Preis"
5
"N"

```

(Beachten Sie bitte die Einschränkungen im Abschnitt  
 Interna  
 .)

- 4 Erzeugen Sie eine leere Datenbankdatei »Test1.DAT«. Dies geht am einfachsten mit Ed.
- 5 Je nach Feldtyp müssen Sie noch Eintragungen in einer speziellen Datei vornehmen (siehe 1.)
- 6 Rufen Sie MicroBase auf und öffnen Sie die alte Datenbankdatei, in unserem Beispiel »Test.DAT«.
- 7 Exportieren Sie die Datenbankdatei unter einem beliebigen Namen:
  - a Die Abfragen nach dem Separator und der Dateiondung bestätigen Sie mit Return.
  - b Nach der Dateiauswahl drücken Sie bitte einmal »j« und einmal »n«, um die korrekten Optionen zu wählen.
- 8 Öffnen Sie die neue Datei (»Test1.DAT«). In der Titelleiste von MicroBase müsste eine Null (0) erscheinen, die eine leere Datei kennzeichnet.
- 9 Importieren Sie die zuvor exportierte Datei (beantworten Sie die Frage durch einen Druck auf die Taste »n«). Überprüfen Sie die Anzahl der Datensätze sowie auf korrekten Datenbankinhalt.

Das Löschen eines Feldes geschieht analog hierzu.

Datenfelder können nur am Ende des Datensatzes hinzugefügt oder gelöscht werden.

### 1.30 Mehr als 65 535 Datensätze verwenden

Normalerweise verwendet MicroBase nur 65 535 Datensätze, um möglichst wenig Arbeitsspeicher zu beanspruchen. Da ein Datensatz 1 Bit benötigt, reichen hierfür 8 Kbyte aus (zusätzlich zu den ca. 180 KByte, die das Programm für sich selbst benötigt).

Falls Sie jedoch mehr Datensätze benötigen, führen Sie folgende Schritte aus:

- 1 Starten Sie MicroBase.
- 2 Schalten Sie zur Workbench um und wählen Sie dort den Menübefehl "Workbench/Befehl ausführen..." Tippen Sie  

```
avail
```

ein und notieren Sie die Größe des größten zusammenhängenden Speicherblocks, z. B. 267 678.
- 3 Erstellen Sie in MicroBase die gewünschte Datei.
- 4 Verlassen Sie MicroBase.
- 5 Multiplizieren Sie den notierten Wert mit 8 und addieren Sie ihn zu 65 535.

```

2 141 424
  65 535
-----
2 206 959
```
- 6 Runden Sie den Wert sicherheitshalber etwas nach unten ab, in unserem Beispiel z. B. auf 2 000 000.
- 7 Laden Sie die zu Ihrer Datenbankdatei gehörige ".PTR"-Datei in einen Texteditor Ihrer Wahl (Ed reicht hier völlig aus) und ändern Sie die Zahl in der ersten Zeile ab (Leerzeichen sind nicht erlaubt!)
- 8 Starten Sie MicroBase und öffnen Sie Ihre Datei. Wenn Sie alles richtig gemacht haben, müsste bei Anwahl des Menüpunktes "Projekt/Info" die neue maximale Datensatzanzahl erscheinen.

Bedenken Sie, dass selbst bei 8 MB Speicher nur ca. 60 Millionen Datensätze möglich sind.

Alternativ können Sie den 'virtuellen Modus' benutzen, der Speicher auf die Festplatte auslagert und über zwei Milliarden Datensätze ermöglicht. Sie aktivieren ihn mit dem Befehl "Optionen/Festlegen".

Ohne Keyfile sind trotzdem nur maximal 20 Datensätze möglich.

### 1.31 Datenaustausch mit MicroWrite

Der Datenaustausch muss über eine sequentielle Datei erfolgen, deren Aufbau beide Programme "verstehen". Benutzen Sie am besten das vorgegebene Format.

Die Datenbankdatei darf maximal 6 Felder haben (bzw. acht, siehe unter 3.)

Datei aus MicroBase exportieren

- 1 Übernehmen Sie einfach die Vorgaben. Lassen Sie Anführungszeichen und Feldbezeichnungen speichern.

Datei in MicroWrite importieren, bearbeiten und exportieren

- 1 Wählen Sie den Befehl "Daten importieren" im Menü Tools und danach die Option "Auswahl...", um dann die Vorgabe zu bestätigen. Übernehmen Sie die Feldbezeichnungen.
- 2 Bearbeiten Sie die Datei, bzw. verwenden Sie sie zum Seriendruck.
- 3 Haben Sie die Datei verändert, so wählen Sie analog zum Importieren den Befehl "Daten exportieren" im gleichen Menü.

Datei in MicroBase importieren

- 1 Erstellen Sie eine neue Datei mit folgenden Merkmalen (falls Sie noch keine haben) oder überschreiben Sie den Inhalt von "Beispiel.DAT".

| Feldlänge | Feldname    |
|-----------|-------------|
| -----     | -----       |
| 30        | Nachname    |
| 30        | Vorname     |
| 30        | Straße      |
| 20        | Telefon-Nr. |
| 30        | PLZ Wohnort |
| 20        | Anrede      |
| 20        | Anrede 2*   |
| 20        | Anrede 3*   |

Die beiden Felder mit Stern (\*) sind dabei optional. Alle Felder sind vom Typ Text (auch die Telefonnummer, da sie Leerzeichen und runde Klammern enthalten kann.)

- 2 Jetzt können Sie die Datei importieren. Lassen Sie den ersten Satz überlesen, da er wirklich die Feldbezeichnungen enthält. Ignorieren Sie die Meldung, dass zu viele Felder vorhanden sind und klicken Sie auf "Importieren".
- 3 Geben Sie jetzt die Nummer des Datensatzes in der MicroBase-Datei an, ab dem importiert werden soll (Vorgabe ist, dass neue Datensätze angefügt werden.) Geben Sie eins (1) ein, um den alten Dateiinhalte zu überschreiben.

## 1.32 Änderung des Passwortes der Datenbank

Führen Sie folgende Schritte aus:

- 1 Öffnen Sie die Datenbank wie gewohnt.
- 2 Exportieren Sie die Datenbank in einem beliebigen Format. Die Datensätze sollten durch Anführungszeichen begrenzt und die Feldbezeichnungen \*nicht\* gespeichert werden.
- 3 Schließen Sie die Datenbank.

- 4 Öffnen Sie die Datenbank mit dem neuen Passwort. Sie werden bemerken, dass die vorhandenen Datensätze jetzt unlesbar sind.
- 5 Importieren Sie die unter (2) exportierte Datenbankdatei. Lassen Sie ab dem Datensatz eins (1) importieren. Der erste Datensatz darf nicht überlesen werden.
- 6 Überprüfen Sie die Inhalte der Datensätze stichprobenartig. Sie müssten jetzt korrekt angezeigt werden.
- 7 Sie können jetzt weitere Datensätze eingeben oder die Datenbank schließen.

### 1.33 Datentypen und Eingabeformate

#### Datentypen

Folgende Datentypen stehen zur Verfügung:

| # | Typ        | Kennung | Länge   | Format                                                                                               |
|---|------------|---------|---------|------------------------------------------------------------------------------------------------------|
| 1 | Text       | T       | 1..2048 | Buchstaben, Zahlen, Sonderzeichen usw.                                                               |
| 2 | Nummer     | N       | 1..19   | Zahlen in einem der unterstützten Formate: dezimal, wissenschaftlich, binär, hexadezimal, oktal. (1) |
| 3 | Datum      | D       | 10      | TT.MM.JJJJ (etwa '29.03.1997') (2)                                                                   |
| 4 | Zeit       | Z       | 8       | HH:MM:SS (etwa '10:00:28')                                                                           |
| 5 | Extern     | X       | 1..2048 | Dateipfad (3)                                                                                        |
| 6 | Berechnung | B       | 1..19   | Formel (4)                                                                                           |
| 7 | Logisch    | L       | 1       | Wahrheitswerte (5)                                                                                   |
| 8 | Relation   | R       | 1       | Verknüpfung mit Datenbankdatei (6)                                                                   |

Sie können mit den Feldoptionen bei der Neuerstellung der Datenbank bzw. einer Änderung der Struktur festlegen, ob der eingegebene Text z. B. in Großbuchstaben gewandelt werden soll.

- (1) Die Beschränkung von Zahlen auf 19 Stellen (inklusive Dezimalpunkt) ist durch dBASE sowie Amiga OS bedingt. Längere Zahlen müssen Sie - allerdings auf Kosten der Genauigkeit - im sog. wissenschaftlichen Format eingeben.

Sie können den erlaubten Bereich der Zahlen mit den Feldoptionen festlegen (siehe Beschreibung des Befehls "Projekt/Neu".) Bei Preisen in DM können Sie z. B. negative Werte vermeiden, indem Sie die Option '>0' benutzen. Für die Ausgabe in Listen bzw. auf Etiketten können auch

Zahlenformate  
definiert werden.

- (2) MicroBase wandelt Datumsangaben während eines Sortiervorganges in serielle Zahlen um (sog. "julianisches Datum"). Das julianische Datum ist definiert als die Anzahl der Tage seit dem 31.12.1899 24:00:00 Uhr.

Obwohl die heute verwendete gregorianische Zeitrechnung erst seit dem 05.10.1582 gültig ist, können frühere Daten trotzdem verwendet werden.

In MicroBase sind theoretisch Datumsangaben zwischen dem 05.02.-27376609 und dem 25.01.27380970 möglich; aus praktischen Gründen ergibt sich ein Zeitraum zwischen dem 01.01.-999 und dem 31.12.9999.

Bei der Eingabe eines Datums wird es automatisch auf seine Gültigkeit überprüft. Der 29.02. wird nur bei einem Schaltjahr akzeptiert.

- (3) Externe Felder speichern lediglich den Pfad einer Datei, z. B. »Work:MicroPaint/Bilder/Demo.IFF«. Der bei der Datenbankdefinition gewählte Viewer bestimmt, was mit dieser Datei geschieht. Dies entspricht einem OLE-Feld in Datenbanksystemen, die unter Windows [95] betrieben werden.

Felder des Typs "extern" können natürlich auch Texte (= "Memos"), Source-Codes usw. beinhalten.

- (4) Felder des Typs "Berechnung" verhalten sich wie Felder des Typs "Nummer", können jedoch nicht ediert werden. Sie erhalten ihren Inhalt durch eine Formel, die bei der Definition der Datenbankdatei eingegeben wurde.

In den Formeln kann über die Buchstaben 'A' bis 'Y' auf die Inhalte der Felder zugegriffen werden.

Für die Ausgabe in Listen bzw. Etiketten können auch  
Zahlenformate  
definiert werden.

Beispielsweise könnten in einer Artikeldatenbank zwei Felder namens "Nettopreis" (Typ "Nummer") und "Bruttopreis" (Typ "Berechnung") vorkommen. Die Formel des Feldes "Bruttopreis" lautet nun 'B \* 1.15', wenn "Nettopreis" das zweite Feld der Datenbank ist. Auf diese Weise muss man nur die Formel ändern, wenn sich die Mehrwertsteuer ändert. Für die Mehrwertsteuer sollte in der Datei »MicroBase.CVD« im Programmverzeichnis eine Konstante definiert werden, die dann für alle Datenbankdateien gilt.

Sie sollten die  
»rexxmathlib.library«  
installieren, um mehr Funktionen zur Verfügung zu haben.

- (5) Felder des Typs "Logisch" speichern Wahrheitswerte. Diese können Sie in folgenden Formaten eingeben:

| wahr  | falsch |                        |
|-------|--------|------------------------|
| ----- |        |                        |
| W     | F      | wahr, falsch           |
| J     | N      | Ja, Nein               |
| T     | F      | TRUE, FALSE            |
| Y     | N      | Yes, NO                |
| 1     | 0      |                        |
| ?     |        | (nicht initialisiert?) |

Diese Formate bestehen zum Teil nur aus Kompatibilitätsgründen zu dBASE.

- (6) Relationsfelder enthalten ihren aktuellen Inhalt über eine Verknüpfung mit einer zweiten Datenbank. Das in der aktuellen Datenbank reservierte Byte ist für eventuelle künftige Erweiterungen vorgesehen.

### Zahlenformate

Zahlen können Sie in MicroBase im dezimalen, hexadezimalen, binären und oktalen oktalen Format eingeben:

| Eingabe | Format      | Basis | Wert | Typ       |
|---------|-------------|-------|------|-----------|
| 11      | dezimal     | 10    | 11   | Kommazahl |
| &H11    | hexadezimal | 16    | 17   | Ganzzahl  |
| &X11    | binär       | 2     | 3    | Ganzzahl  |
| &O11    | oktal       | 8     | 9    | Ganzzahl  |

### Wissenschaftliche Zahlenschreibweise

Dezimale Zahlen können auch in der sogenannten wissenschaftlichen Schreibweise eingegeben werden. In diesem Fall folgt der Zahl ein 'e' oder 'E' sowie eine weitere Zahl (die allerdings kein Komma und keinen Punkt enthalten darf), die den zu verwendenden Exponenten zur Basis 10 angibt.

So steht z. B. '2e5' für 200 000.

### Wertebereiche

Die gültigen Wertebereiche für Zahlen sind wie folgt:

- dezimal, wissenschaftlich:  $-10^{309}..+10^{309}$
- binär, oktal, hexadezimal:  $-2^{31}..+2^{31}-1$

Über die Feldoptionen können Sie engere Wertebereiche definieren.

### Eingabeformate

Für die Typen Datum und Zeit können zur besseren Orientierung sog. Eingabeformate angezeigt werden (' . .' bzw. ' : :'); dies wird jedoch nicht empfohlen. In diesem Fall wird bei der Eingabe der Einfügemodus deaktiviert.

Für jeden Datentyp können Vorgabewerte wie z. B. automatische Seriennummern definiert werden.

### Sortieren

MicroBase schlägt beim Sortieren den passenden Datentyp vor. Bei externen und logischen Feldern wird die Sortiermethode "Text" vorgeschlagen.

Vorschläge für Anwendungen

Sie können Datenbankdateien mit externen Feldern z. B. für folgende Anwendungen einsetzen:

- Wie wäre es mit Adress- oder Artikeldateien, die auch ein Bild der Person bzw. der Ware enthalten?
- Benutzen Sie doch Dokumentverwaltungen, in denen Sie neben dem Dokumentnamen zusätzlich Autoren, Stichworte, Themen, Datum usw. speichern, um den Überblick über Ihre Dokumente (Texte, Bilder, Kalkulationen usw.) nicht zu verlieren. Sie können z. B. nur bestimmte Kategorien anzeigen lassen, Texte suchen oder nach bestimmten Kriterien sortieren.
- Sie können Bilderkataloge erstellen und ausdrucken.

## 1.34 Der 'virtuelle Modus'

Aktivierung des 'virtuellen Modus'

Der virtuelle Modus wird mit dem Befehl "Optionen/Festlegen..." aktiviert und beim Erstellen bzw. Öffnen der nächsten Datenbankdatei wirksam. Bei der Demo-Version von MicroBase ist der virtuelle Modus nicht möglich.

Der Standardmodus

Normalerweise merkt sich MicroBase die selektierten Datensätze in einer Matrix (Feld, Array), welches sich immer im RAM befindet. Für jeden möglichen Datensatz wird nur ein Bit belegt; die maximal möglichen 65 535 Datensätze belegen also  $1 \cdot 65\,535 / 8 = 8\,192$  Bytes. Auf diese Weise sind bei 8 MB freiem RAM etwa 60 Millionen Datensätze möglich.

Der 'virtuelle Modus'

Im virtuellen Modus wird dieses Array auf die Festplatte ausgelagert. Jeder Datensatz belegt dort ein Byte; das Array im RAM entfällt. Auf diese Weise sind mehr als 2 Milliarden Datensätze bei einer max. Dateilänge von 4 GB möglich.

Trotz häufigerer Zugriffe auf die Festplatte wird MicroBase im 'virtuellen Modus' selbst bei aktiver Selektion nur geringfügig verlangsamt. Stellen Sie ggf. für das betroffene Laufwerk mehr Pufferspeicher bereit (siehe den Befehl 'Addbuffers' im AmigaDOS-Handbuch.) Sie sollten Ihr Laufwerk außerdem regelmäßig defragmentieren lassen (siehe Abschnitt

Datenverluste  
)

## 1.35 Datenverluste

---



Beim Importieren, Exportieren, Extrahieren, Selektieren und Sortieren einer Datenbank kann es unter Umständen zu einem Datenverlust kommen (z. B. durch einen Absturz des Computers oder Stromausfall.) Dies kann unter Umständen auch nach dem Verlassen des Programms der Fall sein.

Gehen Sie wie folgt vor:

- 1 Starten Sie den Computer neu. Eventuell wird jetzt der betroffene Datenträger von AmigaDOS "validiert", was unter Umständen einige Minuten dauern kann.
- 2 Starten Sie MicroBase mit deaktiviertem 'virtuellen Modus' und öffnen Sie die beschädigte Datei. Achten Sie ggf. auf korrekte Passwörter.
- 3 Sie können die Datenbank manuell reparieren oder dies von MicroBase automatisch erledigen lassen.

a) manuell

Sehen Sie sich alle Datensätze an. Beschädigte Datensätze sind meist daran zu erkennen, dass der Bildschirm aufblitzt bzw. die Eingabemaske unlesbar ist. Edieren Sie die zerstörten Datensätze und entfernen Sie in jedem Datenfeld den (unsichtbaren) "Datenmüll", indem Sie vor brauchbaren Zeichenfolgen <Backspace> ("<-")> und dahinter <Del> mehrmals drücken. Völlig zerstörte Datenfelder sollten Sie mit <Esc> löschen und die Daten neu eingeben.

b) automatisch

Rufen Sie einfach den Befehl "Extras/Reorganisieren" auf. Er führt die oben beschriebenen Operationen nach einer Sicherheitsabfrage automatisch bei allen vorhandenen Datensätzen aus. Dieser Vorgang kann einige Zeit in Anspruch nehmen.

- 4 Schließen Sie die Datenbankdatei.

MicroBase hält das Risiko eines Datenverlustes sehr gering, da jeweils nur ein Datensatz bearbeitet wird; es kann jedoch nicht völlig ausgeschlossen werden.

Vorbeugende Maßnahmen

- a Erstellen Sie Sicherheitskopien sensibler Dateien auf anderen Datenträgern. Hierfür eignen sich z. B. ABackup oder Commodores "HDBackup".
  - b Greifen Sie nicht mit mehreren Programmen auf denselben Datenträger zu.
  - c Schützen Sie Ihre Datenbankdateien mit Passwörtern, damit sie nicht jeder bearbeiten kann.
  - d Löschen Sie importierte Dateien anderer Formate erst nach einer sorgfältigen Prüfung der entstandenen Datenbankdatei.
  - e Unterbrechen Sie das Programm nicht während eines Zugriffs auf die Datenbankdatei.
-

- f Prüfen Sie regelmäßig Ihre Datenträger (z. B. mit DOSControl oder auch DiskSalv).
- g Um eine maximale Geschwindigkeit zu erreichen, sollten Sie Ihre Datenträger regelmäßig defragmentieren – zum Beispiel mit DOSControl oder ReOrg. Sie können auch ein Cache-Programm aktivieren, das die Geschwindigkeit des Datenträgers – vor allem beim sequentiellen Zugriff – durch Vorablesen der nächsten Daten erhöht. Bei Direktzugriffsdateien, wie sie von MicroBase erstellt werden, sind Caches nicht immer wirksam (eben weil die Daten nicht sequentiell gelesen werden.) Caches können jedoch die Aktualisierung von Relationen verbessern, da sie den Zugriff auf die verknüpfte Datenbank und die als sequentielle Dateien gespeicherten Caches beschleunigen.
- h Lassen Sie Ihre Datenbankdateien reorganisieren, um "Datenmüll" zu entfernen. Dies sollte jedoch eher selten notwendig sein.
- i Starten Sie neben MicroBase möglichst kein anderes Programm.
- j Extrahieren Sie Ihre Datenbanken in Sicherungskopien.

### 1.36 Tips zum Importieren / Exportieren einer sequentiellen Datei

Beim Im- und Exportieren einer sequentiellen Datei haben Sie folgende Optionen:

- Sie können das Feldtrennzeichen frei definieren
- die Feldnamen können optional als erster Datensatz gesichert werden
- die Feldinhalte können durch Anführungszeichen begrenzt werden. Dies ist dann sinnvoll, wenn in den Feldern Leer- und Satzzeichen (z. B. Kommata) enthalten sind

Probleme beim Importieren von Datensätzen

- In den Datenfeldern sind keine Anführungszeichen erlaubt
- Teilweise können Probleme auftreten, wenn die Datenfelder Satzzeichen wie z. B. Semikola enthalten

Problemlösung

Es wird empfohlen, die Datenfelder mit Anführungszeichen zu begrenzen und das standardmäßig vorgegebene Tabulatorzeichen als Datensatztrenner zu benutzen.

Falls in Ihrer sequentiellen Datei alle Felder durch Anführungszeichen begrenzt sind, Sie jedoch einen anderen Feldtrenner benutzen, dann gehen Sie bitte wie folgt vor:

Anmerkung: Der Dateiname wird mit »Test.ASC« angenommen, das Satztrennzeichen mit ', ' (Komma). <Tab> bedeutet, dass Sie hier auf die Tabulatortaste drücken sollen. <Return> bezeichnet die Return- bzw. Enter-taste.

- 1 Starten Sie den von Commodore mitgelieferten Texteditor "MEMacs". Er befindet sich entweder auf der »Extras«-Diskette oder in der Schublade »Tools« Ihres Startlaufwerks.
- 2 Wählen Sie den Befehl "Project/Read-File", um die Datei zu laden. Tippen Sie den gewünschten Dateinamen ein (in unserem Fall »Test.ASC«) und drücken Sie auf <Return>.
- 3 Rufen Sie den Befehl "Search/Search-Replace" auf und geben Sie Folgendes ein:
 

```
Search:      ", "<Return>
Replace:     "<Tab>"<Return>
```

MEMacs ersetzt jetzt die Datenfeldtrennzeichen.
- 4 Beenden Sie das Programm mit dem Befehl "Project/Save-exit".
- 5 Sie können die Datei jetzt ohne Probleme in MicroBase importieren.

## 1.37 Tutorial

Willkommen zum MicroBase-Tutorial!

Wir möchten Ihnen zeigen, wie man mit MicroBase z. B. eine Bilderliste erstellt. Dafür benötigen wir die Dateien im »Bilder«-Verzeichnis.

Für die folgenden Ausführungen setzen wir voraus, dass Sie bereits mit der grundsätzlichen Bedienung des Amiga vertraut sind.

Eine Datenbank anlegen

Zuerst müssen wir uns den grundsätzlichen Aufbau unserer Datenbankdatei überlegen, denn nachträgliche Änderungen an der Datenbankstruktur sollten möglichst vermieden werden.

In unserer Datei möchten wir zu einem Bild zusätzlich eine Beschreibung abspeichern. Hierfür bieten sich die Datentypen Extern (für den Pfad zum Bild) und Text an. Die Beschreibung soll nur kurz sein, der Pfad kann jedoch – vor allem bei CDs – sehr lang werden.

Rufen Sie also den Befehl "Neu" im Menü "Projekt" auf und wählen Sie einen Dateinamen, z. B. »Tutor«. Geben Sie danach eine "2" ein, weil wir zwei Felder erstellen wollen. MicroBase fordert Sie nun zur Eingabe der Felddefinitionen auf. Die folgende Tabelle zeigt Ihnen, was Sie eingeben müssen (betätigen Sie <Esc>, um ein Eingabefeld zu löschen):

| Feldname  | Feldtyp | Feldlänge | Vorgabewert |
|-----------|---------|-----------|-------------|
| Bemerkung | T       | 40        |             |
| Bild      | X       | 255       | \$          |

Falls Sie schon Kickstart 3.0 oder höher haben, bestätigen Sie den vorgegebenen Viewer mit <Return>. Dieser dient später zum Anzeigen (bzw. Bearbeiten) der Bilder. Sonst müssen Sie leider den Pfad zu Ihrem Bildanzeigeprogramm eingeben bzw. ein Dollarzeichen für eine Dateiauswahlbox eintippen. MicroBase stellt einige Viewer für OS 2.X zur Verfügung.

MicroBase fragt Sie nun nach einem Passwort. Sie können eines eingeben, wenn Sie möchten – aber Sie dürfen es niemals vergessen! Die Datenbankdatei würde sonst unbrauchbar und alle Daten gingen verloren. (Niemand kann Ihnen in einem solchen Falle helfen!!!)

Nach einer Sicherheitsabfrage wird die Datei erzeugt.

#### Datensätze eingeben

Zur Eingabe der Datensätze müssen Sie nun die Funktion "Bearbeiten/Eingeben" aktivieren. Es wird eine Maske mit den maximalen Abmessungen der Datenfelder angezeigt.

Geben Sie nun den Titel des ersten Bildes ein: 'MicroPaint-Demobild'. Drücken Sie auf <Return> – dadurch wird die Eingabe dieses Datenfeldes abgeschlossen und Sie gelangen zum nächsten. Hier sehen Sie schon ein Dollarzeichen. Bitte drücken Sie gleich auf <Return>. Sofort erscheint eine Dateiauswahlbox. Wählen Sie die Datei »Demo.IFF« im Verzeichnis »Bilder« aus.

Drücken Sie auf »J«, um den Datensatz zu speichern. Geben Sie nun den zweiten Datensatz ein.

Die Bemerkung soll 'MicroBase-Fenster' lauten; das Bild heißt »MicroBase.IFF«. Drücken Sie auf »J« und »N«, um den Datensatz zu speichern und den Eingabemodus zu verlassen.

#### Die Datenbank sortieren

Wir möchten nun die Datensätze nach der Bemerkung sortieren lassen. Hierfür ist der Menüpunkt "Bearbeiten/Sortieren" zuständig. Drücken Sie einfach auf <Return>, da das richtige Feld bereits vorgegeben ist. Als Sortiermethode ist [T]ext vorgeschlagen. Dies ist auch richtig, da der Pfad der Bilder als normaler Text gespeichert wird. Die Sortierrichtung "+" (für aufsteigend) soll ebenfalls übernommen werden. MicroBase sortiert nun innerhalb kurzer Zeit die Datei, worüber Sie sich mit dem Menüpunkt "Ansehen" vergewissern können.

#### Datensätze ansehen, Bilder anzeigen!

Sie können sich mit dem Befehl "Bearbeiten/Ansehen" Ihre Datensätze ansehen und mit den Cursortasten zwischen ihnen blättern. Uns interessiert aber, ob MicroBase mit den eingegebenen Daten auch etwas anfangen kann. Da unsere Bilder im zweiten Feld eingetragen sind, drücken Sie jetzt bitte gleichzeitig auf <CTRL> und auf <B>. MicroBase zeigt nun mit dem zuvor eingestellten Viewer das Bild an.

#### Listen erzeugen

---

Zum Schluss möchten wir noch eine Liste unserer Bilder anfertigen, damit wir später auch wissen, in welchem Verzeichnis sie sich befinden.

Rufen Sie dazu bitte zunächst den Befehl "Ausgabe/Listen/Edieren" auf. Die Liste soll nur eine Zeile lang sein. Im Listen-Editor geben Sie bitte folgende Zeile ein:

```
&A $B
```

Verlassen Sie bitte jetzt den Editor und wählen Sie die Funktion "Drucken" im Untermenü "Ausgabe/Listen" an. Drücken Sie bitte auf <Escape> und tippen Sie ein:

```
RAM:Tutor.MBO
```

Die Liste wird nun nicht ausgedruckt, sondern in diese Datei geschrieben.

Den "Ausdruck" prüfen

Bitte rufen Sie jetzt noch den Befehl "Neues CLI" im "Spezial"-Menü auf. Geben Sie bitte

```
Ed RAM:Tutor.MBO
```

ein. Sie sehen, dass die Liste korrekt erzeugt wurde. Verlassen Sie nun den Editor sowie anschließend das CLI, indem Sie

```
Endcli
```

eintippen.

Ausblick

Wir sind nun am Ende des Tutorials angelangt. MicroBase bietet jedoch weit mehr, als wir Ihnen gezeigt haben. So können Sie z. B. nach Feldinhalten suchen und sie durch andere ersetzen lassen usw. usf.

Das Lernprogramm führt Ihnen das Tutorial visuell vor.

Viel Spaß mit MicroBase!

## 1.38 Die »rexxmathlib.library«

Sie sollten die »rexxmathlib.library« installieren, um in den Berechnungsformeln mehr Funktionen zur Verfügung zu haben.

Die vollständige englischsprachige Dokumentation finden Sie in dem Verzeichnis »RexxMathLib\_1.3«. Die Library für Workbench 1.2 wurde aus dem Archiv entfernt, Sie können Sie jedoch auf der Aminet CD 4 finden.

Auf dem Aminet können Sie neuerdings auch eine optimierte Version der Library finden.

---

Folgende Funktionen sind vorhanden:

| Funktion                | Aufruf     | Alias (e)   | Voraussetzungen    |
|-------------------------|------------|-------------|--------------------|
| Absolutewert            | abs(x)     | fabs        |                    |
| Arcuscosinus            | acos(x)    |             | -1.0 <= x <= 1.0   |
| Arcuscosinus hyperbol.  | acosh(x)   |             | x >= 1.0           |
| Arcussinus              | asin(x)    |             | -1.0 <= x <= 1.0   |
| Arcussinus hyperbolicus | asinh(x)   |             |                    |
| Arcustangens            | atan(x)    |             |                    |
| Arcustangens hyperb.    | atanh(x)   |             | -1.0 < x < 1.0     |
| Arcustangens y/x        | atan(y,x)  |             | y != 0.0, x != 0.0 |
| Kleinste Ganzzahl > x   | ceil(x)    |             |                    |
| Cosinus                 | cos(x)     |             |                    |
| Cosinus hyperbolicus    | cosh(x)    |             |                    |
| Cotangens               | cot(x)     | cotan       |                    |
| Cotangens               | cotan(x)   | cot         |                    |
| Cosecans                | csc(x)     |             |                    |
| e hoch x                | exp(x)     |             |                    |
| Absolutewert            | fabs(x)    | abs         |                    |
| Fakultät                | fact(x)    |             | x >= 0, x < 171    |
| Größte Ganzzahl < x     | floor(x)   | int         |                    |
| Größte Ganzzahl < x     | int(x)     | floor       |                    |
| natürlicher Logarithmus | ln(x)      | log         | x > 0.0            |
| natürlicher Logarithmus | log(x)     | ln          | x > 0.0            |
| Logarithmus             | log10(x)   |             | x > 0.0            |
| Nächste Ganzzahl bei x  | nint(x)    |             |                    |
| x hoch y                | pow(x,y)   | power, xtoy | x >= 0.0           |
| x hoch y                | power(x,y) | pow, xtoy   | x >= 0.0           |
| Secans                  | sec(x)     |             |                    |
| Sinus                   | sin(x)     |             |                    |
| Sinus hyperbolicus      | sinh(x)    |             |                    |
| Quadratwurzel           | sqrt(x)    |             | x >= 0.0           |
| Tangens                 | tan(x)     |             |                    |
| Tangens hyperbolicus    | tanh(x)    |             |                    |
| x hoch y                | xtoy(x,y)  | pow, power  | x >= 0.0           |

### 1.39 cbsi

»CBSI« ist ein Systeminformations-Utility, das aus dem CLI gestartet werden muss.

Übergabe von Argumenten

Wenn Sie das Programm mit dem Argument '?' aufrufen, wird folgender Hilfetext angezeigt:

```
Crossbones System-Information 1.0.7
```

```
Aufruf: CBSI [Laufwerk [Blocklänge]]
```

CBSI ermittelt einige System-Informationen, z. B. CPU, FPU, Chipset, Betriebssystem und die Prozessorgeschwindigkeit.

Zusätzlich können Sie die Übertragungsrate eines beliebigen Laufwerks bestimmen lassen. Wenn Sie die Blocklänge nicht angeben, so wird ein Standardwert von 1 MByte benutzt (die erzeugte Datei besteht aus 8 Blöcken.)

Standardmäßig wird kein Laufwerk geprüft. Sie sollten beachten, dass die erzeugte Datei - wie oben beschrieben - aus 8 Blöcken besteht. Die oben erwähnte Datei wäre also 8 MB groß. Der Standardwert ist ein optimaler Kompromiss zwischen Dateigröße und Genauigkeit des Ergebnisses.

Ein Beispiel

Auf meinem Rechner erscheint folgender Bericht:

Crossbones System-Information 1.0.7

Chip-RAM: 1,911,312 Bytes  
Fast-RAM: 15,806,872 Bytes

Attn-Flags: 1000000001111111  
CPU: Motorola 68040  
FPU: Motorola 68040 FPU  
Cache: INST: Cache Burst DATA: Cache Burst CopyBack

Betriebssystem: Amiga OS 3.1  
Kickstart: Version 40.68  
Workbench: Version 40.42

RevBits: 00011111  
DMA: AA Alice  
Display: AA (M)Lisa  
Chipset: AA

MIPS: 19.8303  
Amiga 500 Index: 28.09302

Datenträger "hdl":  
Schreibzugriff: 1567 KB/s  
Lesezugriff: 1424 KB/s

Workbench Screen: \$070B8580  
Format: 640 x 256 x 64  
Modus: \$00029000

Die untersuchten Eigenschaften

Der Bericht besteht aus folgenden Abschnitten:

1 Größe des freien Chip- und Fast-RAMs in Bytes.

---

- 2 Ausgabe der Attn-Flags (exec.library). Sie enthalten Informationen über CPU und Co-Prozessoren.
    - a CPU: Motorola 68000 bis Motorola 68060
    - b FPU: Motorola 68881|2, 68040, 68060. Wenn beim 68040 bzw. 68060 die Coprozessor-Emulation (68040|68060.library) fehlt, wird der Zusatz <native> ausgegeben. In diesem Falle können nur noch Befehle der internen FPU benutzt werden, die des 68881|2 funktionieren nicht mehr.
    - c Cache: Für Daten ("DATA") und Befehle ("INST") wird angegeben, ob ein Cache eingeschaltet ist. Bei beiden Caches ist ein "Burst" möglich, bei Daten zusätzlich "CopyBack". Externe Caches werden auch erkannt.
  - 3 Ausgabe der Betriebssystem-Version (z. Zt. 1.0 bis 3.1). Es werden die Versionsnummern von Kickstart und Workbench angezeigt.
  - 4 Die RevBits (graphics.library) werden angezeigt. Sie geben Auskunft über die Coprozessoren.
    - a DMA: Die Chips Agnus, ECS Agnus und AA Alice werden erkannt.
    - b Display: Denise, Hires Denise, AA Lisa oder AA MLisa.
    - c Chipset: OCS (= original), ECS, A oder AA werden erkannt.
  - 5 Die Geschwindigkeit des Systems wird gemessen. Sie sollten alle anderen Programme beenden, um einen korrekten Wert zu erhalten.
    - a MIPS: Gibt an, wie viele Millionen Befehle die CPU in einer Sekunde bearbeiten kann.
    - b Amiga 500 Index: Vergleicht die Geschwindigkeit des Rechners mit der eines Amiga 500.
  - 6 Die Geschwindigkeit eines Datenträgers wird gemessen. Dazu wird eine Datei erzeugt und gelesen.
    - a Schreibzugriff: Erzeugt eine Datei und ermittelt die Schreibgeschwindigkeit in Kilobytes pro Sekunde.
    - b Lesezugriff: Liest die erzeugte Datei und errechnet die Lesegeschwindigkeit.

Anschließend wird die Datei gelöscht. Beide Werte sollten mindestens bei 1 400 KB/Sekunde liegen, wenn der untersuchte Datenträger eine Festplatte ist.
  - 7 Zeigt Informationen über den Workbench Screen an (zuerst die Adresse.)
    - a Format: Breite x Höhe x Farben des Screens.
    - b Modus: Aktueller Modus-ID des Screens.
-



## 1.40 Die mitgelieferten Viewer

Benutzer von OS 2.X können MultiView nicht benutzen. Zu MicroBase werden deshalb die Viewer des MicroWrite-Paketes mitgeliefert.

| Viewer    | Funktion                                                                                     |
|-----------|----------------------------------------------------------------------------------------------|
| hex_view  | Zeigt eine Datei als Hexdump an. Dies ist nützlich bei binären Dateien wie z. B. Programmen. |
| iff_view  | Zeigt ein Bild im Format IFF-ILBM an.                                                        |
| play8svx  | Spielt ein Sound-Sample direkt von der Festplatte ab.                                        |
| playanim  | Zeigt eine Animation an.                                                                     |
| playmod   | Spielt ein ProTracker-Modul ab.                                                              |
| text_view | Zeigt eine Textdatei an.                                                                     |

Alle Viewer können auch aus dem CLI aufgerufen werden und erkennen einen Dateinamen als Argument. Das Utility »playanim« akzeptiert ein zweites Argument, das die Abspielgeschwindigkeit angibt.

## 1.41 Berechnungsformeln verwenden

Um in MicroBase Formeln berechnen zu können, müssen folgende Voraussetzungen

erfüllt sein:

- ARexx muss korrekt installiert sein
- Die »rexxmathlib.library« muss sich im Verzeichnis LIBS: befinden
- Der ARexx-Interpreter »RexxMast« muss aktiv sein
- Die Datei »MicroBase.CVD« im Programmverzeichnis muss - falls sie vorhanden ist - korrekte Formeln enthalten.

Lesen Sie bitte folgende Abschnitte, um mehr über die Formelberechnung in MicroBase zu erfahren:

Einführung in ARexx

Zugriff auf Felder und Konstanten

Die »rexxmathlib.library«

Wenn Sie komplexere Berechnungen, die Möglichkeit zur Veränderung nicht-

numerischer Daten oder flexiblere Abfragemöglichkeiten benötigen, sollten Sie die Verwendung von  
 ARexx-Skripten  
 erwägen.

#### Hinweis

Berechnungsformeln der Version 2.X müssen angepasst werden, da die Variablen-  
 namen für die Datenfelder geändert wurden.

## 1.42 Einführung in ARexx

In diesem Abschnitt wird nur die Formelsyntax von ARexx erläutert; die  
 anderen Elemente von ARexx werden nicht behandelt. Für weiterreichende  
 Informationen sollten Sie das mit Ihrem Computer gelieferte ARexx-Handbuch  
 lesen.

Tip: Einzelne ARexx-Anweisungen können durch Semikola (;) getrennt in  
 eine Zeile geschrieben werden.

#### Zahlenformate

Dezimale Zahlen werden in ARexx wie in MicroBase eingegeben. Bei binären  
 und hexadezimalen Zahlen ergeben sich jedoch gewisse Unterschiede:

| Format      | Beispiele       |
|-------------|-----------------|
| dezimal     | 1.23, -3e7      |
| binär       | '10110010'B     |
| hexadezimal | '2A'X, '12 E7'X |

#### Operatoren

Die Operatoren sind in der folgenden Übersicht absteigend nach ihrer Priori-  
 tät geordnet. Die Berechnungsreihenfolge kann durch die Benutzung von runden  
 Klammern geändert werden.

| Operator | Beispiel | Ergebnis | Bezeichnung       |
|----------|----------|----------|-------------------|
| +        | + "2.09" | 2.09     | Vorzeichen        |
| -        | - "2.09" | -2.09    | Vorzeichen        |
| **       | 2 ** 3   | 8        | Potenzierung      |
| *        | 2 * 3    | 6        | Multiplikation    |
| /        | 4 / 2    | 2        | Division          |
| %        | 5 % 2    | 2        | Ganzzahl-Division |
| //       | 5 // 2   | 1        | Rest (Modulo)     |
| +        | 2 + 3    | 5        | Addition          |
| -        | 2 - 3    | -1       | Subtraktion       |

## Integrierte Funktionen

Die folgenden in ARexx integrierten numerischen Funktionen sind für MicroBase sinnvoll:

| Funktion      | Beispiel    | Ergebnis | Bezeichnung                                                                                                                                                                                                             |
|---------------|-------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ABS(X)        | ABS(-2)     | 2        | Berechnet den Absolutwert einer Zahl                                                                                                                                                                                    |
| HASH(X)       | HASH('1')   | 49       | Berechnet den Hash-Wert einer Zeichenfolge                                                                                                                                                                              |
| MAX(A,B,..)   | MAX(1,2)    | 2        | Berechnet den Maximalwert einiger Werte                                                                                                                                                                                 |
| MIN(A,B,..)   | MIN(1,2)    | 1        | Berechnet den Minimalwert                                                                                                                                                                                               |
| RANDOM(U,O,A) | RANDOM(1,6) | 4???     | Berechnet eine Zufallszahl zwischen U und O. Das Intervall von U bis O darf nicht größer als 1000 sein. Werden die Werte nicht angegeben, nimmt ARexx 0 und 999. Sie können optional noch einen Ausgangswert A angeben. |
| RANDU()       | RANDU()     | 0.104??? | Berechnet eine Zufallszahl zwischen 0 und eins.                                                                                                                                                                         |
| SIGN(X)       | SIGN(-2.34) | -1       | Berechnet das Vorzeichen einer Zahl.                                                                                                                                                                                    |
| TRUNCX,S)     | TRUNC(1.23) | 1        | Kürzt die Nachkommastellen. Geben Sie die Anzahl der Nachkommastellen S nicht an, werden alle Ziffern nach dem Dezimalpunkt entfernt.                                                                                   |

## Beispiele für Formeln

```
ABS(-2) ** 3
'0E 2A B9'X // 7
```

## 1.43 Zugriff auf Felder und Konstanten

### Zugriff auf Feldinhalte

Innerhalb von Formeln können Sie auf die Felder des aktuellen Datensatzes zugreifen, indem Sie die Variablen field1 bis field234 einsetzen.

### Zugriff auf Konstanten

Zusätzlich können Sie vordefinierte Konstanten benutzen, die in jedem Datensatz den gleichen Wert haben. Die Namen der Konstanten beginnen in MicroBase mit dem Zeichen '#' und können sonst aus beliebigen Buchstaben bestehen. Lerzeichen sind - anders als in Feldnamen - allerdings nicht erlaubt.

### Definition der Konstanten

Die Konstanten sind in der Datei »MicroBase.CVD« im Programmverzeichnis definiert. Diese Datei können Sie mit einem beliebigen Texteditor ändern bzw. ergänzen.

Für jede definierte Konstante muss in der Datei eine Zeile vorhanden sein, die dieser einen Wert zuweist. Eine Zeile könnte beispielsweise

```
#VoyagerMaximumWarp = 9.975
```

lauten. Die Syntax von ARexx ist genau einzuhalten.

Die vordefinierten Konstanten

Standardmäßig enthält die Datei »MicroBase.CVD« folgende Einträge:

```
/* MICROBASE.CVD */
NUMERIC DIGITS 14
```

```
#pi = 3.141592653589793
#e = 2.718281828459045
```

Die mathematischen Konstanten  $\pi$  und  $e$  werden definiert und die maximale Rechengenauigkeit von ARexx aktiviert.

Beispiel

Es wird angenommen, dass die Datenbank die beiden Felder "Bruttopreis" und "Nettopreis" enthält. Die Buchhaltung nennt uns für jeden Artikel den Nettopreis; der Bruttopreis soll ermittelt werden.

In diesem Fall ist es sinnvoll, dass "Bruttopreis" ein Formelfeld ist. Die Mehrwertsteuer ändert sich in der nächsten Zeit nicht, sie kann also als Konstante definiert werden.

Folgende Schritte sind erforderlich:

1 Fügen Sie der oben erwähnten Datei »MicroBase.CVD« folgende Zeilen hinzu:

```
#MWSTermaessigt = 7
#MWST = #MWSTermaessigt * 2 + 1
```

Dies ist ein Beispiel für eine Formel in der Konstantendefinitionsdatei, die sich auf einen zuvor definierten Wert bezieht. Es wäre aber effizienter gewesen, gleich die Zahl 15 einzusetzen.

2 Erstellen Sie die Formel für das Feld "Bruttopreis" (es wird angenommen, dass "Nettopreis" das vierte Feld ist):

```
&004 * #MWST
```

Wenn Sie jetzt einen Nettopreis eingeben, wird automatisch der Bruttowert errechnet.

## 1.44 Verschlüsselte Datenbankdateien des alten Formats konvertieren

Falls Ihre Datenbankdatei mit einer älteren Programmversion als 2.1.4 verschlüsselt wurde, müssen Sie sie in das neue Format konvertieren lassen.

Das neue Format bietet folgende Vorteile:

- erhöhte Datensicherheit durch minimierte Fehleranfälligkeit
- erheblich verbesserter Verschlüsselungsalgorithmus, der Ihre sensiblen Daten noch besser schützt

(In den Versionen 2.1.3 und 2.1.4 wurde schrittweise das neue Format eingeführt. MicroBase 2.1.3 unterstützte den neuen Verschlüsselungsalgorithmus jedoch nicht korrekt.)

Gehen Sie wie folgt vor:

- 1 Öffnen Sie die verschlüsselte Datei mit Ihrem \*alten\* MicroBase.
- 2 Exportieren Sie alle Datensätze in eine sequentielle Datei.
- 3 Verlassen Sie MicroBase.
- 4 Öffnen Sie die Datenbankdatei mit dem \*neuen\* MicroBase. Sie können nun ggf. ein neues Passwort verwenden.
- 5 Importieren Sie die soeben erstellte sequentielle Datei ab dem Datensatz '1', ohne den ersten Datensatz zu überlesen.
- 6 Überprüfen Sie die Inhalte der Datensätze auf eventuelle Fehler, die während des Ex- oder Importierens aufgetreten sein könnten.

Sie können die Datenbankdatei nun wie zuvor verwenden.

## 1.45 Die mitgelieferten Datenbanken

Die folgenden Datenbanken werden zu Demonstrationszwecken mitgeliefert. Sie sollten sich die Datenbankstrukturen ansehen; vielleicht bekommen Sie so Anregungen für eigene Realisierungen. Besonders die Feldoptionen und die Vorgaben könnten hilfreich sein.

Bilder/Tutor

Diese Datenbank stellt einen Bilderkatalog dar und wird während des Tutoriums erstellt.

Zu den einzelnen Bildern werden jeweils eine kurze Bezeichnung sowie der Dateipfad gespeichert. Die Bilder können durch Druck auf die Taste CTRL-B auf dem Bildschirm angezeigt werden.

Die zugehörige Listendatei erzeugt einfach eine Liste der Bilder inkl. Dateipfad.

Datenbanken/Beispiel

"Beispiel.DAT" ist eine Adressdatenbank, die die für die Textverarbeitung

---

MicroWrite obligatorischen Datenfelder enthält.

Die Listendatei zeigt, wie einfach Serienbriefe in MicroBase erstellt werden können. Das Layout ist für Fensterumschläge entwickelt worden.

Das zugehörige Skript erstellt die Inhalte der Felder "Briefanrede" und "Adressanrede".

Das Passwort ist "Beispiel".

#### Datenbanken/Textprogramme

Hier sehen Sie die Zusammenfassung eines Vergleichs verschiedener Textverarbeitungsprogramme unter verschiedenen Betriebssystemen.

Die Bewertungskriterien sowie die komplette Untersuchung finden Sie in der mitgelieferten Grafik und der Tabelle im Dokument »Textprogramme.TBL«.

In »Textprogramme.DOC« wird versucht, einige Vor- und Nachteile der Amiga- bzw. PC-Textverarbeitungsprogramme herauszuarbeiten.

#### Datenbanken/Videos

Dies ist ein Extrakt aus einer existierenden Videokassettendatenbank (die komplette Datenbank hat über 570 Datensätze).

Die Beispiellistendatei erstellt eine Videokassettenliste mit folgenden Angaben: Filmtitel, Kassettennummer, Filmnummer und Filmlänge.

Diese Datenbankdatei finden Sie auch im dBASE-Verzeichnis als exportierte dBASE-Datenbankdatei.

Die Dateien »Videos\_SB4.sbf« und »Videos\_SB4.sbd« enthalten die gleichen Informationen, wurden jedoch mittels des Utilities »dat2dbf« in eine dBASE-Datei exportiert und dann in SuperBase Professional IV 1.01 importiert. Mit diesen Dateien kann der Import von SuperBase-Datenbanken getestet werden.

#### Datenbanken/Programme

»Programme.DAT« enthält einen Ausschnitt aus einem existierenden Programm-katalog. Die Original-Datenbank wurde mit Symantec F & A 4.0 für Windows entwickelt und über eine dBASE-Datei nach MicroBase konvertiert.

Die Originaldatenbank hat über 450 Datensätze.

#### Datenbanken/Artikel

In dieser Datenbank wurden einige Artikel erfasst. Für eine reale Artikel-datenbank ist sie natürlich unzureichend; sie soll jedoch einfach nur folgende Eigenschaften zeigen:

- die Artikelnummer wird von MicroBase automatisch hochgezählt
-

- negative Bestände sind nicht möglich
- der Nettopreis wird aus den Selbstkosten errechnet, indem ein Gewinnzuschlag addiert wird

#### Datenbanken/Personal und Datenbanken/Lohn

Die Lohndatenbank enthält nur Angaben, die die Lohnbuchhaltung interessieren: Name, Lohn und Personalnummer.

Die Personaldatenbank enthält neben der Personalnummer die Adresse des Mitarbeiters und sein Geburtsdatum.

Das Interessante an diesen Datenbankdateien ist die eingerichtete Datenbankverknüpfung (Relation). Hierfür wird die Personalnummer als Schlüssel­feld benutzt, da sie in beiden Dateien erfasst ist. Das Geburtsdatum ist in der Lohndatenbank nicht enthalten, sondern wird aus der Personaldatenbank geholt!

#### Datenbanken/Aufträge

Die Datenbank enthält Kundenaufträge. Sie ist mit den Datenbanken "Waren" und "Kunden" verbunden, um eine Mehrfachspeicherung zu vermeiden.

#### Datenbanken/Programmiersprachen

Diese Datenbankdatei enthält eine Übersicht über die wichtigsten Programmiersprachen (wegen der Einschränkungen der Demo-Version von MicroBase sind nur 20 Einträge möglich).

Zu jeder Programmiersprache sind jeweils die Programmlänge (in Bytes) und die Laufzeit (in Ticks = 1/50 Sekunde) für ein Programm angegeben, das von 1 bis 1 000 000 zählt. Folgende Einstellungen wurden vorgenommen:

- Variablen des Typs "lange Ganzzahl" (Long Integer) wurden benutzt
- Fehlerüberwachungen wurden deaktiviert
- Optimierungen wurden aktiviert
- Falls erforderlich, wurde der Prozessortyp "Motorola 68000" gewählt

Eine Erläuterung der Ergebnisse finden Sie in »Programmiersprachen.DOC« im gleichen Verzeichnis.

»Sprachen.TCD« enthält eine etwas ältere Version der Untersuchung im Format von TurboCalc 3.0.

## 1.46 Relationen definieren

Eine Relation ist eine Verknüpfung mit einer anderen Datenbankdatei. Der Inhalt des Relationsfeldes ist zwar in der Maske der aktuellen Datenbank

zu sehen, ist aber nicht in dieser Datei gespeichert. Die Vorteile von Relationen sind also:

- Verhinderung von Redundanz: Daten werden nur einmal gespeichert
- Übersichtlichere, kürzere Datenbankdateien
- Verminderte Fehleranfälligkeit
- Verbesserte(r) Datenschutz / Datensicherheit
- Zugriff nur auf bestimmte Felder der verbundenen Datenbank; die anderen bleiben "unsichtbar" für Unbefugte

Folgendes ist bei MicroBase zu beachten:

- Die zweite Datenbankdatei muss dasselbe Passwort haben wie die aktive Datenbank.
- MicroBase überprüft die referenzielle Integrität der Datenbankdatei, sobald ihr Inhalt verändert wird (z. B. beim Eingeben, Aktualisieren oder Importieren). Dies gilt jedoch nur für den Fall, dass die Option "strikte Referenzintegrität" aktiviert ist. Sonst erscheint lediglich eine Fehlermeldung, falls eine Relation nicht aufgebaut werden konnte.
- Relationen werden nur über eine Ebene unterstützt; das Zieldatenfeld in der zweiten Datenbankdatei darf nicht selbst ein Relationsfeld sein. Dies gilt entsprechend für die Schlüsselfelder.

#### Definition von Relationen

Zur Erstellung einer Relation während der Datenbankdefinition gehen Sie folgendermaßen vor:

- 1 Geben Sie einen Feldnamen ein.
  - 2 Geben Sie den Feldtyp 'R' (für "Relation") an.
  - 3 Die Länge des Datenfeldes wird automatisch auf 1 (eins) gesetzt. Dies ist für eventuelle Erweiterungen in zukünftigen Programmversionen vorgesehen.
  - 4 Geben Sie nun die für die Relation erforderlichen Daten ein.
    - a Hauptschlüssel dieser Datenbank: Dies ist das Feld, das den Suchbegriff enthält.
    - b Hauptschlüssel der 2. Datenbank: In diesem Feld wird nach einer Übereinstimmung gesucht. Der erste passende Datensatz wird mit der ersten Datenbank verknüpft.
    - c, d Nebenschlüssel: Falls der Hauptschlüssel nicht eindeutig ist, können Sie einen Nebenschlüssel definieren. Dieser wird von MicroBase nur dann herangezogen, wenn beim Hauptschlü-
-



sel eine Übereinstimmung gefunden wurde. Die Vorgabe 0 (null) deaktiviert den Nebenschlüssel.

- e Datenfeld der zweiten Datenbank: Der Inhalt dieses Datenfeldes der zweiten Datenbank erscheint in der Maske der neu zu erstellenden (aktuellen) Datenbankdatei.
- f Länge des Datenfeldes: Mit dieser Länge wird das Feld in der Maske angezeigt.

- 5 Überspringen Sie die Vorgaben und Optionen; sie werden in diesem Fall nicht ausgewertet.
- 6 Wenn das Datenfeld numerische Daten enthält, können Sie ein Zahlenformat definieren.
- 7 Das Programm erstellt nun die Datenbank. Die Verknüpfung besteht jedoch noch nicht; sie wird erst bei der Bearbeitung der Datensätze hergestellt.

Die Definition der Relation kann nachträglich geändert werden, indem die Struktur der Datenbank ediert wird.

#### Arbeiten mit Relationen

Das Arbeiten mit durch Relationen verknüpften Datenbanken ist sehr einfach, da es sich nicht von der Vorgehensweise bei "normalen" Datenbanken unterscheidet.

Wenn Sie Datensätze neu eingeben, sind die Relationsfelder leer, da sie ihren Inhalt erst beim Zugriff auf den Datensatz erhalten. Sie werden auf ihre Gültigkeit geprüft, wenn Sie die "strikte Referenzintegrität" aktiviert haben. Dies ist vorgabemäßig der Fall.

Beim Ansehen der Datensätze werden die Relationsfelder wie alle anderen Datenfelder angezeigt. Sie werden lediglich kursiv dargestellt, sofern Sie die Optionen nicht geändert haben. Zusätzlich können Sie wählen, ob die Relationen überhaupt angezeigt werden.

Beim Suchen, Drucken usw. verhalten sich die Relationsfelder so wie alle anderen. Zusätzlich können Zahlenformate definiert werden, falls das Datenfeld numerische Daten enthält.

#### Caches für Relationen

Für Relationsfelder werden automatisch Caches eingerichtet. Diese Cachedateien funktionieren wie folgt:

- a Wenn Sie Daten eingeben, erstellt MicroBase einen leeren Cacheeintrag.
- b Bei der Datenansicht verknüpft MicroBase die Datenbanken. Dazu sieht es zuerst in der Cache-Datei nach. Falls der gefundene Eintrag zu keinem Ergebnis führt, durchsucht das Programm die verknüpfte Datenbank, bis ein passender Datensatz gefunden wurde und aktualisiert dann den Cache.

- c Beim nächsten Zugriff auf den Datensatz ist der Eintrag im Cache korrekt und MicroBase findet den passenden Datensatz sofort.

Sie sollten die Caches regelmäßig aktualisieren lassen, um eine maximale Verarbeitungsgeschwindigkeit zu erreichen.

Beim Löschen und Sortieren von Datensätzen verlieren die Relationscaches ihre Gültigkeit, wenn Sie die Option "strikte Referenzintegrität" deaktiviert haben.

#### Format der Relationsdefinitionen

Die Relationsdefinitionen werden jeweils in einer Datei mit der Dateinamenserweiterung ".REL" abgelegt. Eine Zeile in einer solchen Datei könnte z. B. so aussehen:

```
"Videofilme.DAT",1,3,10,50,0,0
```

Die einzelnen Einträge haben folgende Bedeutung:

```
"Videofilme.DAT": Name der zweiten Datenbankdatei, aus der die Informationen
gelesen werden sollen (muss immer in doppelten
Anführungszeichen stehen, da in Dateinamen Leerzeichen
auftreten können)
1: Hauptschlüsselfeld der aktuellen Datenbankdatei
3: Hauptschlüsselfeld der zweiten Datenbankdatei
10: Datenfeld in der zweiten Datenbankdatei, das die
gewünschten Informationen enthält
50: Anzuzeigende Länge des Datenfeldes
0: (Kein Nebenschlüssel in dieser Datenbank definiert)
0: (Kein Nebenschlüssel in der verknüpften Datenbank
definiert)
```

## 1.47 History

| Version | Bemerkungen                                                                                                                                                                                                                                                                                                        |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.XX    | Aufbau des Programms (nicht veröffentlicht): nur Textfelder mit bis zu 53 Zeichen                                                                                                                                                                                                                                  |
| 2.03    | Unterstützung von Feldern mit mehr als 53 Zeichen; Einführung von Felddatentypen; Berechnungen; dBASE-Importmodule; Verschlüsselung; externe Datenfelder (z. B. für Memos und Datatypes); Standardwerte für Felder; automatische Generierung von Seriennummern (* erste veröffentlichte Version *)                 |
| *       |                                                                                                                                                                                                                                                                                                                    |
| 2.13    | Datentyp "Logisch"; neue Verschlüsselungsroutine; Unterstützung von Konstanten bei Berechnungen; automatische Reparatur von zerstörten Datenbanken; verbesserte Routine zum Sortieren von Zeit- und Datumsfeldern; Optionen für Felder (z. B. Umwandlung in Großbuchstaben); Viewer für Workbench 2.X mitgeliefert |

(\* zweite veröffentlichte Version \*)

- 2.14 Datentyp "Relation" zur Verknüpfung von Datenbanken; Verbesserung der Reparaturmöglichkeiten; Verbesserte Sortieroutine; Einstellungen für Felder werden jetzt beim Schließen der Datenbank korrekt gelöscht
  - 2.15 Relationen werden jetzt durch Caches beschleunigt; eindeutigere Verknüpfungen; verbesserte Verwaltung von Selektionen, dadurch drastischer Geschwindigkeitszuwachs im 'virtuellen Modus'
  - 2.16 Neuer Menüpunkt "Caches aktualisieren" baut Relationscaches automatisch auf; Feldoption "Namen" arbeitet jetzt korrekt; Fehler beim Schließen verknüpfter Datenbanken behoben
  - 2.17 Fehler beim Abbruch der Eingabe neuer Datensätze beseitigt. Über einen neuen Menüpunkt können jetzt selektierte Datensätze gelöscht werden. Überlange Dateien werden nicht mehr sofort reorganisiert.
  - 2.18 Einführung der Sperrung von Datenbanken und -sätzen für den Einsatz von MicroBase in Netzwerken.
  - 2.19 Verbesserte Aktualisierung von Caches; Deaktivierung des 'virtuellen Modus', wenn Datenbankdatei schon benutzt wird.
  - 2.20 Einführung von ARexx-Skripten. Es erfolgt keine Fehlermeldung mehr, wenn ein in Bearbeitung befindlicher Datensatz noch einmal ediert wird. (Es erfolgte eine Meldung, dass der Datensatz gesperrt sei.) Der 'virtuelle Modus' wird jetzt schon bei der nächsten Datenbankdatei (de)aktiviert; ein Neustart des Programms ist nicht mehr notwendig. (\* dritte veröffentlichte Version \*)
  - 2.2.1 Schnittstelle zum Diagrammgenerator "MicroChart" hinzugefügt.
  - 2.2.2 Die Sperrung von Datensätzen kann jetzt deaktiviert werden. Der Import von sequentiellen Dateien wurde intern verbessert. Beim Lesen der Datensätze wird ein neues, optimiertes Verfahren verwendet. Die Handhabung von Zahlen- und Datumsfeldern wurde verbessert.
  - 2.2.3 Beim Erstellen neuer Datenbankdateien werden bestehende Dateien nicht mehr überschrieben. MicroBase erkennt jetzt fehlerhafte Passwörter direkt beim Öffnen einer Datenbankdatei. Aus Kompatibilitätsgründen werden nun Systemdateien erzeugt, die Angaben über die zur Erstellung der Datenbank verwendete Programmversion enthalten. Ein Lernprogramm erläutert das Tutorial visuell.
  - 2.2.4 Die Referenzintegrität wird jetzt ggf. gleich bei der Eingabe geprüft. Es kann endlich auch absteigend sortiert werden. Die Unterstützung von Zahlenformaten in Listen wurde eingeführt. Beim Öffnen einer Datenbankdatei wird diese erzeugt, wenn sie nicht existiert. Zeigerdateien (.PTR) werden gleichfalls automatisch erzeugt.
  - 2.2.5 Phonetisches Suchen und Sortieren wurde eingeführt. MicroBase kann Feldinhalte nun auch "partiell" ersetzen. Der Suchvorgang wurde durch das Ignorieren leerer Suchbegriffe beschleunigt. Die dBASE-Module wurden korrigiert; dBASE-Datenbankdateien können jetzt auch
-

exportiert werden.

- 2.2.6 Optimierung und Beschleunigung des Datenbankzugriffs. Vorbereitung auf die Anhebung der maximalen Feldanzahl. Das Suchen und Sortieren ist nun viel schneller als zuvor.
- 3.0.0 Anhebung der maximalen Feldanzahl auf 234 statt 25. (Dazu wurden alle Listen-, Berechnungs- und Skriptsteuerzeichen usw. geändert!) Die dBASE-Module wurden angepasst. Berechnungen werden jetzt im aktuellen Verzeichnis ausgeführt. Sequentielle Dateien werden nun in die korrekten Datensätze kopiert und können in neue Datenbankdateien (statt in die aktuelle) importiert werden. Ein Listengenerator wird mitgeliefert. SuperBase-Datenbanken können importiert werden.
- 3.0.1 Die Bildschirmmaske zeigt die aktuelle Seite an. Die in Version 2.2.5 eingeführte Routine zur Umwandlung in Großbuchstaben wurde korrigiert. Auch Definitionsdateien werden beim Öffnen automatisch repariert. Die Datensicherheit wurde nochmals erhöht.
- 3.0.2 Die Ausführung von Berechnungen erfolgt jetzt ohne Fenster und schneller. Die Fenster für Viewer und Skripte wurden verkleinert. Das Sortieren von Datenbankdateien wurde beschleunigt. Der Kompatibilitätsmodus verlangsamt jetzt das Sortieren von Datenbankdateien. Der Zugriff auf verknüpfte Datenbankdateien ist ebenfalls schneller als zuvor.

(\* vierte veröffentlichte Version \*)

## 1.48 MicroBase im Mehrbenutzerbetrieb

Seit Version 2.1.8 unterstützt MicroBase die Bearbeitung einer Datenbankdatei durch mehrere Benutzer zur gleichen Zeit. Dazu wird eine spezielle Datei verwendet, in der die zur Zeit in Bearbeitung befindlichen Datensätze vermerkt sind.

In MicroBase existieren folgende Varianten der Zugriffsverweigerung:

### 1. Sperrung des 'virtuellen Modus'

Die Verwaltungsdatei für den 'virtuellen Modus' kann für eine Datenbankdatei nur einmal erzeugt werden. Dies bedeutet, dass die anderen Benutzer zwangsläufig den Standardmodus benutzen müssen. (Dies ist auch die Reaktion des Programms auf diese Situation.)

### 2. Sperrung einer Datenbankdatei

Falls sich eine Operation auf die gesamte Datenbank bezieht, wird sie auch vollständig gesperrt. Dies bedeutet, dass kein weiterer Schreibzugriff auf die vorhandenen Datensätze zulässig ist, bis die Datenbanksperre aufgehoben ist.

Die gesamte Datenbank wird z. B. beim Aktualisieren, Sortieren und Löschen

von Datensätzen gesperrt.

### 3. Sperrung eines Datensatzes

MicroBase sperrt Datensätze, die zur Zeit ediert werden. So kann nur jeweils ein Benutzer den Inhalt eines Datensatzes ändern.

## 1.49 Die Verwendung von ARexx-Skripten

ARexx-Skripte sind ARexx-Programme, die den Inhalt von Datensätzen ändern können. Für sie gilt deshalb grundsätzlich erst einmal dasselbe wie für Berechnungen.

Im Vergleich zu Berechnungsformeln bestehen folgende Unterschiede:

- Sie sind nicht auf ein einziges Datenfeld beschränkt
- Sie können auch den Inhalt von nicht numerischen Datenfeldern ändern
- Es können komplexere Formeln berechnet werden - auch über mehrere Zeilen hinweg
- Es sind flexible Abfragemöglichkeiten vorhanden
- Es besteht die Möglichkeit, eigene Variablen zu definieren und diese für den nächsten Programmablauf abzulegen
- Sie können eigene Prozeduren und Funktionen definieren

Beispiel einer Problemstellung

Ablauf einer Skriptausführung

Übergebene Variablen

Ein Beispiel für ein Skript

Anwendungsbeispiele

Die Entwicklung eines ARexx-Skriptes

Die Erstellung interaktiver Skripte

Für eine Übersicht über die ARexx-Funktionen für die Verarbeitung von Zeichenketten (Strings) konsultieren Sie bitte Ihr mit dem Amiga geliefertes Handbuch zur Programmiersprache ARexx.

Hinweis

Skripte der Versionen 2.X müssen angepasst werden, da sich die Variablen für die Feldinhalte geändert haben.

## 1.50 Beispiel einer Problemstellung

Angenommen, Sie arbeiten in einer Lohn- und Gehaltsbuchhaltung und sollen die neuesten Tarifabschlüsse in die Tat umsetzen. Die Löhne ändern sich wie folgt:

| alter Lohn     | Lohnänderung |
|----------------|--------------|
| < 900 DM       | + 150 DM     |
| 900 – 3 160 DM | + 1.7 %      |
| sonst          | + 1.3 %      |

Wie Sie sehen, kann eine Berechnungsformel das Problem nicht lösen.

Wenn die Höhe des Lohnes im 3. Feld der Datenbank steht, könnte das Skript wie folgt lauten:

```
IF ( C < 900 ) THEN C = C + 150
ELSE DO
  IF ( ( C >= 900 ) & ( C < 3161 ) ) THEN C = C * 1.017
  ELSE C = C * 1.013
END
```

Bei komplexen Problemen können Skripte also sehr nützlich sein.

## 1.51 Ablauf einer Skriptausführung

Während der Ausführung besteht Ihr Skript aus folgenden Teilen:

- der Datei »MicroBase.CVD« aus dem Programmverzeichnis. Sie enthält die globalen Konstanten.
- der von MicroBase eingefügte Programmcode übergibt die Feldinhalte und einige Systemwerte.
- Ihrem Skript. Dieses muss die Feldinhalte im Anschluss an deren Veränderung in eine Datei schreiben, deren Name in der Variablen »mroname« steht.

Alle drei Teile können zusammen maximal so lang wie ein Datensatz werden – also etwa 70 KB. Dies gilt entsprechend für die Ausgabedatei. Alle Dateien werden in dem Verzeichnis erzeugt, in dem sich das Skript befindet.

Diese Phasen werden für jeden einzelnen Datensatz durchlaufen. Vor dem Ablauf eines Skripts entsteht also eine kurze Verzögerung, in der MicroBase das Skript vervollständigt.

Hinweis: Sie können den Ablauf von Skripten beschleunigen, indem Sie sie auf die resetfeste RAM-Disk ("RAD:") kopieren.

## 1.52 Übergebene Variablen

MicroBase übergibt Ihrem Skript folgende Variablen:

| Variable | Typ     | Inhalt                                                                                                                                                   |
|----------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| fieldnnn | Zeichen | Dies sind die Feldinhalte (nnn = Zahl von 1 bis 234)                                                                                                     |
| mroname  | Zeichen | Dies ist der Name der von Ihrem Skript zu erstellenden Ausgabedatei. Sie muss die Inhalte aller Felder in der korrekten Reihenfolge (A bis Y) enthalten. |
| recno    | Zahl    | Nummer des aktuellen Datensatzes.                                                                                                                        |
| reccount | Zahl    | Anzahl der Datensätze.                                                                                                                                   |
| recmax   | Zahl    | Maximale Anzahl der Datensätze.                                                                                                                          |
| recsel   | Zahl    | Anzahl der selektierten Datensätze.                                                                                                                      |
| dbname   | Zeichen | Name der Datenbankdatei.                                                                                                                                 |
| idxname  | Zeichen | Name der Selektionsmatrix.                                                                                                                               |
| mrername | Zeichen | Name des ARexx-Skriptes.                                                                                                                                 |
| fields   | Zahl    | Anzahl der Datenfelder.                                                                                                                                  |

Die Werte der Datenfelder sowie von Variablen des Typs "Zeichen" werden dabei in doppelte Anführungszeichen (") gesetzt.

## 1.53 Ein Beispiel für ein Skript

Zu der Datenbankdatei »Beispiel.DAT« gehört folgendes Skript, das die Inhalte der Felder "Briefanrede" und "Adressanrede" erstellt.

Das Feld "Anrede" enthält z. B. den Eintrag 'er Herr', in einer Adresse muss jedoch 'Herrn' stehen. Die betroffenen Felder sind F, G und H.

Das Skript sieht wie folgt aus:

```
/* Beispiel.MRX */
NUMERIC DIGITS 14
SIGNAL ON ERROR
SIGNAL ON SYNTAX
```

```
/* Fügen Sie bitte hier Ihre Programmzeilen ein! */
```

```
IF field6 = "er Herr" THEN DO
  field7 = "Herr"
  field8 = "Herrn"
END
ELSE DO
  hlp1 = INDEX(field6, " ")
  field7 = SUBSTR(field6, hlp1 + 1)
  field8 = SUBSTR(field6, hlp1 + 1)
  DROP hlp1
END
```

```
/* Zurückschreiben der Feldinhalte */
```

```
CALL OPEN(out,mroname,"W")
CALL WRITELN(out, field1)
CALL WRITELN(out, field2)
CALL WRITELN(out, field3)
CALL WRITELN(out, field4)
CALL WRITELN(out, field5)
CALL WRITELN(out, field6)
CALL WRITELN(out, field7)
CALL WRITELN(out, field8)
CALL WRITELN(out, field9)
CALL WRITELN(out, field10)
CALL WRITELN(out, field11)
CALL WRITELN(out, field12)
CALL WRITELN(out, field13)
CALL WRITELN(out, field14)
CALL WRITELN(out, field15)
CALL WRITELN(out, field16)
CALL WRITELN(out, field17)
CALL WRITELN(out, field18)
CALL WRITELN(out, field19)
CALL WRITELN(out, field20)
CALL WRITELN(out, field21)
CALL WRITELN(out, field22)
CALL WRITELN(out, field23)
CALL WRITELN(out, field24)
CALL WRITELN(out, field25)
CALL WRITELN(out, field26)
CALL WRITELN(out, field27)
CALL WRITELN(out, field28)
CALL WRITELN(out, field29)
CALL WRITELN(out, field30)
CALL WRITELN(out, field31)
CALL WRITELN(out, field32)
CALL WRITELN(out, field33)
CALL WRITELN(out, field34)
CALL WRITELN(out, field35)
CALL WRITELN(out, field36)
CALL WRITELN(out, field37)
CALL WRITELN(out, field38)
CALL WRITELN(out, field39)
CALL WRITELN(out, field40)
CALL WRITELN(out, field41)
```

---



---

CALL WRITELN(out, field42)  
CALL WRITELN(out, field43)  
CALL WRITELN(out, field44)  
CALL WRITELN(out, field45)  
CALL WRITELN(out, field46)  
CALL WRITELN(out, field47)  
CALL WRITELN(out, field48)  
CALL WRITELN(out, field49)  
CALL WRITELN(out, field50)  
CALL WRITELN(out, field51)  
CALL WRITELN(out, field52)  
CALL WRITELN(out, field53)  
CALL WRITELN(out, field54)  
CALL WRITELN(out, field55)  
CALL WRITELN(out, field56)  
CALL WRITELN(out, field57)  
CALL WRITELN(out, field58)  
CALL WRITELN(out, field59)  
CALL WRITELN(out, field60)  
CALL WRITELN(out, field61)  
CALL WRITELN(out, field62)  
CALL WRITELN(out, field63)  
CALL WRITELN(out, field64)  
CALL WRITELN(out, field65)  
CALL WRITELN(out, field66)  
CALL WRITELN(out, field67)  
CALL WRITELN(out, field68)  
CALL WRITELN(out, field69)  
CALL WRITELN(out, field70)  
CALL WRITELN(out, field71)  
CALL WRITELN(out, field72)  
CALL WRITELN(out, field73)  
CALL WRITELN(out, field74)  
CALL WRITELN(out, field75)  
CALL WRITELN(out, field76)  
CALL WRITELN(out, field77)  
CALL WRITELN(out, field78)  
CALL WRITELN(out, field79)  
CALL WRITELN(out, field80)  
CALL WRITELN(out, field81)  
CALL WRITELN(out, field82)  
CALL WRITELN(out, field83)  
CALL WRITELN(out, field84)  
CALL WRITELN(out, field85)  
CALL WRITELN(out, field86)  
CALL WRITELN(out, field87)  
CALL WRITELN(out, field88)  
CALL WRITELN(out, field89)  
CALL WRITELN(out, field90)  
CALL WRITELN(out, field91)  
CALL WRITELN(out, field92)  
CALL WRITELN(out, field93)  
CALL WRITELN(out, field94)  
CALL WRITELN(out, field95)  
CALL WRITELN(out, field96)  
CALL WRITELN(out, field97)  
CALL WRITELN(out, field98)

---

---

CALL WRITELN(out, field99)  
CALL WRITELN(out, field100)  
CALL WRITELN(out, field101)  
CALL WRITELN(out, field102)  
CALL WRITELN(out, field103)  
CALL WRITELN(out, field104)  
CALL WRITELN(out, field105)  
CALL WRITELN(out, field106)  
CALL WRITELN(out, field107)  
CALL WRITELN(out, field108)  
CALL WRITELN(out, field109)  
CALL WRITELN(out, field110)  
CALL WRITELN(out, field111)  
CALL WRITELN(out, field112)  
CALL WRITELN(out, field113)  
CALL WRITELN(out, field114)  
CALL WRITELN(out, field115)  
CALL WRITELN(out, field116)  
CALL WRITELN(out, field117)  
CALL WRITELN(out, field118)  
CALL WRITELN(out, field119)  
CALL WRITELN(out, field120)  
CALL WRITELN(out, field121)  
CALL WRITELN(out, field122)  
CALL WRITELN(out, field123)  
CALL WRITELN(out, field124)  
CALL WRITELN(out, field125)  
CALL WRITELN(out, field126)  
CALL WRITELN(out, field127)  
CALL WRITELN(out, field128)  
CALL WRITELN(out, field129)  
CALL WRITELN(out, field130)  
CALL WRITELN(out, field131)  
CALL WRITELN(out, field132)  
CALL WRITELN(out, field133)  
CALL WRITELN(out, field134)  
CALL WRITELN(out, field135)  
CALL WRITELN(out, field136)  
CALL WRITELN(out, field137)  
CALL WRITELN(out, field138)  
CALL WRITELN(out, field139)  
CALL WRITELN(out, field140)  
CALL WRITELN(out, field141)  
CALL WRITELN(out, field142)  
CALL WRITELN(out, field143)  
CALL WRITELN(out, field144)  
CALL WRITELN(out, field145)  
CALL WRITELN(out, field146)  
CALL WRITELN(out, field147)  
CALL WRITELN(out, field148)  
CALL WRITELN(out, field149)  
CALL WRITELN(out, field150)  
CALL WRITELN(out, field151)  
CALL WRITELN(out, field152)  
CALL WRITELN(out, field153)  
CALL WRITELN(out, field154)  
CALL WRITELN(out, field155)

---

---

CALL WRITELN(out, field156)  
CALL WRITELN(out, field157)  
CALL WRITELN(out, field158)  
CALL WRITELN(out, field159)  
CALL WRITELN(out, field160)  
CALL WRITELN(out, field161)  
CALL WRITELN(out, field162)  
CALL WRITELN(out, field163)  
CALL WRITELN(out, field164)  
CALL WRITELN(out, field165)  
CALL WRITELN(out, field166)  
CALL WRITELN(out, field167)  
CALL WRITELN(out, field168)  
CALL WRITELN(out, field169)  
CALL WRITELN(out, field170)  
CALL WRITELN(out, field171)  
CALL WRITELN(out, field172)  
CALL WRITELN(out, field173)  
CALL WRITELN(out, field174)  
CALL WRITELN(out, field175)  
CALL WRITELN(out, field176)  
CALL WRITELN(out, field177)  
CALL WRITELN(out, field178)  
CALL WRITELN(out, field179)  
CALL WRITELN(out, field180)  
CALL WRITELN(out, field181)  
CALL WRITELN(out, field182)  
CALL WRITELN(out, field183)  
CALL WRITELN(out, field184)  
CALL WRITELN(out, field185)  
CALL WRITELN(out, field186)  
CALL WRITELN(out, field187)  
CALL WRITELN(out, field188)  
CALL WRITELN(out, field189)  
CALL WRITELN(out, field190)  
CALL WRITELN(out, field191)  
CALL WRITELN(out, field192)  
CALL WRITELN(out, field193)  
CALL WRITELN(out, field194)  
CALL WRITELN(out, field195)  
CALL WRITELN(out, field196)  
CALL WRITELN(out, field197)  
CALL WRITELN(out, field198)  
CALL WRITELN(out, field199)  
CALL WRITELN(out, field200)  
CALL WRITELN(out, field201)  
CALL WRITELN(out, field202)  
CALL WRITELN(out, field203)  
CALL WRITELN(out, field204)  
CALL WRITELN(out, field205)  
CALL WRITELN(out, field206)  
CALL WRITELN(out, field207)  
CALL WRITELN(out, field208)  
CALL WRITELN(out, field209)  
CALL WRITELN(out, field210)  
CALL WRITELN(out, field211)  
CALL WRITELN(out, field212)

---

```

CALL WRITELN(out, field213)
CALL WRITELN(out, field214)
CALL WRITELN(out, field215)
CALL WRITELN(out, field216)
CALL WRITELN(out, field217)
CALL WRITELN(out, field218)
CALL WRITELN(out, field219)
CALL WRITELN(out, field220)
CALL WRITELN(out, field221)
CALL WRITELN(out, field222)
CALL WRITELN(out, field223)
CALL WRITELN(out, field224)
CALL WRITELN(out, field225)
CALL WRITELN(out, field226)
CALL WRITELN(out, field227)
CALL WRITELN(out, field228)
CALL WRITELN(out, field229)
CALL WRITELN(out, field230)
CALL WRITELN(out, field231)
CALL WRITELN(out, field232)
CALL WRITELN(out, field233)
CALL WRITELN(out, field234)
CALL CLOSE(out)

```

```
/* Programm beenden */
```

```

CALL REMLIB("rexxmathlib.library")
CALL REMLIB("rexxsupport.library")
EXIT

```

```
/* Fehlerbehandlung */
```

```

SYNTAX:
SAY "##SYNTAX"
EXIT 10

```

```

ERROR:
SAY "##ERROR"
EXIT 10

```

```
/* Fügen Sie hier ggfs. Ihre Funktionen und Prozeduren ein! */
```

Es wird während der Laufzeit wie folgt abgewandelt. In den durch '//' vom Programmcode getrennten Kommentaren finden Sie eine kurze Erläuterung.

```

/* RAM_0:Beispiel.rexx */
CALL ADDLIB("rexxsupport.library",0,-30,0)
CALL ADDLIB("rexxmathlib.library",0,-30,0)
/* MICROBASE.CVD */
NUMERIC DIGITS 14

#pi = 3.141592653589793
#e = 2.718281828459045
field1 = "Meier"
field2 = "Bettina"

```

//Kopfzeile des von Micro-  
//Base geänderten Skripts:  
//Öffnen der Bibliotheken  
  
//MicroBase.CVD wird  
//eingefügt (Konstanten)  
  
//Einschub: Übergabe der  
//Feldinhalte

```
field3 = "Meierdorfer Weg 12"  
field4 = "0"  
field5 = "12345 Wundorf"  
field6 = "es Fräulein"  
field7 = "Fräulein"  
field8 = "Fräulein"  
field9 = ""  
field10 = ""  
field11 = ""  
field12 = ""  
field13 = ""  
field14 = ""  
field15 = ""  
field16 = ""  
field17 = ""  
field18 = ""  
field19 = ""  
field20 = ""  
field21 = ""  
field22 = ""  
field23 = ""  
field24 = ""  
field25 = ""  
field26 = ""  
field27 = ""  
field28 = ""  
field29 = ""  
field30 = ""  
field31 = ""  
field32 = ""  
field33 = ""  
field34 = ""  
field35 = ""  
field36 = ""  
field37 = ""  
field38 = ""  
field39 = ""  
field40 = ""  
field41 = ""  
field42 = ""  
field43 = ""  
field44 = ""  
field45 = ""  
field46 = ""  
field47 = ""  
field48 = ""  
field49 = ""  
field50 = ""  
field51 = ""  
field52 = ""  
field53 = ""  
field54 = ""  
field55 = ""  
field56 = ""  
field57 = ""  
field58 = ""  
field59 = ""
```

---

---

```
field60 = ""
field61 = ""
field62 = ""
field63 = ""
field64 = ""
field65 = ""
field66 = ""
field67 = ""
field68 = ""
field69 = ""
field70 = ""
field71 = ""
field72 = ""
field73 = ""
field74 = ""
field75 = ""
field76 = ""
field77 = ""
field78 = ""
field79 = ""
field80 = ""
field81 = ""
field82 = ""
field83 = ""
field84 = ""
field85 = ""
field86 = ""
field87 = ""
field88 = ""
field89 = ""
field90 = ""
field91 = ""
field92 = ""
field93 = ""
field94 = ""
field95 = ""
field96 = ""
field97 = ""
field98 = ""
field99 = ""
field100 = ""
field101 = ""
field102 = ""
field103 = ""
field104 = ""
field105 = ""
field106 = ""
field107 = ""
field108 = ""
field109 = ""
field110 = ""
field111 = ""
field112 = ""
field113 = ""
field114 = ""
field115 = ""
field116 = ""
```

---

---

```
field117 = ""
field118 = ""
field119 = ""
field120 = ""
field121 = ""
field122 = ""
field123 = ""
field124 = ""
field125 = ""
field126 = ""
field127 = ""
field128 = ""
field129 = ""
field130 = ""
field131 = ""
field132 = ""
field133 = ""
field134 = ""
field135 = ""
field136 = ""
field137 = ""
field138 = ""
field139 = ""
field140 = ""
field141 = ""
field142 = ""
field143 = ""
field144 = ""
field145 = ""
field146 = ""
field147 = ""
field148 = ""
field149 = ""
field150 = ""
field151 = ""
field152 = ""
field153 = ""
field154 = ""
field155 = ""
field156 = ""
field157 = ""
field158 = ""
field159 = ""
field160 = ""
field161 = ""
field162 = ""
field163 = ""
field164 = ""
field165 = ""
field166 = ""
field167 = ""
field168 = ""
field169 = ""
field170 = ""
field171 = ""
field172 = ""
field173 = ""
```

---

---

```
field174 = ""
field175 = ""
field176 = ""
field177 = ""
field178 = ""
field179 = ""
field180 = ""
field181 = ""
field182 = ""
field183 = ""
field184 = ""
field185 = ""
field186 = ""
field187 = ""
field188 = ""
field189 = ""
field190 = ""
field191 = ""
field192 = ""
field193 = ""
field194 = ""
field195 = ""
field196 = ""
field197 = ""
field198 = ""
field199 = ""
field200 = ""
field201 = ""
field202 = ""
field203 = ""
field204 = ""
field205 = ""
field206 = ""
field207 = ""
field208 = ""
field209 = ""
field210 = ""
field211 = ""
field212 = ""
field213 = ""
field214 = ""
field215 = ""
field216 = ""
field217 = ""
field218 = ""
field219 = ""
field220 = ""
field221 = ""
field222 = ""
field223 = ""
field224 = ""
field225 = ""
field226 = ""
field227 = ""
field228 = ""
field229 = ""
field230 = ""
```

---



```
field231 = ""
field232 = ""
field233 = ""
field234 = ""
mroname = "RAM_0:Beispiel_.MRO" //Einschub: Übergabe der
recno = 5 //Variablen
reccount = 5
recmax = 20
recsel = 0
dbname = "RAM_0:Beispiel.DAT"
idxname = ""
mriname = "RAM_0:Beispiel_.MRX"
fields = 8
/* Beispiel.MRX */
NUMERIC DIGITS 14
SIGNAL ON ERROR
SIGNAL ON SYNTAX

/* Fügen Sie bitte hier Ihre Programmzeilen ein! */

IF field6 = "er Herr" THEN DO
  field7 = "Herr"
  field8 = "Herrn"
END
ELSE DO
  hlp1 = INDEX(field6, " ")
  field7 = SUBSTR(field6, hlp1 + 1)
  field8 = SUBSTR(field6, hlp1 + 1)
  DROP hlp1
END

/* Zurückschreiben der Feldinhalte */

CALL OPEN(out,mroname,"w")
CALL WRITELN(out, field1)
CALL WRITELN(out, field2)
CALL WRITELN(out, field3)
CALL WRITELN(out, field4)
CALL WRITELN(out, field5)
CALL WRITELN(out, field6)
CALL WRITELN(out, field7)
CALL WRITELN(out, field8)
CALL WRITELN(out, field9)
CALL WRITELN(out, field10)
CALL WRITELN(out, field11)
CALL WRITELN(out, field12)
CALL WRITELN(out, field13)
CALL WRITELN(out, field14)
CALL WRITELN(out, field15)
CALL WRITELN(out, field16)
CALL WRITELN(out, field17)
CALL WRITELN(out, field18)
CALL WRITELN(out, field19)
CALL WRITELN(out, field20)
CALL WRITELN(out, field21)
CALL WRITELN(out, field22)
CALL WRITELN(out, field23)
```

---

CALL WRITELN(out, field24)  
CALL WRITELN(out, field25)  
CALL WRITELN(out, field26)  
CALL WRITELN(out, field27)  
CALL WRITELN(out, field28)  
CALL WRITELN(out, field29)  
CALL WRITELN(out, field30)  
CALL WRITELN(out, field31)  
CALL WRITELN(out, field32)  
CALL WRITELN(out, field33)  
CALL WRITELN(out, field34)  
CALL WRITELN(out, field35)  
CALL WRITELN(out, field36)  
CALL WRITELN(out, field37)  
CALL WRITELN(out, field38)  
CALL WRITELN(out, field39)  
CALL WRITELN(out, field40)  
CALL WRITELN(out, field41)  
CALL WRITELN(out, field42)  
CALL WRITELN(out, field43)  
CALL WRITELN(out, field44)  
CALL WRITELN(out, field45)  
CALL WRITELN(out, field46)  
CALL WRITELN(out, field47)  
CALL WRITELN(out, field48)  
CALL WRITELN(out, field49)  
CALL WRITELN(out, field50)  
CALL WRITELN(out, field51)  
CALL WRITELN(out, field52)  
CALL WRITELN(out, field53)  
CALL WRITELN(out, field54)  
CALL WRITELN(out, field55)  
CALL WRITELN(out, field56)  
CALL WRITELN(out, field57)  
CALL WRITELN(out, field58)  
CALL WRITELN(out, field59)  
CALL WRITELN(out, field60)  
CALL WRITELN(out, field61)  
CALL WRITELN(out, field62)  
CALL WRITELN(out, field63)  
CALL WRITELN(out, field64)  
CALL WRITELN(out, field65)  
CALL WRITELN(out, field66)  
CALL WRITELN(out, field67)  
CALL WRITELN(out, field68)  
CALL WRITELN(out, field69)  
CALL WRITELN(out, field70)  
CALL WRITELN(out, field71)  
CALL WRITELN(out, field72)  
CALL WRITELN(out, field73)  
CALL WRITELN(out, field74)  
CALL WRITELN(out, field75)  
CALL WRITELN(out, field76)  
CALL WRITELN(out, field77)  
CALL WRITELN(out, field78)  
CALL WRITELN(out, field79)  
CALL WRITELN(out, field80)

---

---

CALL WRITELN(out, field81)  
CALL WRITELN(out, field82)  
CALL WRITELN(out, field83)  
CALL WRITELN(out, field84)  
CALL WRITELN(out, field85)  
CALL WRITELN(out, field86)  
CALL WRITELN(out, field87)  
CALL WRITELN(out, field88)  
CALL WRITELN(out, field89)  
CALL WRITELN(out, field90)  
CALL WRITELN(out, field91)  
CALL WRITELN(out, field92)  
CALL WRITELN(out, field93)  
CALL WRITELN(out, field94)  
CALL WRITELN(out, field95)  
CALL WRITELN(out, field96)  
CALL WRITELN(out, field97)  
CALL WRITELN(out, field98)  
CALL WRITELN(out, field99)  
CALL WRITELN(out, field100)  
CALL WRITELN(out, field101)  
CALL WRITELN(out, field102)  
CALL WRITELN(out, field103)  
CALL WRITELN(out, field104)  
CALL WRITELN(out, field105)  
CALL WRITELN(out, field106)  
CALL WRITELN(out, field107)  
CALL WRITELN(out, field108)  
CALL WRITELN(out, field109)  
CALL WRITELN(out, field110)  
CALL WRITELN(out, field111)  
CALL WRITELN(out, field112)  
CALL WRITELN(out, field113)  
CALL WRITELN(out, field114)  
CALL WRITELN(out, field115)  
CALL WRITELN(out, field116)  
CALL WRITELN(out, field117)  
CALL WRITELN(out, field118)  
CALL WRITELN(out, field119)  
CALL WRITELN(out, field120)  
CALL WRITELN(out, field121)  
CALL WRITELN(out, field122)  
CALL WRITELN(out, field123)  
CALL WRITELN(out, field124)  
CALL WRITELN(out, field125)  
CALL WRITELN(out, field126)  
CALL WRITELN(out, field127)  
CALL WRITELN(out, field128)  
CALL WRITELN(out, field129)  
CALL WRITELN(out, field130)  
CALL WRITELN(out, field131)  
CALL WRITELN(out, field132)  
CALL WRITELN(out, field133)  
CALL WRITELN(out, field134)  
CALL WRITELN(out, field135)  
CALL WRITELN(out, field136)  
CALL WRITELN(out, field137)

---

CALL WRITELN(out, field138)  
CALL WRITELN(out, field139)  
CALL WRITELN(out, field140)  
CALL WRITELN(out, field141)  
CALL WRITELN(out, field142)  
CALL WRITELN(out, field143)  
CALL WRITELN(out, field144)  
CALL WRITELN(out, field145)  
CALL WRITELN(out, field146)  
CALL WRITELN(out, field147)  
CALL WRITELN(out, field148)  
CALL WRITELN(out, field149)  
CALL WRITELN(out, field150)  
CALL WRITELN(out, field151)  
CALL WRITELN(out, field152)  
CALL WRITELN(out, field153)  
CALL WRITELN(out, field154)  
CALL WRITELN(out, field155)  
CALL WRITELN(out, field156)  
CALL WRITELN(out, field157)  
CALL WRITELN(out, field158)  
CALL WRITELN(out, field159)  
CALL WRITELN(out, field160)  
CALL WRITELN(out, field161)  
CALL WRITELN(out, field162)  
CALL WRITELN(out, field163)  
CALL WRITELN(out, field164)  
CALL WRITELN(out, field165)  
CALL WRITELN(out, field166)  
CALL WRITELN(out, field167)  
CALL WRITELN(out, field168)  
CALL WRITELN(out, field169)  
CALL WRITELN(out, field170)  
CALL WRITELN(out, field171)  
CALL WRITELN(out, field172)  
CALL WRITELN(out, field173)  
CALL WRITELN(out, field174)  
CALL WRITELN(out, field175)  
CALL WRITELN(out, field176)  
CALL WRITELN(out, field177)  
CALL WRITELN(out, field178)  
CALL WRITELN(out, field179)  
CALL WRITELN(out, field180)  
CALL WRITELN(out, field181)  
CALL WRITELN(out, field182)  
CALL WRITELN(out, field183)  
CALL WRITELN(out, field184)  
CALL WRITELN(out, field185)  
CALL WRITELN(out, field186)  
CALL WRITELN(out, field187)  
CALL WRITELN(out, field188)  
CALL WRITELN(out, field189)  
CALL WRITELN(out, field190)  
CALL WRITELN(out, field191)  
CALL WRITELN(out, field192)  
CALL WRITELN(out, field193)  
CALL WRITELN(out, field194)

---

```
CALL WRITELN(out, field195)
CALL WRITELN(out, field196)
CALL WRITELN(out, field197)
CALL WRITELN(out, field198)
CALL WRITELN(out, field199)
CALL WRITELN(out, field200)
CALL WRITELN(out, field201)
CALL WRITELN(out, field202)
CALL WRITELN(out, field203)
CALL WRITELN(out, field204)
CALL WRITELN(out, field205)
CALL WRITELN(out, field206)
CALL WRITELN(out, field207)
CALL WRITELN(out, field208)
CALL WRITELN(out, field209)
CALL WRITELN(out, field210)
CALL WRITELN(out, field211)
CALL WRITELN(out, field212)
CALL WRITELN(out, field213)
CALL WRITELN(out, field214)
CALL WRITELN(out, field215)
CALL WRITELN(out, field216)
CALL WRITELN(out, field217)
CALL WRITELN(out, field218)
CALL WRITELN(out, field219)
CALL WRITELN(out, field220)
CALL WRITELN(out, field221)
CALL WRITELN(out, field222)
CALL WRITELN(out, field223)
CALL WRITELN(out, field224)
CALL WRITELN(out, field225)
CALL WRITELN(out, field226)
CALL WRITELN(out, field227)
CALL WRITELN(out, field228)
CALL WRITELN(out, field229)
CALL WRITELN(out, field230)
CALL WRITELN(out, field231)
CALL WRITELN(out, field232)
CALL WRITELN(out, field233)
CALL WRITELN(out, field234)
CALL CLOSE(out)

/* Programm beenden */

CALL REMLIB("rexxmathlib.library")
CALL REMLIB("rexxsupport.library")
EXIT

/* Fehlerbehandlung */

SYNTAX:
SAY "##SYNTAX"
EXIT 10

ERROR:
SAY "##ERROR"
EXIT 10
```

---

/\* Fügen Sie hier ggfs. Ihre Funktionen und Prozeduren ein! \*/

MicroBase hat also Code für die Konstanten, Feldinhalte und Variablen hinzugefügt.

## 1.54 Anwendungsbeispiele

Für ARexx-Skripte sind unter anderem folgende Anwendungsgebiete denkbar:

Aktualisierung von Datensätzen

Skripte ermöglichen hier sehr flexible Auswahlmöglichkeiten; auch ist die Verknüpfung mehrerer Kriterien möglich.

Auch die Möglichkeiten zur Änderung eines Datensatzes sind vielfältiger als die eines Datenbankverwaltungssystems selbst.

Ausführung von Berechnungen

In Skripten können Sie umfangreiche Berechnungen ausführen; im Gegensatz zu Berechnungsformeln sind Sie nicht auf nur eine Zeile beschränkt.

Es besteht die Möglichkeit, eigene Funktionen zu erstellen.

Berechnungen können auch über mehrere Datensätze hinweg vorgenommen werden.

Spezielle Exportskripte

Falls Ihnen die Exportmöglichkeiten von MicroBase nicht ausreichen, können Sie Exportskripte schreiben.

In Exportskripten können Sie das Format der Felder und die Feld- bzw. Datensatztrennzeichen selbst bestimmen. Sie können auch für jedes Feld entscheiden, ob es überhaupt exportiert werden soll.

Hinweis: Sie müssen neben der Ausgabedatei eine zweite Datei öffnen. MicroBase erzeugt eine Fehlermeldung, wenn es keine (korrekte) Ausgabedatei vorfindet.

Flexible Berichte (Listen)

Mit Skripten können Sie den Inhalt des Berichtes frei bestimmen.

Eine interessante Möglichkeit ist z. B. die Erstellung von Listen mit Gesamtsummen. Gehen Sie dazu wie folgt vor:

- 1 Setzen Sie den Wert einer Variablen (z. B. Summe) auf 0 (null), wenn der erste Datensatz bearbeitet wird. In diesem Fall hat die Systemvariable

recno den Wert 1.

- 2 Wenn Sie nicht im ersten Datensatz sind, müssen Sie den aktuellen Wert der Variablen einlesen - z. B. mit `Summe = GETCLIP("SUM")`.
- 3 Erhöhen Sie in jedem Programmdurchlauf den Wert der Variablen um den Wert des gewünschten Feldes.
- 4 Speichern Sie den Wert der Variablen in einer CLIP-Liste ab - z. B. mit `CALL SETCLIP("SUM", Summe)`
- 5 Wenn das Programm im letzten Datensatz angekommen ist (`recno = reccount`), hat die Variable den korrekten Wert und Sie können sie ausdrucken. Vergessen Sie nicht, die Variable aus der CLIP-Liste zu entfernen!

## 1.55 Die Erstellung interaktiver Skripte

Normalerweise erlaubt MicroBase keine interaktiven ARexx-Skripte. Die Ausgabe von Bildschirmmeldungen ist ebenso unmöglich wie die Anforderung von Benutzereingaben.

Dies resultiert daraus, dass die Skripte von der Workbench gestartet werden und deshalb kein eigenes Programmfenster besitzen (MicroBase öffnet lediglich ein CLI-Fenster für den Aufruf des ARexx-Interpreters).

Die Lösung liegt also darin, im Skript selbst ein Fenster zu öffnen und dann dieses für die Interaktion mit dem Benutzer zu verwenden.

Das folgende Beispiel fordert den Benutzer auf, den Wert des ersten Datenfeldes neu einzugeben:

```
/* Fenster.MRX */
NUMERIC DIGITS 14
SIGNAL ON ERROR
SIGNAL ON SYNTAX

/* Fügen Sie bitte hier Ihre Programmzeilen ein! */

CALL OPEN("win","CON:0/0/640/256/" || mrxname)
CALL WRITELN("win", "Geben Sie einen neuen Inhalt für Feld 1 ein:")
field1 = READLN("win")
CALL WRITELN("win", "Neuer Inhalt:")
CALL WRITELN("win", field1)
CALL WRITELN("win", "Weiter mit <RETURN>")
dummy = READCH("win",1)
DROP dummy
CALL CLOSE("win")

/* Zurückschreiben der Feldinhalte */

CALL OPEN(out,mroname,"w")
CALL WRITELN(out, field1)
CALL WRITELN(out, field2)
CALL WRITELN(out, field3)
CALL WRITELN(out, field4)
```

---

CALL WRITELN(out, field5)  
CALL WRITELN(out, field6)  
CALL WRITELN(out, field7)  
CALL WRITELN(out, field8)  
CALL WRITELN(out, field9)  
CALL WRITELN(out, field10)  
CALL WRITELN(out, field11)  
CALL WRITELN(out, field12)  
CALL WRITELN(out, field13)  
CALL WRITELN(out, field14)  
CALL WRITELN(out, field15)  
CALL WRITELN(out, field16)  
CALL WRITELN(out, field17)  
CALL WRITELN(out, field18)  
CALL WRITELN(out, field19)  
CALL WRITELN(out, field20)  
CALL WRITELN(out, field21)  
CALL WRITELN(out, field22)  
CALL WRITELN(out, field23)  
CALL WRITELN(out, field24)  
CALL WRITELN(out, field25)  
CALL WRITELN(out, field26)  
CALL WRITELN(out, field27)  
CALL WRITELN(out, field28)  
CALL WRITELN(out, field29)  
CALL WRITELN(out, field30)  
CALL WRITELN(out, field31)  
CALL WRITELN(out, field32)  
CALL WRITELN(out, field33)  
CALL WRITELN(out, field34)  
CALL WRITELN(out, field35)  
CALL WRITELN(out, field36)  
CALL WRITELN(out, field37)  
CALL WRITELN(out, field38)  
CALL WRITELN(out, field39)  
CALL WRITELN(out, field40)  
CALL WRITELN(out, field41)  
CALL WRITELN(out, field42)  
CALL WRITELN(out, field43)  
CALL WRITELN(out, field44)  
CALL WRITELN(out, field45)  
CALL WRITELN(out, field46)  
CALL WRITELN(out, field47)  
CALL WRITELN(out, field48)  
CALL WRITELN(out, field49)  
CALL WRITELN(out, field50)  
CALL WRITELN(out, field51)  
CALL WRITELN(out, field52)  
CALL WRITELN(out, field53)  
CALL WRITELN(out, field54)  
CALL WRITELN(out, field55)  
CALL WRITELN(out, field56)  
CALL WRITELN(out, field57)  
CALL WRITELN(out, field58)  
CALL WRITELN(out, field59)  
CALL WRITELN(out, field60)  
CALL WRITELN(out, field61)

---



---

CALL WRITELN(out, field62)  
CALL WRITELN(out, field63)  
CALL WRITELN(out, field64)  
CALL WRITELN(out, field65)  
CALL WRITELN(out, field66)  
CALL WRITELN(out, field67)  
CALL WRITELN(out, field68)  
CALL WRITELN(out, field69)  
CALL WRITELN(out, field70)  
CALL WRITELN(out, field71)  
CALL WRITELN(out, field72)  
CALL WRITELN(out, field73)  
CALL WRITELN(out, field74)  
CALL WRITELN(out, field75)  
CALL WRITELN(out, field76)  
CALL WRITELN(out, field77)  
CALL WRITELN(out, field78)  
CALL WRITELN(out, field79)  
CALL WRITELN(out, field80)  
CALL WRITELN(out, field81)  
CALL WRITELN(out, field82)  
CALL WRITELN(out, field83)  
CALL WRITELN(out, field84)  
CALL WRITELN(out, field85)  
CALL WRITELN(out, field86)  
CALL WRITELN(out, field87)  
CALL WRITELN(out, field88)  
CALL WRITELN(out, field89)  
CALL WRITELN(out, field90)  
CALL WRITELN(out, field91)  
CALL WRITELN(out, field92)  
CALL WRITELN(out, field93)  
CALL WRITELN(out, field94)  
CALL WRITELN(out, field95)  
CALL WRITELN(out, field96)  
CALL WRITELN(out, field97)  
CALL WRITELN(out, field98)  
CALL WRITELN(out, field99)  
CALL WRITELN(out, field100)  
CALL WRITELN(out, field101)  
CALL WRITELN(out, field102)  
CALL WRITELN(out, field103)  
CALL WRITELN(out, field104)  
CALL WRITELN(out, field105)  
CALL WRITELN(out, field106)  
CALL WRITELN(out, field107)  
CALL WRITELN(out, field108)  
CALL WRITELN(out, field109)  
CALL WRITELN(out, field110)  
CALL WRITELN(out, field111)  
CALL WRITELN(out, field112)  
CALL WRITELN(out, field113)  
CALL WRITELN(out, field114)  
CALL WRITELN(out, field115)  
CALL WRITELN(out, field116)  
CALL WRITELN(out, field117)  
CALL WRITELN(out, field118)

---

CALL WRITELN(out, field119)  
CALL WRITELN(out, field120)  
CALL WRITELN(out, field121)  
CALL WRITELN(out, field122)  
CALL WRITELN(out, field123)  
CALL WRITELN(out, field124)  
CALL WRITELN(out, field125)  
CALL WRITELN(out, field126)  
CALL WRITELN(out, field127)  
CALL WRITELN(out, field128)  
CALL WRITELN(out, field129)  
CALL WRITELN(out, field130)  
CALL WRITELN(out, field131)  
CALL WRITELN(out, field132)  
CALL WRITELN(out, field133)  
CALL WRITELN(out, field134)  
CALL WRITELN(out, field135)  
CALL WRITELN(out, field136)  
CALL WRITELN(out, field137)  
CALL WRITELN(out, field138)  
CALL WRITELN(out, field139)  
CALL WRITELN(out, field140)  
CALL WRITELN(out, field141)  
CALL WRITELN(out, field142)  
CALL WRITELN(out, field143)  
CALL WRITELN(out, field144)  
CALL WRITELN(out, field145)  
CALL WRITELN(out, field146)  
CALL WRITELN(out, field147)  
CALL WRITELN(out, field148)  
CALL WRITELN(out, field149)  
CALL WRITELN(out, field150)  
CALL WRITELN(out, field151)  
CALL WRITELN(out, field152)  
CALL WRITELN(out, field153)  
CALL WRITELN(out, field154)  
CALL WRITELN(out, field155)  
CALL WRITELN(out, field156)  
CALL WRITELN(out, field157)  
CALL WRITELN(out, field158)  
CALL WRITELN(out, field159)  
CALL WRITELN(out, field160)  
CALL WRITELN(out, field161)  
CALL WRITELN(out, field162)  
CALL WRITELN(out, field163)  
CALL WRITELN(out, field164)  
CALL WRITELN(out, field165)  
CALL WRITELN(out, field166)  
CALL WRITELN(out, field167)  
CALL WRITELN(out, field168)  
CALL WRITELN(out, field169)  
CALL WRITELN(out, field170)  
CALL WRITELN(out, field171)  
CALL WRITELN(out, field172)  
CALL WRITELN(out, field173)  
CALL WRITELN(out, field174)  
CALL WRITELN(out, field175)

---

CALL WRITELN(out, field176)  
CALL WRITELN(out, field177)  
CALL WRITELN(out, field178)  
CALL WRITELN(out, field179)  
CALL WRITELN(out, field180)  
CALL WRITELN(out, field181)  
CALL WRITELN(out, field182)  
CALL WRITELN(out, field183)  
CALL WRITELN(out, field184)  
CALL WRITELN(out, field185)  
CALL WRITELN(out, field186)  
CALL WRITELN(out, field187)  
CALL WRITELN(out, field188)  
CALL WRITELN(out, field189)  
CALL WRITELN(out, field190)  
CALL WRITELN(out, field191)  
CALL WRITELN(out, field192)  
CALL WRITELN(out, field193)  
CALL WRITELN(out, field194)  
CALL WRITELN(out, field195)  
CALL WRITELN(out, field196)  
CALL WRITELN(out, field197)  
CALL WRITELN(out, field198)  
CALL WRITELN(out, field199)  
CALL WRITELN(out, field200)  
CALL WRITELN(out, field201)  
CALL WRITELN(out, field202)  
CALL WRITELN(out, field203)  
CALL WRITELN(out, field204)  
CALL WRITELN(out, field205)  
CALL WRITELN(out, field206)  
CALL WRITELN(out, field207)  
CALL WRITELN(out, field208)  
CALL WRITELN(out, field209)  
CALL WRITELN(out, field210)  
CALL WRITELN(out, field211)  
CALL WRITELN(out, field212)  
CALL WRITELN(out, field213)  
CALL WRITELN(out, field214)  
CALL WRITELN(out, field215)  
CALL WRITELN(out, field216)  
CALL WRITELN(out, field217)  
CALL WRITELN(out, field218)  
CALL WRITELN(out, field219)  
CALL WRITELN(out, field220)  
CALL WRITELN(out, field221)  
CALL WRITELN(out, field222)  
CALL WRITELN(out, field223)  
CALL WRITELN(out, field224)  
CALL WRITELN(out, field225)  
CALL WRITELN(out, field226)  
CALL WRITELN(out, field227)  
CALL WRITELN(out, field228)  
CALL WRITELN(out, field229)  
CALL WRITELN(out, field230)  
CALL WRITELN(out, field231)  
CALL WRITELN(out, field232)

---

```

CALL WRITELN(out, field233)
CALL WRITELN(out, field234)
CALL CLOSE(out)

/* Programm beenden */

CALL REMLIB("rexxmathlib.library")
CALL REMLIB("rexxsupport.library")
EXIT

/* Fehlerbehandlung */

SYNTAX:
SAY "##SYNTAX"
EXIT 10

ERROR:
SAY "##ERROR"
EXIT 10

/* Fügen Sie hier ggfs. Ihre Funktionen und Prozeduren ein! */

```

Wie Sie sehen, wird die Ein- und Ausgabe des Skripts über das neu geöffnete Fenster "win" abgewickelt. Das Fenster wird am Ende des Skripts geschlossen, nachdem auf einen Tastendruck gewartet wurde.

Die normalen ARexx-Befehle für Textausgabe und Benutzereingaben – SAY und PULL – können nicht verwendet werden. Das Fenster wird wie eine Datei behandelt; deshalb werden die Anweisungen READLN und WRITELN benutzt.

## 1.56 Die Entwicklung eines ARexx-Skriptes

Die Entwicklung eines ARexx-Skriptes ist am einfachsten, wenn Sie eines der mitgelieferten Skripte an Ihre Bedürfnisse abwandeln.

Folgende Basisskripte stehen zur Verfügung:

| Skript      | Bemerkung                                                                                                                                                                                                      |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Empty.MRX   | Dies ist ein völlig "leeres" Skript. Mit dieser Basis ist eine Interaktion mit dem Anwender nicht möglich.                                                                                                     |
| Fenster.MRX | Dieses Skript ist die Grundlage für interaktive ARexx-Skripte. Beachten Sie bitte, dass Sie statt SAY ... und PULL ... die Anweisungen CALL WRITELN("win", ...) bzw. ... = CALL READLN("win") benutzen müssen. |

Nach dem Laden des Skriptes bestehen folgende Bearbeitungsmöglichkeiten:

- Öffnen neuer Bibliotheken (Libraries) zu Beginn des Skriptes. Vergewissern Sie sich bitte, dass von Ihnen geöffnete Bibliotheksdateien auch wieder geschlossen werden.

- Erstellung von Funktionen und Prozeduren unterhalb der die korrekte Stelle kennzeichnenden Kommentarzeile.
- Einfügen Ihres Programmcodes, der sich auf die Feldinhalte bezieht und diese ggf. verändert.
- Öffnen von Dateien, z. B. für den Export von Datensätzen. Diese Dateien müssen Sie selbst wieder schließen.

Bitte denken Sie daran, die Inhalte der Feldinhalte in die Ausgabedatei zu schreiben. Der Name der Ausgabedatei steht in der Variablen "mroname".

## 1.57 Datenbankentwicklung

Die Entwicklung eines Datenbanksystems besteht aus den folgenden Schritten:

- (1) Analyse der zu erfassenden Daten
- (2) Entwurf der Datenbank
- (3) Definition der Datenbank
- (4) Erstellen von Verknüpfungen zwischen Datenbanken

Im Folgenden wird davon ausgegangen, dass eine Auftragsverwaltung auf elektronische Datenverarbeitung umgestellt werden soll.

- (1) Analyse der zu erfassenden Daten

Die zu erfassenden Daten müssen analysiert und zusammengestellt werden. Im Falle unserer Auftragsverwaltung müssen die Daten der zu liefernden Artikel sowie die Adressen der Auftraggeber verwaltet werden.

Eine Ermittlung der Daten erfolgt durch eine Mitarbeiterbefragung bzw. durch eine Überprüfung ihrer Tätigkeit im Hinblick auf die Art und Menge der anfallenden Informationen.

- (2) Entwurf der Datenbank

Jetzt müssen die zu erstellenden Datenbanken entworfen werden. Hierzu gehört neben der Definition der Datenfelder (Feldname, Feldtyp und Feldlänge) auch deren Aufteilung auf eine oder mehrere Datenbanken.

Bei der Verwendung von mehreren Datenbanken muss entschieden werden, wie diese Datenbanken miteinander verknüpft werden sollen – über ein gemeinsames Datenfeld oder über einen Index?

Hinweis: MicroBase unterstützt Indizes nicht und verwendet Relationscaches, um die Auflösung der Relationen zu beschleunigen.

Folgende Bedingungen müssen dabei erfüllt sein:

- Vermeidung von Redundanz (keine Mehrfachspeicherung der Daten)
-

- Effizienz (statt weniger großer besser viele kleine Datenfelder; Felder nicht länger als notwendig)
- Verwendung des richtigen Datentyps (z. B. "Text" bei Telefonnummern wegen der möglichen runden Klammern) zur Vermeidung von Inkonsistenzen
- Verwendung eines eindeutigen Ordnungsbegriffs, über den die Verknüpfung der Datenbanken erfolgen kann

Der Entwurf erfolgt mit einem Schema wie diesem:

| Nr. | Datenbank | Datenfeld   | Datentyp | Feldlänge | Verknüpfung mit     |
|-----|-----------|-------------|----------|-----------|---------------------|
| 1   | Aufträge  | Artikelnr.  | Nummer   | 10        | ./.                 |
| 2   | Aufträge  | Artikel     | Relation | 1         | Artikel:Bezeichnung |
| ... |           |             |          |           |                     |
| 1   | Artikel   | ArtNr.      | Nummer   | 10        | ./.                 |
| 2   | Artikel   | Bezeichnung | Text     | 40        | ./.                 |
| ... |           |             |          |           |                     |

In diesem Beispiel wird in der Datei "Aufträge" eine Verknüpfung zu der Datenbank "Artikel" hergestellt, um die Bezeichnung des Artikels zu ermitteln. Die beiden Datenbanken werden dazu über die Felder "Artikelnummer" und "ArtNr." miteinander verknüpft, da diese den gleichen Inhalt haben.

Es wurde erreicht, dass nur noch die für die Verknüpfung benötigte Artikelnummer doppelt gespeichert wird; die 40 Zeichen für die Artikelbezeichnung wurden eingespart und somit Redundanz vermieden. Bei 2 000 Datensätzen werden so schon 80 KB gespart!

### (3) Definition der Datenbanken

Die in Schritt (2) entworfenen Datenbanken werden in diesem Schritt mit Hilfe eines Datenbankverwaltungssystems (kurz DBMS) – z. B. mit MicroBase – erstellt.

Falls das DBMS Optionen zur Gewährleistung der Datensicherheit und des Datenschutzes bietet, sollten Sie diese möglichst aktivieren (z. B. eine Plausibilitätskontrolle für Datumsangaben oder einen Passwortschutz.) So vermeiden Sie Inkonsistenzen in Ihren Daten wie z. B. den "31.02.1997". Die genannten Optionen sind in MicroBase verfügbar.

### (4) Erstellen von Verknüpfungen zwischen Datenbanken

Nach der Erstellung der Nebendatenbanken können Sie die Hauptdatenbank erstellen, die sich über Verknüpfungen auf die anderen Datenbanken bezieht.

Hierbei ist darauf zu achten, dass die richtigen Datenfelder miteinander verknüpft werden. Die Datentypen der Felder müssen gleich sein.

Es werden hauptsächlich zwei Arten von Datenbankverwaltungssystemen unterschieden, die Verknüpfungen unterstützen:

hierarchische DBMS: Hier werden einer Hauptdatenbank mehrere Nebendatenbanken zugeordnet. Eine Verknüpfung ist nur über die Hauptdatenbank möglich.

relationale DBMS: Hier kann jede Datenbank mit einer beliebigen anderen verknüpft werden. Ein Umweg über die Hauptdatenbank ist nicht notwendig. Datenbanken werden hier als eine Art von Tabellen interpretiert.

Relationale Datenbanken sind also wesentlich flexibler als hierarchische.

Hinweis: MicroBase verfügt zwar über die Funktionalität eines relationalen Datenbankverwaltungssystems, unterstützt aber nicht die Anzeige der Datenbanken in Tabellenform. Dies ist jedoch für Relationen zwischen mehr als zwei Datensätzen notwendig.

Das DBMS sollte möglichst auf die Einhaltung der referenziellen Integrität achten. Z. B. sollte eine Verknüpfung mit der Artikelnummer 4 711 nicht möglich sein, wenn es den entsprechenden Artikel nicht mehr gibt.

MicroBase überprüft Datensätze beim Eingeben, Importieren, Sortieren, Löschen usw. auf die Einhaltung der referenziellen Integrität und hilft Ihnen, alle die Referenzintegrität verletzenden Datensätze zu finden (siehe Menüpunkt "Extras/Caches aktualisieren").

## 1.58 Zahlenformate

Für numerische und Berechnungsfelder können Zahlenformate definiert werden. Diese werden beim Ausdruck von Listen und Etiketten verwendet, um die Zahlen in einem einheitlichen Format zu drucken.

Durch Zahlenformate können die Zahlen z. B. auf zwei Nachkommastellen gerundet und mit einem Währungszeichen ausgegeben werden.

In Zahlenformaten sind folgende Zeichen erlaubt:

Zeichen Wirkung

| Zeichen | Wirkung                                                      |
|---------|--------------------------------------------------------------|
| #       | Ziffer bzw. Leerzeichen als Füllzeichen                      |
| .       | Tausendertrennzeichen                                        |
| ,       | Dezimalkomma                                                 |
| *       | Ziffer bzw. Verwendung des Sterns als Füllzeichen            |
| -       | Position des negativen Vorzeichens (falls notwendig)         |
| +       | Position des Vorzeichens (auch positiv)                      |
| \$      | Währungszeichen '\$' voranstellen                            |
| ^^^     | Exponentialformat benutzen (z. B. 2.0E+09 statt 200000000.0) |
| _       | nächstes Zeichen nicht interpretieren und einfach übernehmen |

Alle anderen Zeichen werden unverändert übernommen.

Zahlen werden immer auf die angegebene Zahl von Nachkommastellen gerundet und

dazu ggf. mit Leerzeichen aufgefüllt, falls die Zahl weniger Ziffern hat als im Zahlenformat angegeben sind.

Beispiele

| Format      | Zahl      | Ergebnis    |
|-------------|-----------|-------------|
| *####       | 11.78     | ***12       |
| DM #.###,## | 1234.5678 | DM 1.234,57 |

Fehler

Zahlenformate können folgende Codes als Fehlermeldungen liefern:

Code Bedeutung

|   |                                                                                            |
|---|--------------------------------------------------------------------------------------------|
| % | Die Zahl ist länger, als das Zahlenformat zulässt.                                         |
| ? | Das Zahlenformat ist konnte nicht interpretiert werden, weil es ungültige Zeichen enthält. |

## 1.59 Problemlösungen

Bildschirmmasken werden nicht korrekt angezeigt

Da die Bildschirmmasken des Editors auf den Standardzeichensatz »Topaz 8« ausgelegt sind, sollte dieser oder ein anderer nicht proportionaler Zeichensatz mit 8 Punkten in den Voreinstellungen aktiviert werden.

Das Eingabefeld ist leer, obwohl es Text enthält

Bei Feldern mit mehr als 53 Zeichen werden die Daten auf der linken Seite des Cursors nicht angezeigt. Sie sind jedoch immer noch vorhanden und können mit Hilfe der Cursortasten erreicht werden.

Das eingegebene Datum wird nicht akzeptiert

MicroBase schreibt für Datumsangaben das Format "TT.MM.JJJJ" vor.

Sie sollten zunächst prüfen, ob Ihre Eingabe diesem Format entspricht und ob das Datum gültig ist (z. B. gibt es keinen 31.04.1997). Der 29.02. wird nur bei Schaltjahren akzeptiert.

Zahlen werden falsch gedruckt

In diesem Fall sollten Sie das  
Zahlenformat  
korrigieren.



Das Programm sortiert falsch

Vergewissern Sie sich bitte zunächst, dass Sie die korrekte Sortiermethode und -richtung gewählt haben.

Falls Sie die phonetische Suche aktiviert haben, werden alle Textfelder nach ihren Lautwerten sortiert - "Müller" kann also vor "Miller" einsortiert werden.

Feldnamen werden nicht korrekt in dBASE-Dateien konvertiert

Dies ist eine Einschränkung von dBASE: Feldnamen dürfen nur 10 Zeichen lang sein. Es sind nur Großbuchstaben erlaubt; Leerzeichen dürfen nicht verwendet werden.

Umlaute werden nicht korrekt dargestellt

Der Zeichensatz der Datenbankdatei wurde beim Importieren nicht konvertiert (bzw. konvertiert, obwohl es nicht notwendig war).

Am Anfang (Ende) des Datenfeldes befinden sich merkwürdige Zeichen

Dies ist ein Hinweis auf Fehler beim Datenbankzugriff.

Sie sollten zunächst den Kompatibilitätsmodus aktivieren. Falls dies keine Wirkung hat, müssen Sie die Datenbankdatei reorganisieren lassen.

Das Programm stürzt beim (nach dem) Sortieren ab

MicroBase verwendet ein beschleunigtes Sortierverfahren, das in einigen Fällen zu Problemen führen kann. Sie sollten den Kompatibilitätsmodus aktivieren, damit das Sortieren verlangsamt wird.

---