

DDE und ODBC zwischen SAS 6.12 und MS Office 2000

Paul Eberhard Rudolph^{1)*}, Carina Ortseifen^{2)**}, Armin Tuchscherer¹⁾

1) Forschungsinstitut für die Biologie landwirtschaftlicher Nutztiere Dummerstorf-Rostock,
Wilhelm-Stahl-Allee 2, D-18196 Dummerstorf

* eMail: rudolph@fbn-dummerstorf.de

2) Ruprecht-Karls-Universität Heidelberg, Universitätsrechenzentrum,
Im Neuenheimer Feld 293, D-69120 Heidelberg

** eMail: carina.ortseifen@urz.uni-heidelberg.de

1 Einleitung

Der Datenaustausch zwischen SAS und anderen Windows-Anwendungen ist mit DDE (Dynamic Data Exchange) und ODBC (Open Data Base Connectivity) inzwischen relativ komfortabel. Mit der angekündigten SAS-Version 8 für Windows-Betriebssysteme soll dieser Datenaustausch aus der Sicht von SAS künftig noch komfortabler ohne detaillierte DDE- bzw. ODBC-Kenntnisse möglich sein. Allerdings wird sicher auch nach Auslieferung der neuen SAS-Version eine große Zahl von SAS-Anwendern weiterhin mit der Version 6.12 arbeiten. Ein Umstieg von MS Office95/97 auf MS Office2000 dagegen ist bereits für viele Anwender Realität geworden. In diesem Zusammenhang ist es bestimmt von Interesse, wie der u.a. von RUDOLPH/GEISSLER/ORTSEIFEN (1998) beschriebene Datenaustausch mit DDE bzw. ODBC zwischen SAS und einigen MS Office95/97-Produkten mit den entsprechenden Office2000-Produkten funktioniert.

In den folgenden Abschnitten werden in Anlehnung an die Arbeit von RUDOLPH/GEISSLER/ORTSEIFEN (1998), vgl. auch SAS-Anwenderhandbuch im Netz, Version 2.1 vom August 1999 (www.rzuser.uni-heidelberg.de/~x16/sas-ah.html), Hinweise zum Datenaustausch zwischen SAS, Version 6.12 und den (deutschsprachigen) MS Office2000-Produkten Access, Excel und Word mittels DDE und ODBC gegeben. Die dargestellten Bildschirm- bzw. Fensterinhalte wurden unter Windows NT 4.0 (Servicepack 5) erstellt. Unter Windows 95/98 können die entsprechenden Inhalte eventuell von den hier dargestellten abweichen. Bei den beschriebenen Vorgehensweisen sind Fenstertitel durch Unterstreichen/fett und die Auswahl aus in diesen Fenstern vorhandenen Schaltflächen oder Menüs durch Kursivschrift/fett gekennzeichnet.

2 DDE (Dynamic Data Exchange)

Beim dynamischen Datenaustausch (DDE) konnten zwischen Office 95/97 und Office 2000 keine Unterschiede festgestellt werden. Das Vorgehen wird im folgenden dennoch an einem Beispiel aufgezeigt und auf mögliche Probleme hingewiesen. In Ergänzung zu den Ausführungen in RUDOLPH/GEISSLER/ORTSEIFEN (1998) wird vorgestellt, wie durch spezielle Befehle die Vorgänge innerhalb der Excel-Anwendung automatisiert werden können.

2.1 Datenexport von SAS nach MS Excel2000

Zur Beschreibung des Datenexports von SAS nach Excel2000 sollen die in der folgenden Tabelle aufgelisteten Beispieldaten verwendet werden.

Tabelle 2.1: Beispieldaten

Name	Alter	Gewicht
Peter	23	67.5
Anke	34	51.0
Jochen	56	89.5
Elke	44	56.5

Die Daten aus der Tabelle 2.1 liegen als SAS-Datei **data.sd2** mit den Variablen `name`, `alter` und `gewicht` im Verzeichnis (Ordner) `c:\sas\dd` vor. Nachdem nun eine SAS-Sitzung und eine Excel-Sitzung gestartet wurden, sind folgende SAS-Anweisungen notwendig:

```
/* 1. Anweisung LIBNAME: Vereinbarung der Bibliothek libdd */
libname libdd 'c:\sas\dd';

/* 2. Anweisung FILENAME: Vereinbarung der Verbindung zwischen Excel und SAS */
filename test dde 'excel | Tabelle1 ! z1s1:z4s3';

/* 3. Datenschnitt zur Übertragung der Datenwerte */
data _null_;
  set libdd.data;
  file test;
  put name alter gewicht:commax6.2;
run;
```

Für die Gewichtsangaben, als Zahlenwerte mit Kommas zum Trennen der Dezimalstellen, ist ebenso wie für Datumsangaben und andere Nichtstandard-Daten die Angabe eines geeigneten Formats notwendig, damit die Werte korrekt in Excel übertragen werden.

Bemerkung 2.1:

Probleme treten auf, wenn in der SAS Datei fehlende Werte enthalten sind. Fehlende Werte bei numerischen Variablen werden in der Excel-Datei als . (Punkt) dargestellt. Problematischer sind fehlende Werte bei Textvariablen, da diese in zwei Tabulatorzeichen verwandelt werden und dadurch zwei leere Felder verursachen, wodurch die übrigen Variablenwerte in der gleichen Zeile um eine Position nach rechts verschoben werden und der Wert der letzten Variable verschwindet. Ein Ausweg kann dadurch geschaffen werden, dass die Anweisung FILE um die Option NOTAB erweitert wird und im Datenschnitt zwischen den Variablennamen ein Tabulatorzeichen ('09'x) eingefügt wird.

```
data _null_;
  set libdd.data;
  file test notab;
  put name '09'x alter '09'x gewicht:commax6.2;
run;
```

Die Option NOTAB ist auch dann von Vorteil, wenn in der SAS-Datei Zeichenketten auftreten, die Leerzeichen enthalten.

Ein weiteres Problem betrifft die Beschränkung der Spaltenzahl in Excel: Mehr als 256 Variablen können nicht übertragen werden. Das SAS System gibt in solch einem Fall leider keine Fehlermeldung aus. Man sieht lediglich an der leeren Excel-Tabelle, dass mit der Übertragung irgendetwas schiefgegangen sein muss.

Automation der Excel-Befehle

Soll die Übertragung von SAS in Excel-Tabellen automatisiert werden, etwa durch Ablauf eines SAS-Makros, dann möchte man natürlich auch die Excel-Sitzung ohne Mausklick starten und die fertige Excel-Tabelle in eine neue (oder vorhandene) Tabelle abspeichern. Dazu bedient man sich der Anweisung X und spezieller Befehle innerhalb der Anweisungen PUT. Voraussetzung dafür ist lediglich, dass man den vollständigen Pfad der Datei EXCEL.EXE auf seinem Rechner kennt.

```
/* 1. Starten der Excel-Anwendung durch Aufruf der entsprechenden Exe-Datei. Die beiden
Options NOXWAIT und NOXSYNC sorgen dafür, dass die Kontrolle nach dem Star-
ten der Excel-Anwendung wieder an das SAS System zurückgeht.*/
options noxwait noxsync;
x ' "c:\programme\ms office2000\office\excel.exe" ';

/* 2. Eventuell muss man dem System etwas Zeit gönnen, bis die Excel-Sitzung komplett
aufgebaut ist (hier: 5 Sekunden). */
data _null_;
  x=sleep(5);
run ;

/* 3. Für die weiteren Excel-Befehle wird die nächste Anweisung FILENAME benötigt. */
filename commands dde 'excel | system';

/* 4. Erzeugte Excel-Tabelle wird unter dem Namen data.xls im Ordner c:\excel abgespei-
chert. */
data _null_;
  file commands;
  put '[save.as("c:\excel\data.xls")]';
run;

/* 5. Beenden der Excel-Anwendung mit QUIT. */
data _null_;
  file commands;
  put '[quit]';
run;
```

Wegen des Leerzeichens in der Ordnerbezeichnung 'MS OFFICE2000' muss die gesamte Pfadbezeichnung in Anführungszeichen eingeschlossen werden. Das Kommando, das mit der Anweisung X ausgeführt werden soll, wird zusätzlich in weitere (andere!) Anführungszeichen eingefügt.

Bemerkung 2.2:

Eine Dokumentation aller Funktionen, die von der Excel 4.0 Makrosprache unterstützt werden, enthält die Help-Datei MACROFUN.HLP, die auf dem Web-Server von Microsoft in gepackter Form als MACROFUN.EXE zur Verfügung steht. Visual Basic, das seit Excel 97 als Makrosprache verwendet wird, wird von Microsofts DDE nicht unterstützt.

Bemerkung 2.3:

Ein etwas einfacherer Weg, um Daten aus einer SAS-Datei in eine Excel-Datei zu exportieren, ist die Verwendung des Export Wizard von SAS. Dazu wird in SAS über die Auswahl **File** → **Export** der Export Wizard aufgerufen. Im Export Wizard wird über die Spezifizierung von **LIBRARY** und **MEMBER** die gewünschte Datei ausgewählt. Nach der Wahl von **Next** > bietet der Export Wizard die Auswahl eines Filetyps für den Export an. Wählt man hier **Comma Separated Values (*.csv)**, dann kann über **Next** > die Auswahl/Angabe eines Ordners und Dateinamens erfolgen, wo die exportierte SAS-Datei dann unter dem angegebenen Namen mit der Extension **csv** gespeichert wird. Dateien dieses Typs können von Excel direkt in eine Exceldatei umgewandelt werden.

Die unter Bemerkung 2.1 erwähnten Probleme mit fehlenden Werten in der SAS Datei treten mit dem Export Wizard nicht auf. Vorsicht ist allerdings geboten bei Datumsangaben. Während beim Transfer mittels DDE in der Anweisung PUT ein Ausgabeformat angegeben wurde, muss für den Export Wizard dieses Format permanent mit der Variablen in der Datei verknüpft werden (mit der Anweisung FORMAT im Datenschnitt oder der Prozedur DATASETS).

2.2 Datenimport in SAS aus MS Excel2000

Als Beispiel dienen wiederum die in Tabelle 1.2 abgelegten Daten in der Excel-Tabelle **data.xls** im Ordner **c:\sas\dd**, die in die SAS-Datei **data.sd2** importiert werden.

```

/* 1. Starten der Excel-Anwendung und Öffnen der Excel-Tabelle. */
options noxwait noxsync;
x "c:\programme\ms office2000\office excel.exe" ';
filename commands dde 'excel | system';
data _null_;
  file commands;
  put '[open("c:\sas\dd\data.xls")]';
run;

/* 2. Import der Daten in die SAS Datei data.sd2.*/
libname libdd 'c:\sas\dd';
filename dd dde
  'excel|c:\sas\dd\[data.xls]Tabelle1!z2s1:z5s3';
data libdd.data;
  infile dd;
  input name $ alter gewicht:commax6.2;
run;

```

Der erste Schritt, der befehlsgesteuerte Aufruf von Excel, ist natürlich nicht erforderlich, wenn die Excel-Sitzung bereits geöffnet wurde.

Analog zum Datenexport müssen hierbei Einlese-Formate explizit angegeben werden, wenn wie in obigem Beispiel numerische Werte mit Dezimalstellen oder auch Datums- und Zeitangaben eingelesen werden.

Wenn Datenwerte fehlen, dann müssen die Optionen DLM=, NOTAB, DSD und MISSEVER der Anweisung INFILE zum Einsatz kommen. Nähere Erläuterungen hierzu finden Sie in RUDOLPH/GEISSLER/ORTSEIFEN (1998).

Bemerkung 2.4:

Enthält die erste Zeile der Excel-Tabelle die für SAS gültigen Variablennamen und bei Daten, die nicht dem SAS Standard entsprechen (Datumsangaben, Worte mit mehr als 8 Zeichen ...), die erforderlichen Einleseformate:

Name \$	Alter	Gewicht:commax6.2
...

können die Werte mit folgendem Programm automatisch eingelesen werden.

```

/* 1. Zwei Makrovariablen für die Größe der Excel-Tabelle. */
%let varnum=3; * Anzahl der Spalten;
%let obsnum=5; * Anzahl der Zeilen (inkl. Variablennamen);

/* 2. Datenschnitt liest zuerst nur die erste Zeile ein zur Bestimmung der Variablennamen,
dann die Datenzeilen. */
filename in1 dde 'excel | Tabelle1!z1s1:z1sn&varnum';
data _null_;
  length n1 - n&varnum. $ 30;
  infile in1 notab dsd dlm='09'x missover;
  input (n1 -n&varnum.) ($);
  put (n1 - n&varnum. ) (=);
  array nm (*) n1 - n&varnum.;
  call execute ('filename in2 dde
                "excel | Tabelle1!z2s1:z&obsnum.s&varnum";
                " || 'data data; infile in2 notab dsd
                dlm='09'x missover; input ");
  do i=1 to n&varnum.;
    call execute ( nm(i));
  end;
  call execute (" ; run; ");
  stop;
run;

```

Bemerkung 2.5:

Eine Gegenüberstellung dieser und weiterer Möglichkeiten des Datenex- und -imports aus und in MS Excel findet man bei M.T. Mumma (1999), weitere Informationen zu DDE samt Automation bei Schreier (1999).

Datentransfer einer MS Word2000-Tabelle nach SAS

Der Transfer von Daten aus einer MS Word2000-Tabelle funktioniert völlig analog zu der in RUDOLPH/GEISSLER/ORTSEIFEN (1998), Beispiel 1.5.1, für Word97 beschriebenen Vorgehensweise. Leider werden hierbei die üblicherweise in der Kopfzeile der Tabelle (1. Tabellenzeile) stehenden Variablen nicht übernommen, sondern müssen im `input`-Statement des SAS-Programms angegeben werden. Das ist insbesondere dann ärgerlich, wenn es sich um eine größere Anzahl von Variablennamen handelt. Ein einfacher Weg, die Variablennamen aus der Word-Tabelle in das SAS-Programm zu übertragen, besteht darin, die entsprechende Tabellenzeile in Word zu markieren. Anschließend wird die markierte Tabellenzeile in durch Leerzeichen getrennten Text umgewandelt. Dieser Text kann dann einfach in das SAS-Programm kopiert werden. Dabei ist die in SAS 6.12 noch immer gültige Beschränkung auf 8 Zeichen für Variablennamen zu beachten.

Bemerkung 2.6:

Der umgekehrte Weg des Datentransfers von einer SAS-Datendatei (*.sd2) in eine MS Word2000-Tabelle lässt sich sehr einfach über den `Export Wizard` von SAS realisieren (vgl. Bemerkung 2.1). Wählt man im `Export Wizard` als Filetyp für den Export z. B. **Tab delimited file (*.txt)**, dann erfolgt über **Next** > die Auswahl/Angabe eines Ordners und Dateinamens, wo die exportierte SAS-Datei dann als Textdatei gespeichert wird. Diese Textdatei kann in ein Word2000-Dokument eingefügt werden und wegen der Tabulatortrennzeichen leicht über *Tabelle* → *Umwandeln* → *Text in Tabelle...* in eine Tabelle umgewandelt werden.

3 ODBC (Open Data Base Connectivity)

Eine allgemeine Einführung in ODBC, ODBC-fähige Anwendungen, ODBC-Treiber und ODBC-Datenquellen kann in RUDOLPH/GEISSLER/ORTSEIFEN (1998) nachgelesen werden. Hier sollen nur bestimmte mit MS Office2000 verbundene Änderungen gegenüber den dort bereits beschriebenen Vorgehensweisen aufgezeigt werden. Es wird allerdings noch einmal daran erinnert, dass die Installation von SAS Access to ODBC und des SAS-ODBC-Treibers Voraussetzung dafür ist, Daten aus SAS in andere Anwendungen zu exportieren. Im umgekehrten Fall ist die Installation des SAS-ODBC-Treibers nicht notwendig.

Der Datenaustausch zwischen SAS und anderen Windows-Anwendungen mit ODBC ist im Gegensatz zu DDE im Prinzip immer ein Import, da er stets von dem Programm initiiert wird, in das die jeweiligen Daten importiert werden sollen.

3.1 ODBC-Datenquellen-Administrator

Der mit MS Office2000 installierte ODBC-Datenquellen-Administrator (s. Abb. 3.1), der in der Systemsteuerung nicht mehr unter dem Namen 32bit-ODBC sondern unter ODBC-Datenquellen zu finden ist, hat sein Aussehen gegenüber der mit MS Office97 gelieferten Version nur geringfügig geändert.

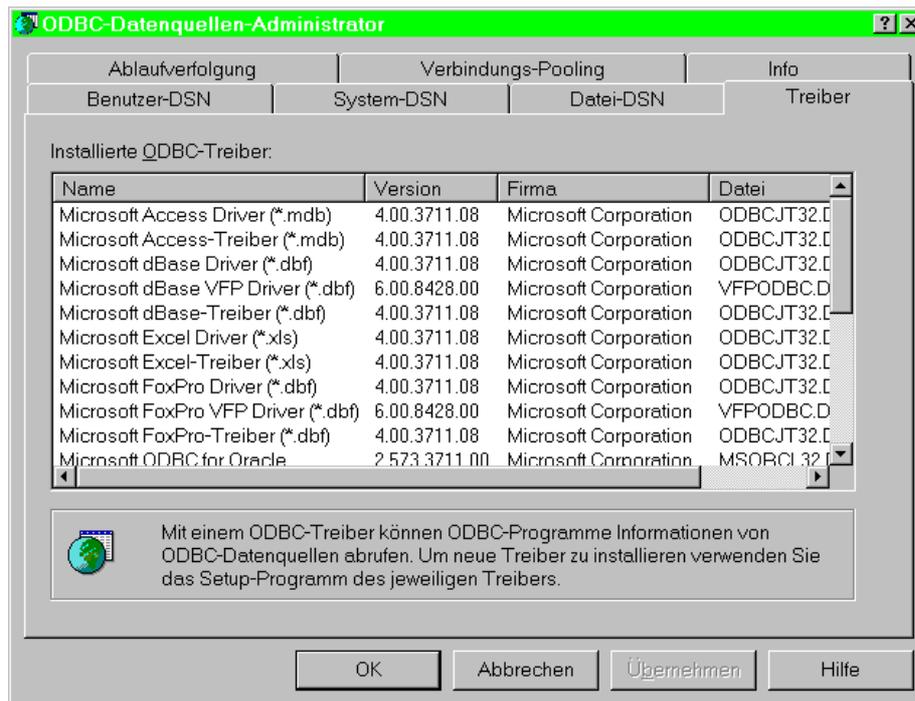


Abb. 3.1: ODBC-Datenquellen-Administrator (Treiber, MS Office 2000)

Die ehemalige Registerkarte ODBC-Treiber heißt jetzt einfach Treiber, und anstelle der Registerkarte **Protokollieren** steht jetzt die Registerkarte **Ablaufverfolgung**. Allerdings haben diese beiden Registerkarten in der zugehörigen on-line-Hilfe ihre alten Namen behalten, was zu dem Schluss führt, dass MS hier doch nicht allzuviel geändert hat. Das bestätigt sich auch, wenn man Benutzer-DSN definieren will. Hier ist alles beim Alten geblieben. Zur Definition von Benutzer-Datenquellennamen (Benutzer-DSN) sei deshalb auf die entsprechenden Beschreibungen in RUDOLPH/GEISSLER/ORTSEIFEN (1998) hingewiesen. Dass im ODBC-Datenquellen-Administrator jetzt neben den deutschen auch die englischen Bezeichnungen der Treiber stehen, ist sicher nicht besonders störend. Welche davon man jeweils für die Definition einer Datenquelle für eine bestimmte Anwendung wählt, ist unabhängig von der Funktionalität der definierten Datenquelle.

3.2 Import von SAS-Dateien in MS Access2000 und Excel2000

3.2.1 Import einer SAS-Datei in MS Access2000

Der Import einer SAS-Datei in MS Access2000 mittels ODBC unterscheidet sich kaum von der in RUDOLPH/GEISSLER/ORTSEIFEN (1998), Beispiel 2.4.1 beschriebenen Vorgehensweise. Trotzdem soll die Vorgehensweise hier noch einmal am Beispiel erläutert werden.

Zur Demonstration des Imports einer SAS-Datei in MS Access2000 wird angenommen, dass die Beispieldaten aus Tabelle 2.1 mit den Variablen `name`, `alter` und `gewicht` in der SAS-Datei `data` im Verzeichnis `c:\sas\dd` vorliegen sollen. Weiterhin sei eine SAS-Datenquelle mit dem Datenquellennamen `sasdq1` (Servername: `sasserv`, Library Name: `dd`, Host File Name: `c:\sas\dd`) definiert, und es existiere im Verzeichnis `c:\sas\dd` die (leere) Access-Datenbank `data.mdb`.

Zur Realisierung des Datentransfers der SAS-Daten aus der Datei `data` in eine Tabelle der Access-Datenbank `data.mdb` sind die folgenden Schritte notwendig:

1. Starten von MS Access2000
2. Bestehende Datenbank `c:\sas\dd\data` öffnen → **data: Datenbank**
3. **data: Datenbank: Tabellen** → *Neu* → **Neu Tabelle**
4. **Neu Tabelle: Tabelle importieren** → *OK* → **Importieren**
5. **Importieren**: Dateityp: *ODBC Databases* → **Datenquelle auswählen**
6. **Datenquelle auswählen**: Computerdatenquelle → Datenquellennamen: *sasdq1* → *OK* → (SAS startet) **Objekte importieren**
7. **Objekte importieren**: *DD.DATA* → *OK* → **DD_DATA** (als neue Tabelle in Access)
8. Beenden der ODBC-Verbindung in SAS mit der Tastenkombination **<STRG> + <Pause>** → *Stop the Server* → *OK* (SAS kann jetzt normal beendet werden.)

Bemerkung 3.1:

Eventuell wird bei dem beschriebenen Datentransfer neben den Variablen `name`, `alter` und `gewicht` auch die SAS-Systemvariable `_obstat_` in Access importiert. Diese kann jedoch problemlos wieder gelöscht werden.

3.2.2 Import einer SAS-Datei in MS Excel2000

Obwohl es sicherlich einfachere Wege gibt als den über ODBC (vgl. Abschnitt 2.1 und Bemerkung 2.1), um SAS-Dateien in MS Excel2000 zu importieren, soll der Import mittels ODBC hier doch ausführlicher dargestellt werden. Die Gründe hierfür sind, dass in RUDOLPH/GEISSLER/ORTSEIFEN (1998), Abschnitt 2.4.2 nur der Import in Excel, Version 5.0, beschrieben wurde und außerdem die dort beschriebene Vorgehensweise sich doch recht stark von der im folgenden dargestellten unterscheidet.

Um die Vorgehensweise zum Import beispielsweise der SAS-Datei `data` im Verzeichnis `c:\sas\dd` in Excel2000 mittels ODBC zu beschreiben, soll angenommen werden, dass eine SAS-Benutzer-Datenquelle mit dem Datenquellennamen `sasdq1` ohne Angaben für **Library Name** und **Host File Name** im ODBC-Datenquellen-Administrator definiert wurde. Weiterhin soll SAS über die `libname`-Anweisung `'libname dd 'c:\sas\dd';'` in einer `autoexec.sas`-Datei automatisch die so definierte library `dd` verwalten können. Die Datei `data` soll die Variablen `name`, `alter` und `gewicht` mit den Beispieldaten (s. Tabelle 2.1) enthalten.

Folgende Schritte sind dann notwendig, um die SAS-Datei `dd.data` in MS Excel2000 zu importieren:

1. Excel2000 starten. (Es wird eine neue Excel-Datei erstellt mit dem Namen Mappel.)
Daten → **Externe Daten** → **Neue Abfrage Erstellen** → (MS Query startet mit dem Fenster:) **Datenquelle auswählen**
2. **Datenquelle auswählen: Datenbanken** → **sasdq1** → **O.K.** → (SAS startet) **MS Query** mit dem zusätzlichen Fenster **Tabellen hinzufügen** (s. Abb. 3.2)

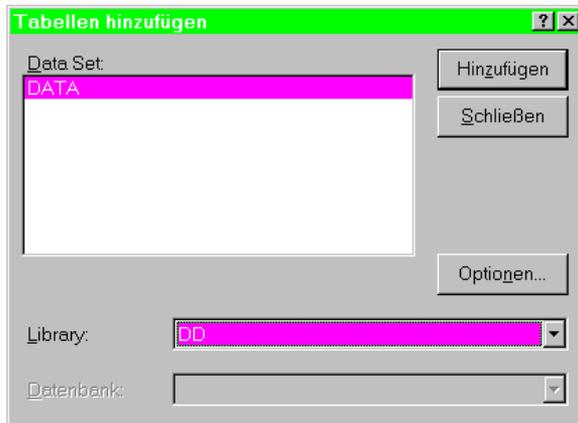


Abb. 3.2: Fenster Tabellen hinzufügen

3. **Tabellen hinzufügen**: Library: **DD** → Data Set: **DATA** → **Hinzufügen** → **Schließen** → **Microsoft Query** mit **Abfrage von sasdq1** (mit einem Fenster, das die in der Datei DATA vorhandenen Variablen anzeigt, s. Abb. 3.3)

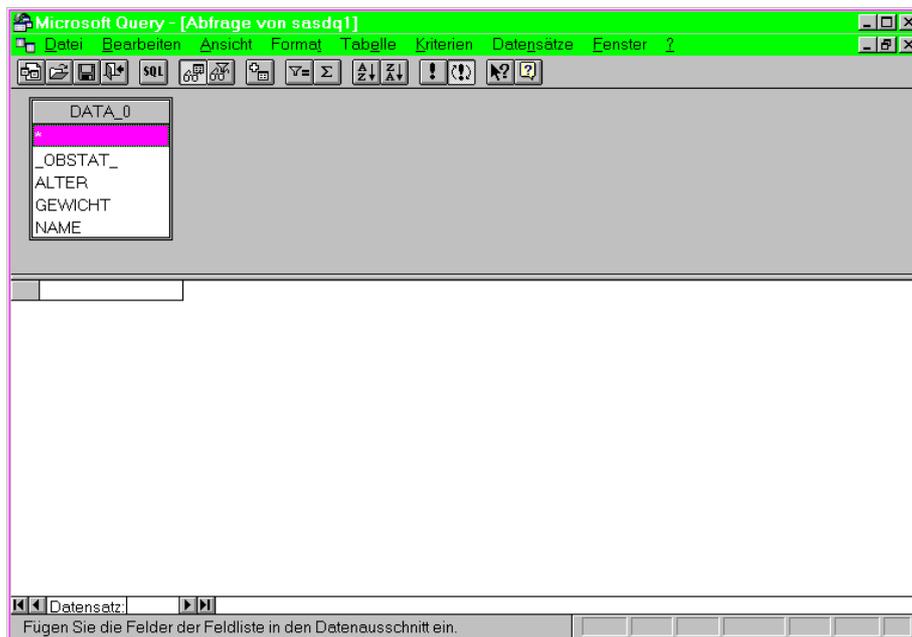


Abb. 3.3: MS Query – [Abfrage von sasdq1]

4. **Abfrage von sasdq1**: Durch Doppelklick auf die zu importierenden Variablen wird im unteren Teil des Fensters eine Tabelle aus den Daten der SAS-Datei data aufgebaut (s. Abb. 3.4). Doppelklicken von * wählt alle verfügbaren Variablen auf einmal aus. Das entspricht der Erstellung der Abfrage, wobei der entsprechende SQL-Code im Hintergrund aufbereitet wird. (Über die Schaltfläche SQL kann dieser Code auch eingesehen werden.) Durch Schließen des Fensters **MS Query** gelangt man zum Fenster **Externe Daten an Excel zurückgeben**.

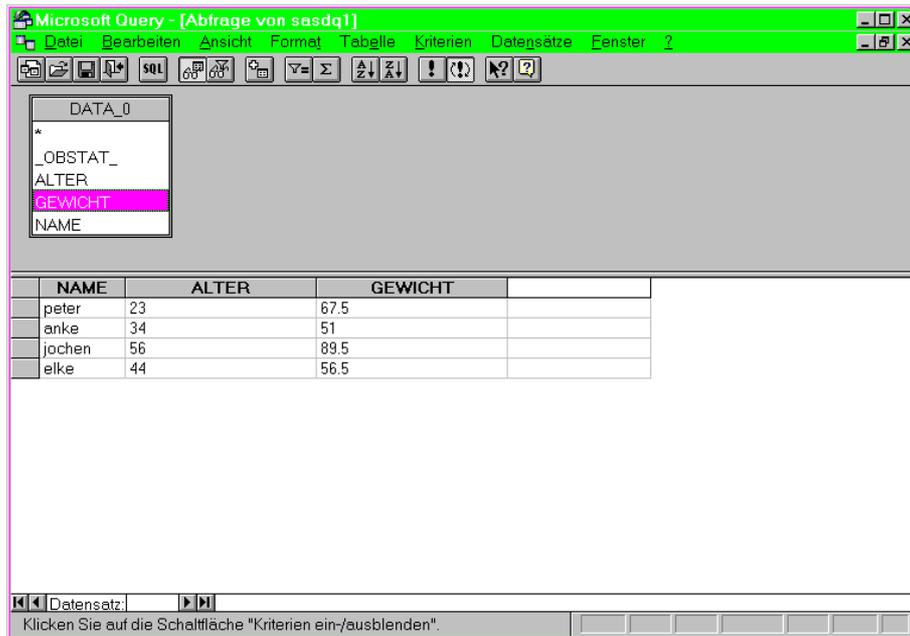


Abb. 3.4: MS Query [Abfrage von sasdq1] mit Tabelle

5. **Externe Daten an Excel zurückgeben**: Angabe von Informationen darüber, wo die zu übernehmenden Daten in der geöffneten Excel-Datei stehen sollen. → **OK** → Übernahme der Daten in Excel.

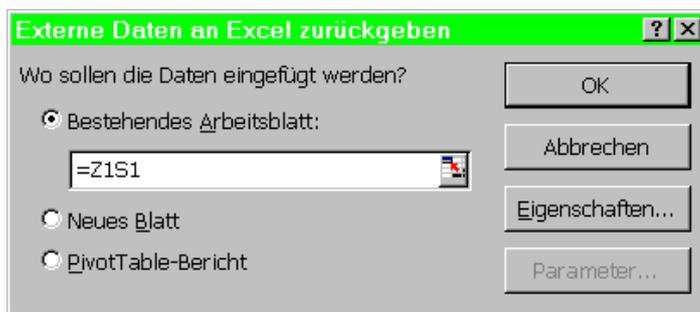


Abb. 3.5: Fenster Externe Daten an Excel zurückgeben

6. Mit <Strg> + <Pause> und **Stop the sever** → **OK** wird die Rolle von SAS als Server beendet. Danach kann SAS wie üblich beendet werden.

Bemerkung 3.2:

Die Übernahme von SAS-Daten in Excel sieht zwar zunächst etwas kompliziert aus, aber über MS Query wird hier nicht nur eine einfache Datenübernahme angeboten, sondern es können gezielt Daten aus der SAS-Datei herausgefiltert werden. Auf Einzelheiten dazu soll jedoch hier nicht näher eingegangen werden.

3.3 Import von MS Access2000- und Excel2000-Dateien in SAS

3.3.1 Import von MS Access2000-Daten in SAS

Der Import von MS Access2000-Daten in SAS funktioniert hier ebenfalls analog zu dem von MS Access97-Daten und ist bei RUDOLPH/GEISSLER/ORTSEIFEN (1998), Abschnitt 2.5, Beispiel 2.5.1 ausführlich beschrieben. Um jedoch die Unterschiede zu dem im folgenden Abschnitt beschriebenen Import von MS Excel2000-Daten in SAS aufzuzeigen, soll an dieser Stelle ein entsprechendes SAS-Programm vorgestellt werden.

Es wird angenommen, dass die MS Access2000-Datei **data.mdb** in einer Tabelle mit dem Namen **Tabelle1** die Beispieldaten (vgl. Tabelle 2.1) mit den Variablennamen **name**, **alter** und **gewicht** enthält. Diese Daten sollen in eine temporäre SAS-Datei mit dem Namen **data** importiert werden. Dazu wird außerdem vorausgesetzt, dass die Access-Datei **data.mdb** über den MS Access-ODBC-Treiber mit dem ODBC-Datenquellen-Namen **accessdq1** verknüpft ist. Die Submittierung des folgenden SAS-Programms realisiert diesen Datenimport, ohne dass MS Access gestartet wird.

SAS-Programm zum Import von Access-Daten in SAS:

```
/* 1. ODBC-Verbindung mit der Access-Datenquelle accessdq1 über proc sql */
proc sql;
    connect to odbc (dsn=accessdq1);

/* 2. Erstellen der (temporären) SAS-Datei data und einfügen der Daten aus Tabelle1 */
create table data as select * from connection to odbc
(select * from [Tabelle1]);

/* 3. Schließen der ODBC-Verbindung und beenden von proc sql */
disconnect from odbc;
quit;
```

Bemerkung 3.3:

Die Verwendung des Zeichens ***** in der in Klammern stehenden select-Anweisung bewirkt, dass alle verfügbaren Daten aus **Tabelle1** in die SAS-Datei übernommen werden. Anstelle des Zeichens ***** kann auch eine geeignete SQL-Access-Abfrage (query) zur Auswahl nur bestimmter Daten stehen. Für SAS-Anwender ohne SQL-Kenntnissen ist die *****-Lösung sicher die Vorzugsvariante.

3.3.2 Import von MS Excel2000-Daten in SAS

In SAS Technical Note TS-589B (1998) werden vier verschiedene Methoden zum Import von MS Excel-Daten in SAS beschrieben. Darunter ist auch der Import mittels ODBC, wobei als SAS-Programm ein Analogon zu dem im Abschnitt 3.3.1 beschriebenen SAS-Programm mit der (**select * ...**)-Anweisung angegeben wird. Diese *****-Lösung funktioniert allerdings beim Import von MS Excel97- und MS Excel2000-Daten in SAS nicht. Im folgenden wird an einem Beispiel beschrieben, wie dieser Import tatsächlich funktioniert.

Dazu soll wieder angenommen werden, dass in einem Arbeitsblatt mit dem Namen **Tabelle1** der MS Excel2000-Datei **data.xls** in den Zeilen 1-5 und Spalten 1-3 die Daten aus Tabelle 2.1 stehen.

Über die Systemsteuerung (*Ländereinstellungen* → *Zahlen* → *Dezimaltrennzeichen*) ist zunächst als Dezimaltrennzeichen der Punkt anstelle des Kommas einzustellen. Die Excel-Datei **data.xls** im Verzeichnis **c:\sas\dd** sei über den MS Excel-ODBC-Treiber mit dem ODBC-Datenquellen-Namen (DSN) **exceldq1** verknüpft. Zur Realisierung des Datenimports von Excel in eine temporäre SAS-Datei mit dem Namen **data** wird zunächst analog zur Vorgehensweise im Abschnitt 3.3.1 das folgende (noch unvollständige) SAS-Programm im SAS-Programm-Editor bereitgestellt.

SAS-Programm (unvollständig) zum Import von Excel-Daten in SAS

```
/* 1. ODBC-Verbindung mit der Excel-Datenquelle exceldq1 über proc sql */
proc sql;
  connect to odbc (dsn=exceldq1);

/* 2. Erstellen der SAS-Datei data und einfügen der Daten aus Tabelle1 */
  create table data as select * from connection to odbc
  (Excel-Abfrage);

/* 3. Schließen der ODBC-Verbindung und beenden von proc sql */
  disconnect from odbc;
quit;
```

Um das angegebene Programm lauffähig zu machen, ist nur der Ausdruck **Excel-Abfrage** durch den tatsächlichen SQL-Query-Code zu ersetzen. Dieser SQL-Code (Excel-Query) ist relativ leicht erhältlich. Dazu sind in Anlehnung an die im Abschnitt 3.2.2 beschriebene Vorgehensweise beim Import von SAS-Daten in Excel folgende Schritte notwendig:

1. Excel2000 starten. (Es wird eine neue Excel-Datei mit dem Namen Mappel geöffnet.) *Daten* → *Externe Daten* → *Neue Abfrage Erstellen* → (MS Query startet mit dem Fenster:) **Datenquelle auswählen**
2. **Datenquelle auswählen**: *Datenbanken* → *exceldq1* → *o.k.* → **MS Query** mit dem zusätzlichen Fenster **Tabellen hinzufügen**
3. **Tabellen hinzufügen**: *Tabelle1\$* → *Hinzufügen* → *Schließen* → **MS Query – [Abfrage von exceldq1]** (mit einem Fenster, das die in der Datei data.xls, Tabelle1 vorhandenen Variablen (Spalten) anzeigt, s. Abb. 3.6)
4. **MS Query – [Abfrage von exceldq1]**: Durch Doppelklick auf der Schaltfläche * im Feld mit den Variablennamen werden eine Tabelle mit allen Daten und im Hintergrund der SQL-Code dieser Abfrage erstellt. → *SQL* → **SQL** (s. Abb. 3.6 und 3.7)

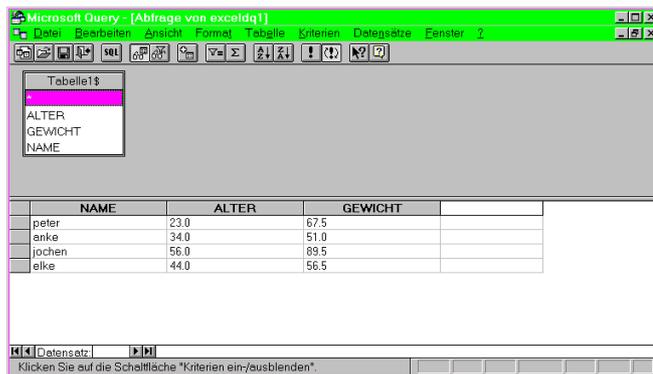


Abb. 3.6: Fenster MS Query [Abfrage von exceldq1]

5. **SQL**: Kopieren des SQL-Codes (im Beispiel: `SELECT `Tabelle1$`.NAME, `Tabelle1$`.ALTER, `Tabelle1$`.GEWICHT FROM `C:\SAS\dd\daten`. `Tabelle1$` `Tabelle1$``, s. Abb. 3.7) in die Zwischenablage → **Abbrechen** → **Microsoft Query** → **Schließen** (×) → **Externe Daten an Excel zurückgeben** → **Abbrechen** → Excel beenden (ohne Speichern)

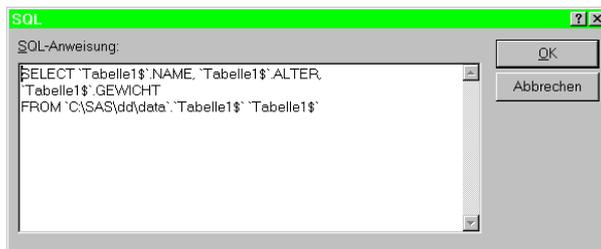


Abb. 3.7: SQL-Fenster

6. **SQL-Code** aus der Zwischenablage an die Stelle von **Excel-Abfrage** in obiges SAS-Programm kopieren.

Nach Abarbeitung dieser 6 Schritte ist das SAS-Programm lauffähig und realisiert nach Submittierung den gewünschten Datenimport.

Bemerkung 3.4:

Probleme beim Import von Excel-Daten in SAS werden sicher immer dann auftreten, wenn z.B. die SAS-Namenskonventionen in der Excel-Datei nicht berücksichtigt werden oder kompliziertere Formate übertragen werden sollen. Deshalb wird dazu geraten, derartige Daten in einer Datenbank (z. B. Access) zu halten, in der es nicht zur Mischung von Variablennamen und zugehörigen Daten bzw. unterschiedlicher Formate oder auch Datentypen wie in Excel kommen kann.

4 Literatur

- MUMMA, M.T.: The Redmond to Cary Express – A Comparison of Methods to Automatic Data Transfer Between SAS and Microsoft Excel. Beitrag der NESUG Conference, 1999. <http://www.ita.doc.gov/industry/otea/dcsug/excelsas.pdf>.
- RUDOLPH, P. E.; GEISSLER, H.; ORTSEIFEN, C.: Datenaustausch zwischen SAS und anderen Anwendungen mit DDE und ODBC. Konferenzbeiträge der 2. Konferenz für SAS-Benutzer in Forschung und Entwicklung (KSFE), Friedrich-Schiller-Univ. Jena 1998, S. 200-220.
- RUDOLPH, P. E.: ODBC zwischen SAS 6.12 und MS Access/Excel2000. Vortragsmanuskript, Dummerstorf/Rostock, 1999.
- SAS Institute Inc.: SAS Language: Reference, Version 6, First Edition, Cary, NC: SAS Institute Inc., 1990, 1042 pp.
- SAS Institute Inc.: SAS ODBC Driver Technical Report: Users Guide and Programmer's Reference, Release 6.11, Cary, NC: SAS Institute Inc., 1995. 50 pp.
- SAS Institute Inc.: SAS Companion for the Microsoft Windows Environment, Version 6, Second Edition, Cary, NC: SAS Institute Inc., 1996. 302 pp.
- SAS Institute Inc.: SAS Technical Note TS-589B, Importing Excel Files to SAS-Datasets, Cary, NC: SAS Institute Inc., 1998.
- SCHREIER, H.: Getting Started with Dynamic Data Exchange. Beitrag der NESUG Conference, 1999. <http://www.ita.doc.gov/industry/otea/dcsug/ddepaper.pdf>.