

Anhang zum Vortrag
von Joachim Bartels



www.agamedia.de

**„Migration von A2K nach A07/MySQL
+ Professionalisierung durch OCXe“**

AEK 13 – Nürnberg, Hannover

Herbst 2010

Inhalt

Einleitung	3
Diverse Tools und Infos zur MySQL-Access-Migration	3
Architektur von MySQL	5
Feature-Vergleich: MySQL mit SQL-Server	5
Kommunikation: Access ⇔ MySQL	6
Tipps zur Server-Installation	7
1. Installation des MySQL-Servers	7
2. Installation des Management-Tools SQLyog	8
3. Erstellen der Datenbank(en) auf dem Server	8
4. Anlegen von Usern auf dem Server	8
Installation des ODBC-Treibers	10
Tipps zur Migration	11
MySQL Migration Toolkit	11
SQLyog: Import External Data	14
Vergleich der Migrationsergebnisse	15
Bullzip Access To MySQL	16
Datentypen Access <=> MySQL	17
ODBC-Datenquelle (DSN) erstellen	19
MySQL-Tabellen per Hand in Access einbinden	21
Login via Access mit automatischer Verknüpfung der Tabellen (DSN-less)	22
Diverse nützliche Routinen	24
Pass-Through-Abfragen aus VBA erzeugen	24
ODBC-Error in Access abfangen	25
GetLastInsertID (mit ADOExecuteSQL)	26
ADOExecuteSQL (aus VBA direkt auf Server)	27
Datum aufbereiten für MySQL	28
MySQL-Datumsfunktionen	28
Beispiele für MySQL-SQL-Syntax (Insert, Update, Delete)	29
Beispiele: Strings mit MySQL konkatenieren (Concat, IsNull, If, As, View, CAST)	30
Systemabfragen mit MySQL-Funktionen	31
View auf dem Server erstellen	31
Professionalisierung	33
1. MapPoint-Control	33
2. ctSchedule	39
3. sevDataGrid2	40
4. Global Error Handler (vbWatchDog)	41

Einleitung

Dieser Text ist eine Anlage zu meinem Vortrag auf der *Access-Entwicklerkonferenz 13* (AEK) im Herbst 2010 in Nürnberg und Hannover. Die Vortragsfolien (Download unter www.donkarl.com) und die vorliegende Anlage sind in erster Linie für die Teilnehmer/innen dieser Vorträge bestimmt, können jedoch auch an interessierte Dritte weitergegeben werden. Ich behalte mir alle Rechte vor; (auszugsweise) Veröffentlichungen bedürfen in jedem Fall meiner vorherigen Zustimmung.

Der Text zeigt eine praktische Lösung auf, wie man mit Access (2007) über ODBC auf den MySQL-Server > 5.1.18 (InnoDB-Engine) zugreifen kann. Darüber hinaus werden viele praktische Codebeispiele präsentiert, die spezielle Lösungen für auftauchende Fragen/Probleme anbieten. Mit Hilfe dieser Informationen sollte das Zusammenspiel von Access und MySQL daher ohne größere Verwerfungen in die Tat umzusetzen sein. Sollte doch einmal etwas „haken“: Kommerzielle Supportanfragen beantworte ich gerne unter T. **05126/803303** oder per E-Mail: jbartels@agamedia.de; in allen anderen Fällen möchte ich auf die einschlägigen Foren verweisen (ich selbst bin öfter im Microsoft Office Forum anzutreffen: www.ms-office-forum.net). Alle Informationen erfolgen nach bestem Wissen, eine Haftung – egal welcher Art – kann ich nicht übernehmen.

Ich wünsche viel Spaß und Erfolg beim Ausprobieren der vorgestellten Lösungen!

Diverse Tools und Infos zur MySQL-Access-Migration

(Stand: 09/2010 – MySQL GA 5.1.50, ODBC-Connector 5.1.7)

Es gibt zahlreiche Hilfestellungen zum Thema im Internet; die wichtigsten Infos sind hier zusammengefasst und sollten einen guten Start mit anschließender Vertiefung ermöglichen.

- **Administrationstools (Auswahl):**

AnySQL Maestro: www.sqlmaestro.com

DBACentral for MySQL: www.microolap.com/products/database/mydbacentral (2007!)

dbForge Studio for MySQL: www.devart.com/dbforge/mysql/studio (erfordert .Net-Framework 2/3/3.5)

EMS SQL Manager for MySQL: www.sqlmanager.net/products/mysql

HeidiSQL: www.heidisql.com (Open Source)

MySQL-Admin: <http://dev.mysql.com/downloads/gui-tools/5.0.html> (veraltet, aber gut!)

MySQL-Enterprise-Monitor: <http://www.mysql.com/products/enterprise/monitor.html> (kostenpflichtig, ab 600 USD/Jahr in Verbindung mit Enterprise-Server-Abo)

MySQL Front: www.mysqlfront.de/wp (auf der Suche nach einer neuen Heimat ...)

MySQL-Workbench: <http://www.mysql.com/products/workbench/>

Navicat for MySQL: www.navicat.com/en/products/navicat_mysql/mysql_overview.html

SQLyog: www.webyog.com (Mein TIPP!)

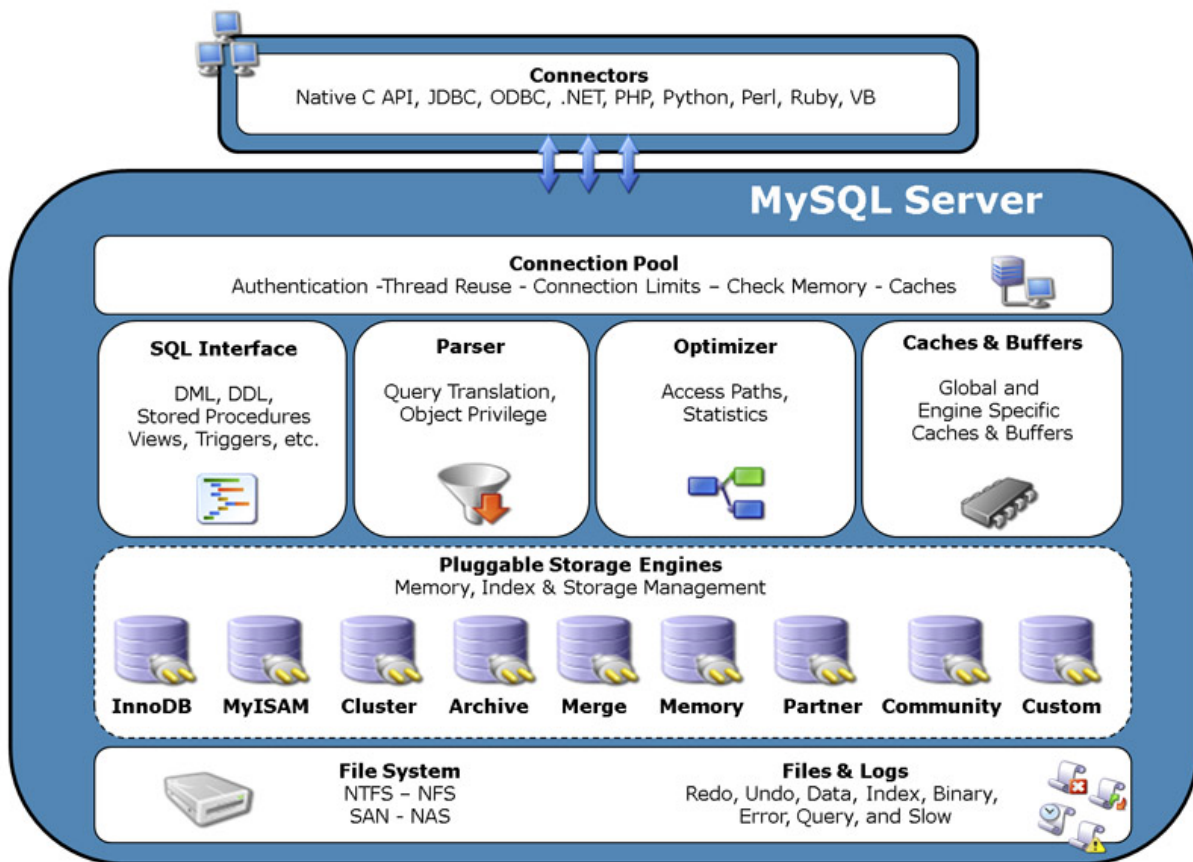
MonYog: www.webyog.com (Überwachung + Optimierung im laufenden Betrieb)

- **A Guide for Migrating From Microsoft Access to MySQL [2009]:**

Whitepaper im ZIP-File enthalten.

- **A Guide for Migrating From Microsoft Access to MySQL** [2003]:
<http://www.kitebird.com/articles/access-migrate.html>
- Sehr gute Artikelserie von Sascha Trowitzsch in *Access im Unternehmen* (2008/09):
www.access-im-unternehmen.de (Suche nach „MySQL“ im Archiv; kostenpflichtig)
- **MySQL-Dokumentation als PDF-Download**
<http://dev.mysql.com/doc/>
Kauf-Tipp: Michael Kofler: *MySQL 5 – Einführung, Programmierung, Referenz*; Addison-Wesley, mehrere Auflagen
- **GPL-Rechtsfragen:**
Die GPL kommentiert und erklärt (Einsatz von Open Source-Software allgemein aus Sicht von Rechtsanwältinnen)
<http://www.ifross.org>
- **MySQL Forums – Access:**
<http://forums.mysql.com/list.php?65>
- **Migrating From MS Access To MySQL** [2007]:
<http://dev.mysql.com/tech-resources/articles/active-grid.html>
- **Migrationstools (Auswahl):**
Bullzip (kostenlos): www.bullzip.com („Access To MySQL“; läuft nur mit MySQL Connector 5.1!)
DB-Convert for Access-MySQL: <http://www.dbconvert.com/convert-access-to-mysql-pro.php?DB=2> (79 USD; interessant, aber nicht getestet)
MySQL-Migration-Toolkit (kostenlos): <http://dev.mysql.com/downloads/gui-tools/5.0.html>
(Download enthält darüber hinaus den MySQL-Administrator sowie den QueryBrowser; erfordert JRE Java Runtime Environment ab 5.0 SP 8; Stand: 03/2009)
SQLyog (ab 139 USD): www.webyog.com
- **MySQL Migration-Toolkit-Doku:**
<http://dev.mysql.com/doc/migration-toolkit/en/index.html> (Chapter 13: Preparing a Microsoft Access Database for Migration)
- **MySQL-ODBC-Treiber-Download:**
5.1.7: <http://dev.mysql.com/downloads/connector/odbc/>
3.51.27: <http://dev.mysql.com/downloads/connector/odbc/3.51.html#downloads>
- **MySQL-Server-Download:**
<http://dev.mysql.com/downloads/mysql/5.1.html>
- **MySQL-Upgrade: Tipps & Tricks**
<http://lstigile.wordpress.com/2009/06/15/mysql-windows-install-troubleshooting/>
- **Visueller Leitfaden zu Microsoft Access-Front-Ends mit MySQL** [2010]:
Whitepaper im ZIP-File enthalten.
- **Why Move to MySQL from Microsoft SQL Server?**
http://dev.mysql.com/tech-resources/articles/move_from_microsoft_sql_server.html
(Interessanter Feature-Vergleich, tabellarischer Auszug s. folgende Seite)
- **Darüber hinaus finden sich zahlreiche White-Papers zum Download auf der Website von MySQL** (bspw. zur Replikation unter Windows, Performance-Optimierung, Sicherheitsaspekte usw.)

Architektur von MySQL



Feature-Vergleich: MySQL mit SQL-Server

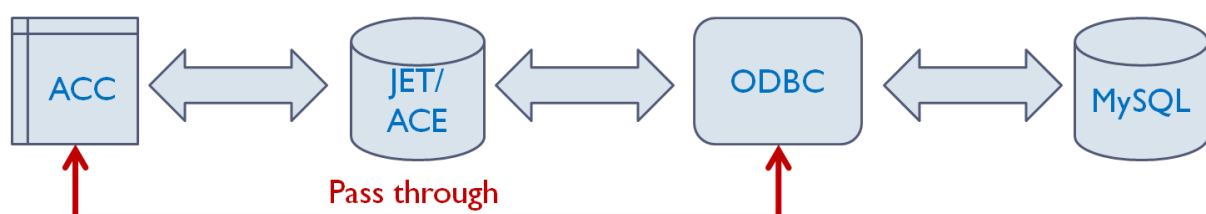
Database Feature	MySQL	SQL-Server
<i>(Kommentare beziehen sich nur auf MySQL)</i>		
Open Source <i>(wenn nicht kommerziell genutzt ...)</i>	X	
Available on two-dozen platforms (32 and 64 bit): RedHat, SuSE, Fedora, Solaris, HPUS, AIX, SCO, FreeBSD, Mac OS, Windows	X	
Pluggable Storage Engine Architecture (MyISAM, InnoDB , etc.) <i>(ab 5.1)</i>	X	
High-Availability Clustered Database	X	X
ANSI SQL , SubQueries , Joins , Cursors, Prepared Statements	X	X
Stored Procedures, Triggers, SQL and User-Defined Functions	X	X
Updateable Views	X	X
ACID Transactions with Commit, Rollback	X	X
Distributed Transactions	X	X
Row-level Locking	X	X
Snapshot/Consistent Repeatable Reads (readers don't block writers and vice-versa)	X	X

Server-enforced Referential Integrity	X	X
Strong Data type support (Numeric, VARCHAR, BLOB, etc)	X	X
High-Precision Numeric Data types	X	X
Indexing (clustered, b-tree, hash, full-text) (<i>Volltext nur MyISAM</i>)	X	X
Dynamic Memory Caches	X	X
Unique Query Cache (stores query and result set)	X	
Cost-Based Optimizer	X	X
Unicode, UTF-8	X	X
XML, XPath	X	X
Geospatial support (<i>na, ja ... vgl. Kofler in databasePro 5/10</i>)	X	X
Replication (<i>ab 5.1 Row based und für Cluster</i>)	X	X
Table and index Partitioning (<i>ab 5.1 nur Zeilen, nicht Spalten</i>)	X	X
VLDB (terabytes) capable (= <i>Very large databases</i>)	X	X
High-speed, data load utility	X	X
Online Backup with Point-in-Time Recovery (<i>kostenpflichtig!</i>)	X	X
Automatic Restart/Crash Recovery	X	X
Automatic Storage Management (auto-expansion, rollback management)	X	X
Compressed and Archive Tables	X	X
Information Schema/Data Dictionary	X	X
Security (GRANT/REVOKE, SSL, fine grained object privileges)	X	X
Built-in data encryption and decryption	X	X
Built-in Task Scheduler (<i>ab 5.1</i>)	X	X
Drivers (ODBC , JDBC, .NET, PHP, etc) (<i>ebenfalls unter GPL</i>)	X	X
GUI management and development tools	X	X

Quelle (Auszug): Why Move to MySQL from Microsoft SQL Server?

http://dev.mysql.com/tech-resources/articles/move_from_microsoft_SQL_Server.html

Kommunikation: Access ⇔ MySQL



Access nutzt einen ODBC-Treiber von MySQL, um via Jet/Ace mit dem Server zu kommunizieren. Unter Nutzung sog. Pass-Through-Abfragen (ADO/OLE DB) kann der Server auch direkt in seiner Syntax (ANSI SQL) abgefragt werden.

Tipps zur Server-Installation

1. Installation des MySQL-Servers

Zunächst laden Sie den Server unter der oben angegebenen Adresse in der gewünschten Version (32-/64-Bit, Betriebssystem u.a.: ich empfehle die vollständige MSI-Variante mit ca. 115 MB) herunter. Bei der Installation gehen Sie bitte in folgender Reihenfolge vor:

- Wählen Sie den Server (oder den Entwicklungsrechner) aus, auf dem die Datenbank installiert werden soll und erstellen dort den Ordner **..\MySQL**.
- Starten Sie die Datei MySQLxxx.msi (wobei xxx für die jeweilige Versionsangabe steht) mit Admin-Rechten.
- Wählen Sie unter „Setup Type“ den Eintrag „Custom“ (nur hierdurch können Sie im weiteren Verlauf den gewünschten Server bzw. den Zielordner festlegen)
- Klicken Sie im „Custom Setup“ auf „MySQL-Server“ und dann auf den Button „Change“: Geben Sie anschließend den gewünschten Ordner auf dem Server als Ziel an.
- Klicken Sie danach auf „MySQL Server Datafiles“ und dann auf den Button „Change“: Auch hier geben Sie bitte den gewünschten Ordner auf dem Server als Ziel an. Sie können hier ebenfalls \MySQL wählen, ein \Data-Ordner wird automatisch hinzugefügt.
- Klicken Sie dann auf „Next“ und starten die Installation des Servers.
- Nach kurzer Zeit erscheint ein Werbe-Dialog, den Sie bitte mit „Next“ fortsetzen, bis der Eintrag „Configure the MySQL Server now“ auftaucht. Mit Auswahl dieser Option klicken Sie anschließend auf „Finish“.

Im Folgenden wird der Server mit einigen Einstellungen konfiguriert:

Ja - *Detailed Configuration*

Ja - Entweder *Developer Machine* (während der Entwicklung) oder *Server Machine* bzw.

Dedicated MySQL Server Machine (für den produktiven Einsatz; je nach Ressourcen und zu erwartender Last).

Die folgenden Optionen können Sie so wie vorgeschlagen übernehmen: lassen Sie dabei unbedingt die Option *Standard Character Set* (Latin1) aktiviert und klicken die Option *Include Bin Directory in Windows Path* an (erleichtert das spätere Backup mit MySQLDump). **Wichtig** ist, dass Sie sich das Root-Passwort notieren, sonst haben Sie später keinen Zugriff auf den Server!

- Nach Abschluss der Installation beenden Sie zunächst den Server-Dienst über Systemsteuerung/Verwaltung/Dienste/MySQL „beenden“
- Erstellen Sie dann eine Sicherheitskopie der Datei My.ini, die Sie im definierten Ordner für MySQL auf dem Server finden.
- Editieren Sie dann die Datei My.ini, ergänzen unter der Rubrik [mysqld] den Eintrag **max_allowed_packet=16M**
Der Eintrag ist Voraussetzung für den späteren reibungslosen Import größerer Feldinhalte (Blobs, Memos) aus einem Dump; der Standardwert von 1M ist deutlich zu klein.
Als Entwickler sollten Sie noch folgende Einträge ergänzen (Erläuterung dazu weiter unten):
character-set-server=latin1
collation-server= latin1_german2_ci
Speichern Sie die Datei My.ini abschließend.

Ggf. könnten auch die beiden folgenden Einträge interessant sein (ab Version 5.1.38!), die eine verbesserte InnoDB-Version als Plug-In aktivieren (soll ab etwa 16 User performanter sein; es gibt weitere Parameter hierzu, die eine Feintuning des Plug-Ins erlauben):

ignore-builtin-innodb

plugin-load=innodb=ha_innodb_plugin.so

- Starten Sie nun den Serverdienst erneut:
Systemsteuerung/Verwaltung/Dienste/MySQL „neu starten“
- Mit dieser Grundkonfiguration kommt man auch im produktiven Einsatz gut zurecht; für weiteres Tuning seien die Advisory-Wizards von MonYog (s. Links) sowie einschlägige Handbücher empfohlen: Michael Kofler (s. Links) und „High Performance MySQL – Optimization, Backups, Replication, and more.“ von Baron Schwartz, Peter Zaitsev et al. O'Reilly, verschiedene Auflagen.
- Um die Parameter des laufenden Servers abzufragen, verwendet man „SHOW VARIABLES;“ (ohne „“) als Abfrage direkt auf dem Server (z.B. mit SQLyog; liefert rund 275 Parameter).
- Zur Abfrage des aktuellen Status auf dem Server verwendet man „SHOW STATUS;“ (ohne „“).


2. Installation des Management-Tools SQLyog

- Installieren Sie bei Bedarf das Hilfsprogramm SQLyog auf Ihrem Server/Entwicklungsrechner (es kann als 30-Tage-Testversion von webyog.com bezogen werden). Alternativ können Sie die von den MySQL GUI-Tools installierten Verwaltungstools nutzen.

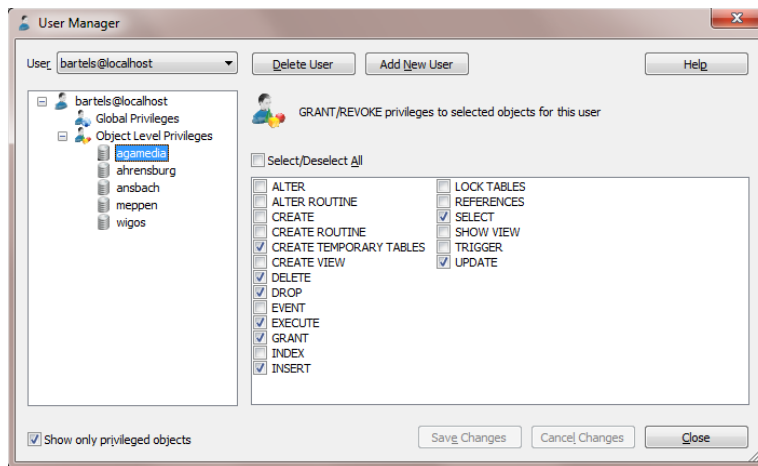
3. Erstellen der Datenbank(en) auf dem Server

- Starten Sie bspw. SQLyog und erstellen Sie eine neue Verbindung zum SQL-Server: Der Name der „Connection“ sollte dem Namen Ihrer Datenbank entsprechen. Beim ersten Start existiert nur der User „root“. Ergänzen Sie das notierte Root-Passwort und klicken Sie auf „Connect“ (die Verbindungsdaten können gespeichert werden).
- Klicken Sie in der Mitte der Symbolleiste auf das Symbol „Restore from SQL Dump“, wählen eine vorhandene Dump-Datei aus und starten abschließend den Import. Nach erfolgreichem Import nutzen Sie in der Symbolleiste den bunten Refresh-Button, um das Ergebnis Ihres Imports zu sehen.
- Alternativ nutzen Sie die im Vortrag vorgestellten Migrationstools, eine DB zu erzeugen.

4. Anlegen von Usern auf dem Server

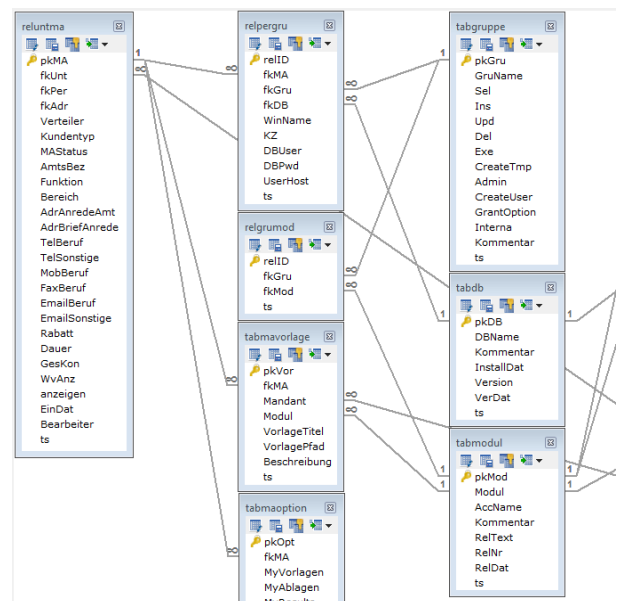
- Klicken Sie in SQLyog auf den Button „User Manager“  (STRG+U) und wählen „Add New User“.
- Erstellen Sie den gewünschten User (Kleinbuchstaben!) auf dem gewünschten Host, womit der Rechner des Users gemeint ist, von dem aus der Zugriff auf den Server erfolgt: z.B. agamedia@10.1.% (aus Sicherheitsgründen sollte der Zugriff auf den Server möglichst nur von wenigen Rechnern, nicht von „%“ generell erfolgen) oder bei Entwicklern localhost bzw. die IP 127.0.0.1 (entspricht localhost). Vergeben Sie anschließend das Passwort (Kleinbuchstaben!). Erzeugen Sie dann den User mit Hilfe des Buttons „Create“.
- Im Anschluss wählen Sie den User mit Hilfe der gleichnamigen Combobox links oben aus.

- Wenn Sie einen neuen ADMIN anlegen, dann sollten Sie diesem Globale Rechte auf den Server geben (Global Privileges). „Normalen“ Usern, die später über den Access-Client auf MySQL zugreifen, weisen Sie nur die einzelnen Rechte pro Datenbank zu (Object Level Privileges), indem Sie zuerst die gewünschte Datenbank auf dem Server markieren und anschließend die folgenden Rechte zuweisen:
 - **SELECT, INSERT, UPDATE, DELETE**
 - ggf. GRANT, DROP (für Userverwaltung auf Client), CREATE TEMPORARY TABLES, EXECUTE
 In seltenen Fällen kann es erforderlich sein, weitere Rechte an einzelnen Tabellen, Views, SPs u.a. zu definieren, worunter allerdings oft die Übersichtlichkeit und letztlich auch die Performance leidet.



- Aus Sicherheitsgründen sollten Sie unbedingt einen neuen ADMIN-User (z.B. myadmin) anlegen, dem Sie sämtliche globalen Rechte zuordnen. Löschen Sie abschließend den alten ADMIN „root“.
- Es empfiehlt sich, mit einem angelegten Standard-User (nicht mit dem Admin) in der Entwicklung zu arbeiten, so dass Sie von vornherein fehlende Rechte erkennen können.

Der Hauptkritikpunkt am Usersystem von MySQL ist die fehlende Gruppenzuordnung: die Rechte müssen für jeden User/DB einzeln definiert werden. Eine einfache Lösung besteht jedoch darin, die Gruppenzugehörigkeit über eigene Tabellen abzubilden (bspw. wie in der nebenstehenden Abbildung mit den beiden Tabellen *tabGruppe* – *relPerGruppe* zu sehen) und darüber den User und sein Set an Rechten auf dem Server via CREATE USER, GRANT, REVOKE, DROP USER aus Access heraus anzulegen bzw. zu verwalten (weitere Informationen zu den genannten Kommandos finden Sie im MySQL-Benutzerhandbuch -> Links).



Installation des ODBC-Treibers

Es gibt zwei Versionen des MySQL-ODBC-Treibers für Windows (32- u. 64-Bit), die im Marketing-sprech aktuell „MySQL Connectors“ heißen: die ältere Version 3.51 und die neuere Version 5.1 mit Unicode-Unterstützung, beide unter GPL oder im Rahmen der MySQL-OEM-Lizenzierung erhältlich (Download-Quellen s. oben). 5.1 ist für den Einsatz auf Win-64-Systemen optimiert und kann als Standard eingesetzt werden. Der 3.51-Treiber ist ausgereift und hat sich im Zusammenspiel mit Access 2007/ACE bewährt, wenn Sie mit Latin1 arbeiten. Weitere Details zu den Treibern können Sie dem folgenden Auszug aus dem MySQL-Handbuch entnehmen:

Connector/ODBC 3.51 is the current release of the 32-bit ODBC driver, also known as the MySQL ODBC 3.51 driver. Connector/ODBC 3.51 has support for ODBC 3.5x specification level 1 (complete core API + level 2 features) in order to continue to provide all functionality of ODBC for accessing MySQL.

Connector/ODBC 5.1 also includes the following changes and improvements over the 3.51 release:

- *Improved support on Windows 64-bit platforms.*
- *Full Unicode support at the driver level. This includes support for the [SQL_WCHAR](#) datatype, and support for Unicode login, password and DSN configurations. For more information, see [Microsoft Knowledgebase Article #716246](#).*
- *Support for the [SQL_NUMERIC_STRUCT](#) datatype, which provides easier access to the precise definition of numeric values. For more information, see [Microsoft Knowledgebase Article #714556](#)*
- *Native Windows setup library. This replaces the Qt library based interface for configuring DSN information within the ODBC Data Sources application.*
- *Support for the ODBC descriptor, which improves the handling and metadata of columns and parameter data. For more information, see [Microsoft Knowledgebase Article #716339](#).*

Die Installation des ODBC-Treibers ist denkbar einfach: sie wird (nach der Installation des Servers) unter Admin-Rechten und mit dem Parameter „Complete“ in etwa einer Minute durchgeführt.

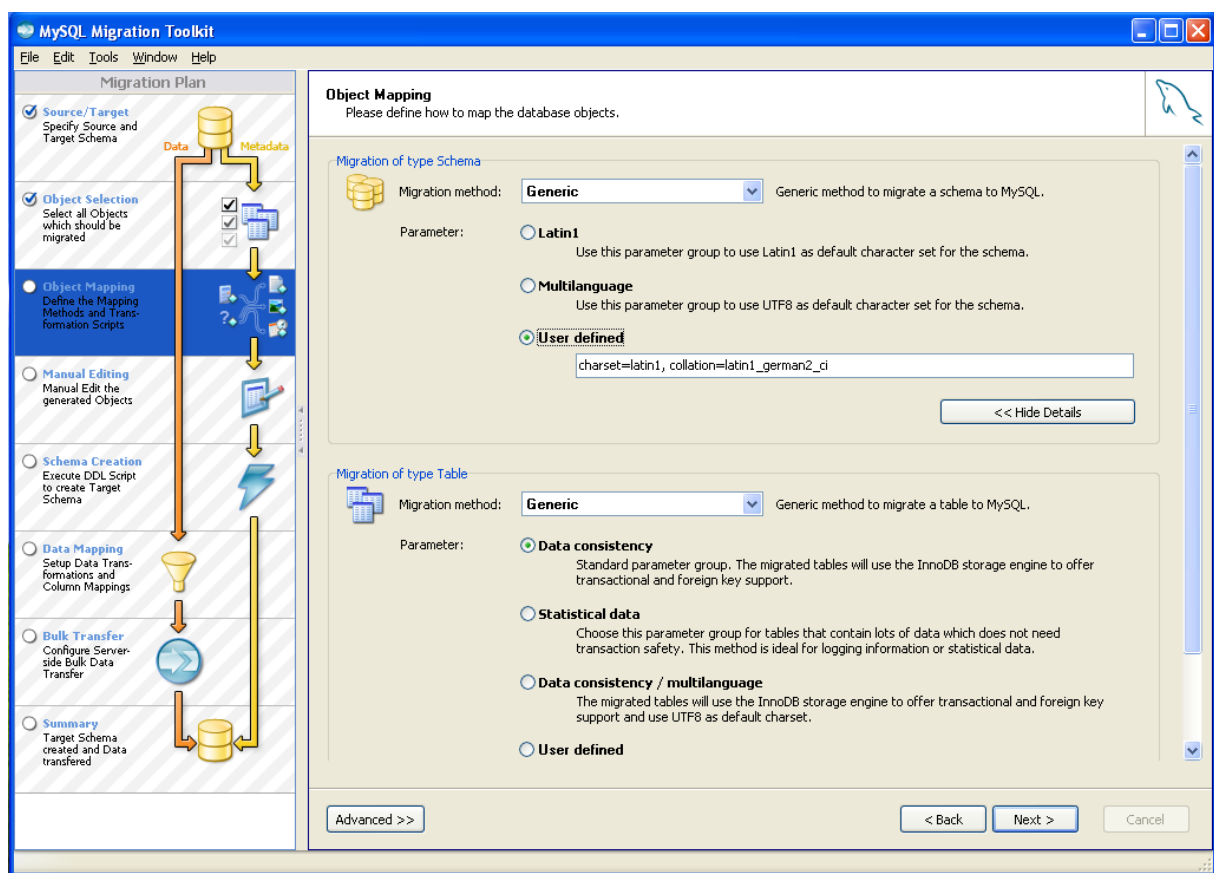
Um während der Entwicklung ggf. mit unterschiedlichen MySQL-Datenbanken arbeiten zu können, empfiehlt sich die Anlage einer ODBC-Datenquelle (sog. DSN = Data Source Name), die die jeweiligen Verbindungsdaten dauerhaft speichert (mehr dazu s. Inhaltsverzeichnis).

Tipps zur Migration

MySQL Migration Toolkit

Bei der Migration einer Access-DB mit dem MySQL Migration Toolkit kann einiges schiefgehen, was man jedoch teilweise durch geeignete Konfiguration im Migrationsdialog verhindern kann. Um später eine Access-konforme Suche und Sortierung mit den Daten vom MySQL-Server durchführen zu können, benötigt man unbedingt die Kombination `charset=latin1, collation=latin1_german2_ci` für die Tabellen der betreffenden Datenbank (collation_database). Die Standard-Einstellung bei `charset=latin1` ist übrigens `collation= latin1_swedish_ci`, was ebenso wie das oft empfohlene `utf8` zu unerwünschten Ergebnissen bei der Sortierung führt! (Suche `ä*`; gefunden wird `ä* UND a*`).

Im Migrationsdialog sollte man also im Schritt „Object Mapping“ die entsprechenden Parameter bei „User defined“ setzen, nachdem man diese zuvor mit „Show Details“ eingblendet hat:



Bei bereits bestehenden Datenbanken kann man sich mit Hilfe der beiden Kommandos „SHOW VARIABLES LIKE 'character_set%';“ und „SHOW VARIABLES LIKE 'collation%';“ einen Überblick über die aktuellen Einstellungen des Servers bzw. der Datenbank verschaffen (wichtig ist `collation_database`):

Variable_name	Value
<input type="checkbox"/> collation_connection	utf8_general_ci
<input checked="" type="checkbox"/> collation_database	latin1_german2_ci
<input type="checkbox"/> collation_server	latin1_swedish_ci

Für die Datenbank (character_set_database) ist der Zeichensatz latin1 wichtig; die übrigen Werte können anders lauten (mehr zu diesem komplexen Thema findet sich im Kap. 10 des Handbuchs):

Variable_name	Value
<input type="checkbox"/> character_set_client	utf8
<input type="checkbox"/> character_set_connection	utf8
<input checked="" type="checkbox"/> character_set_database	latin1
<input type="checkbox"/> character_set_filesystem	binary
<input type="checkbox"/> character_set_results	utf8
<input type="checkbox"/> character_set_server	latin1
<input type="checkbox"/> character_set_system	utf8
<input type="checkbox"/> character_sets_dir	C:\AGAMedia\MySQL\share\charsets\

Falls der Zug diesbezüglich „schon abgefahren ist“, besteht die Möglichkeit, im Nachhinein mittels VBA nachzuhelfen, was ebenfalls nicht ganz ohne ist (vgl. Kommentar im Code):

```
Public Sub AlterCollation(strMySQLDB As String)
' *****
' Sub: AlterCollation
' Parameter:
' Autor: ST/AGA
' Erstellt am: 25.02.2009
' Geändert am: 25.02.2009
' Funktion: Aufruf: AlterCollation("daten")
' Funktion ändert die Collation für Datenbanken, die nicht die korrekten Einstellungen haben.
' Sie produziert ein SQL-Skript, das dann in MySQL ausgeführt wird. Übergeben wird der Name der MySQL-DB,
' ausgeführt wird sie innerhalb der MDB, die bereits alle verknüpften ODBC-Tabellen enthält.
' ACHTUNG: Löscht Kommentare und NotNull-Werte der Spalten (MySQL 5.1.30)!
' *****

Dim tdf As TableDef
Dim dbs As Database
Dim fld As DAO.Field
Dim strSingle As String
Dim strAlter As String
Dim i As Long

Set dbs = CurrentDb

'Fremdschlüsselprüfung ausschalten
strAlter = "SET FOREIGN_KEY_CHECKS = 0;" & vbCrLf

For Each tdf In dbs.TableDefs
If Len(tdf.Connect) > 0 Then
If Left(tdf.Connect, 4) = "ODBC" Then
strSingle = "ALTER TABLE `" & strMySQLDB & "`." & tdf.Name & "` CHARACTER SET latin1 COLLATE lat-
in1_german2_ci;"
strAlter = strAlter & strSingle & vbCrLf
strSingle = "ALTER TABLE `" & strMySQLDB & "`." & tdf.Name & "` "
i = 0
For Each fld In tdf.Fields
Select Case fld.type
Case dbText
strSingle = strSingle & "MODIFY COLUMN `" & fld.Name & "` VARCHAR(" & fld.size & ") CHARACTER SET latin1
COLLATE latin1_german2_ci, "
i = i + 1
Case dbMemo
strSingle = strSingle & "MODIFY COLUMN `" & fld.Name & "` TEXT CHARACTER SET latin1 COLLATE lat-
in1_german2_ci, "
```

```
        i = i + 1
    End Select
Next fld
If i > 0 Then
    strSingle = left(strSingle, Len(strSingle) - 2) & ";"
    strAlter = strAlter & strSingle
End If
End If
End If
Next tdf

'Fremdschlüsselprüfung einschalten
strAlter = strAlter & vbNewLine & "SET FOREIGN_KEY_CHECKS = 1;"

Dim f As Integer
f = FreeFile
Open CurrentProject.Path & "\altercollation.sql" For Binary As f
Put f, strAlter
Close f

Set tdf = Nothing
Set dbs = Nothing

End Sub
```

Eine (bessere) Alternative hierzu besteht im guten alten „Suchen&Ersetzen“-Dialog, den man auf einen Dump anwenden kann, der im Anschluss erneut nach MySQL importiert wird. Hierbei wird z.B. der Eintrag „CHARSET=latin1“ oder „CHARSET=utf8“ ersetzt durch „CHARSET=latin1 COLLATE=latin1_german2_ci“.

Um sicherzustellen, dass der Server für alle neuen Datenbanken und Tabellen gleich von Beginn an die richtigen Sprach-/Sortiereinstellungen verwendet, sollten Sie in der My.ini unter [mysqld] die folgenden Einträge ergänzen:

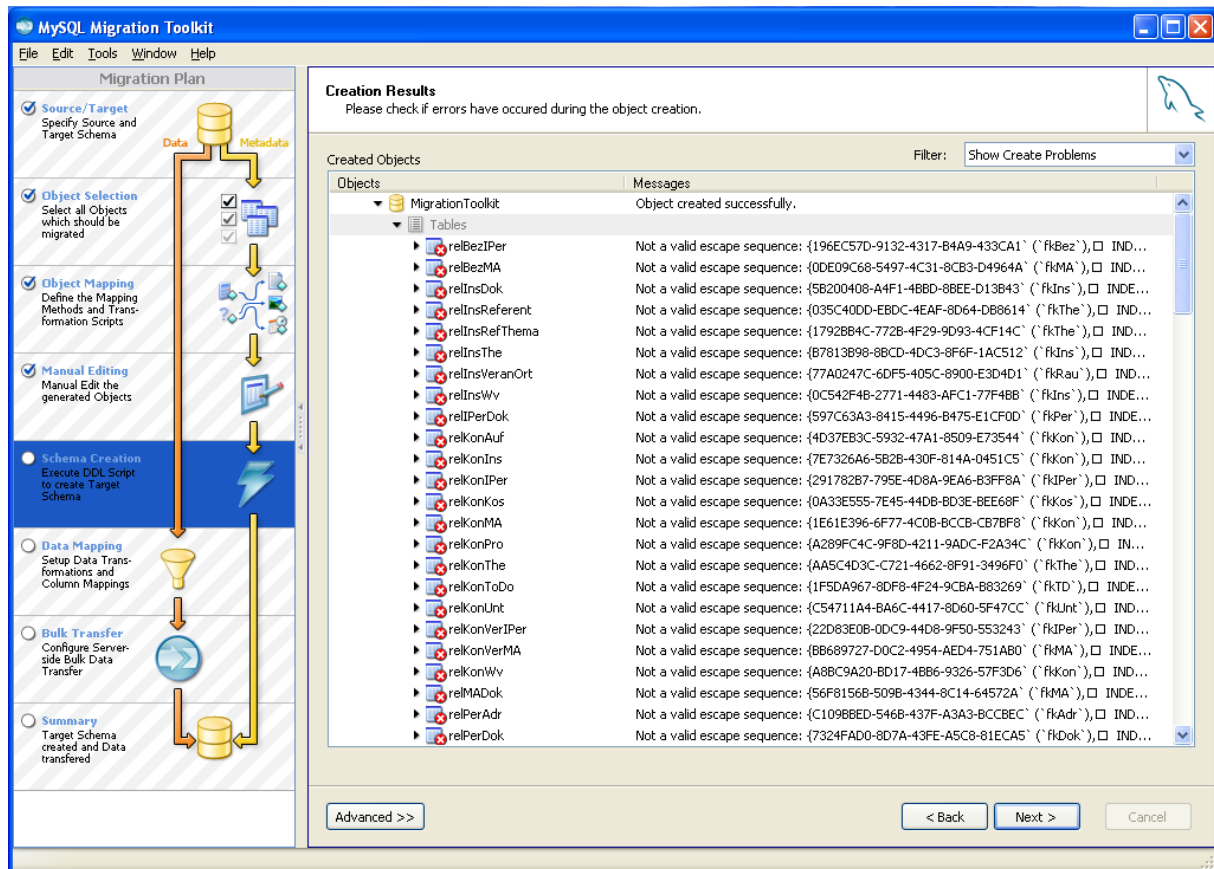
```
character-set-server=latin1
collation-server= latin1_german2_ci
```

Wichtig: Die neuen Einstellungen werden erst nach einem Neustart des Servers gültig!

Das **zweite Problem** mit dem Migration-Toolkit lässt sich leider nicht beheben. Zunächst einmal empfiehlt die Hilfe zum Migration Toolkit, die Systemtabellen der zu importierenden Access DB sichtbar zu machen und explizit für den Admin freizugeben (Benutzer-, nicht Gruppenrechte!). Tut man dies brav, dann unterbleiben in der Folge Fehlermeldungen, die man über den jeweiligen „Advanced“-Button sichten kann, aber im Ergebnis wurden von den 198 Tabellen der Test-DB ganze 82 (!) übertragen – nämlich genau die, die nicht über Constraints verknüpft waren. Gibt man die Systemtabellen nicht vorher frei, dann werden zwar alle Tabellen übertragen –jedoch wie im ersten Fall ohne Constraints.

Der Grund hierfür liegt in folgender Problematik begründet: Wenn man in Access im Beziehungsfenster zwei Tabellen miteinander verknüpft, dann resultiert daraus ein Eintrag für den Constraint in MSysRelationships.SzRelationship, der bspw. „tabInsMediumrelSemTechnik“ (1:n) heißt. So weit, so gut, diesen Eintrag könnte das Migration Toolkit vermutlich sauber verarbeiten. Wenn jedoch die Datenbank in eine neue Access-DB importiert wird – was ja guter Stil vor einer Auslieferung ist – dann werden die bis dahin „sprechenden“ Constraints in GUID-ähnliche Gebilde verwandelt, z.B.

„{035C40DD-EBDC-4EAF-8D64-DB861472FD60}“. Dieser String ist zum einen zu lang für das Toolkit, zum anderen gibt es Probleme mit der Java-RT, die in bezug auf die geschweiften Klammern eine „ungültige Escape-Sequenz“ findet (weil die schließende Klammer fehlt):



Die gleiche Problematik besteht mit den definierten Access-Indizes – auch diese werden zu aussage-losen GUID-Strings verwandelt. Das Problem habe ich bereits 2008 an SUN/MySQL gemeldet, ohne dass die zugesicherte Fehlerbereinigung erfolgt ist. Wer sich also mit dem Migration Toolkit an die Migration einer Access DB macht, der fährt besser damit, die Systemtabellen nicht freizugeben und im Nachgang die Beziehungen mit sprechenden Bezeichnungen zu rekonstruieren – per Hand!

SQLyog: Import External Data

Auch hier ist es erforderlich, die Systemtabellen in Access zuvor freizugeben. Für die Migration einer Access DB nach MySQL startet man SQLyog (ab Vers. Enterprise), legt eine leere DB in MySQL an und ruft anschließend den Wizard für den Import auf (Powertools/Import External Data). Zur Auswahl als Datenquellen stehen Access-DBs, Excel-Dateien, CSVs oder beliebige ODBC-Datenquellen.

Wie zuvor gibt es einige Probleme, so wird etwa der Zeichensatz der Tabellen ungefragt und ohne Einflussmöglichkeit auf utf8 mit der Collation (Sortierung) utf8_general_ci festgelegt, was zu den oben beschriebenen Problemen führt (Lösungsansätze dafür finden sich ebenfalls im vorherigen Kapitel). Feld-Kommentare bleiben im Gegensatz um Migration Toolkit jedoch erhalten. Mit Hilfe der in diesem Kapitel vorgestellten Tipps kommt man mit SQLyog bei einer Access-Migration am weitesten.

Vergleich der Migrationsergebnisse

Der Vergleich einer Access-Tabelle aus der Quell-DB

tabPerson : Tabelle			
	Feldname	Felddatentyp	
	pkPer	AutoWert	
	Kundentyp	Text	Gründer, Seminarist
	Status	Text	Status der betreute
	Zuständig	Text	Zuständiger (Haupt-
	Anrede	Text	Herrn, Frau, Firma
	Briefanrede	Text	Briefanrede
	Titel	Text	Akademische oder A
	Name	Text	Nachname der Perso
	Vorname	Text	Vorname der Person
	RZusatz	Text	ggf. weitere Infos z
	Rabatt	Zahl	Grund für Rabatt
	Jahrgang	Zahl	Geburtsjahr
	KfmKenntnisse	Text	Umfang der kaufmänn
	Idee	Text	vorläufige Geschäft
	Kontakt von	Datum/Uhrzeit	Erstkontakt
	Kontakt bis	Datum/Uhrzeit	letzter Kontakt
	telefonisch	Zahl	Anzahl telefonischer
	persoenlich	Zahl	Anzahl persönlicher
	schriftlich	Zahl	Anzahl schriftlicher I
	Interview	Zahl	Anzahl Interviews
	Dauer	Zahl	Gesamtdauer aller K
	WvAnz	Zahl	Anzahl der aktiven \
	WarnStat	Text	Besondere Hinweise
	Verzeichnis	Memo	Kundenspezifischer
	SemKal	Ja/Nein	Seminarkalender
	Eingabedatum	Datum/Uhrzeit	
	Korrekturdatum	Datum/Uhrzeit	

als Importergebnis zwischen den Migrationskandidaten macht die Unterschiede schnell deutlich. Man beachte nachfolgend die grundsätzlich fehlenden Defaultwerte, aber auch das leicht unterschiedliche Mapping der Datentypen bei „Kontakt von“ (Datum/Uhrzeit) „SemKal“ (J/N) oder „Verzeichnis“ (Memo): jedes Programm hat so seine Eigenarten, die per Hand korrigiert werden müssen.

Alter Table 'tabperson' in 'migrationtoolkit'													
	Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?	Auto Incr?	ZeroFill?	Charset	Collation	Comment	
*	pkPer	int	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
	Kundentyp	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	latin1	latin1_swedish_ci		
	Status	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	latin1	latin1_swedish_ci		
	Zuständig	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	latin1	latin1_swedish_ci		
	Anrede	varchar	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	latin1	latin1_swedish_ci		
	Briefanrede	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	latin1	latin1_swedish_ci		
	Titel	varchar	25		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	latin1	latin1_swedish_ci		
	Name	varchar	35		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	latin1	latin1_swedish_ci		
	Vorname	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	latin1	latin1_swedish_ci		
	RZusatz	varchar	100		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	latin1	latin1_swedish_ci		
	Rabatt	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	Jahrgang	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	KfmKenntnisse	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	latin1	latin1_swedish_ci		
	Idee	varchar	100		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	latin1	latin1_swedish_ci		
	Kontakt von	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	Kontakt bis	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	telefonisch	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	persoenlich	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	schriftlich	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	Interview	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	Dauer	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	WvAnz	smallint	5		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	WarnStat	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	latin1	latin1_swedish_ci		
	Verzeichnis	longtext			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	latin1	latin1_swedish_ci		
	SemKal	tinyint	1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	Eingabedatum	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	Korrekturdatum	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				

Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Charset	Collation	Comment
* pkPer	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Kudentyp	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	utf8	utf8_general_ci	Gründer, Seminari:
Status	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	utf8	utf8_general_ci	Status der betreu:
Zuständig	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	utf8	utf8_general_ci	Zuständiger (Haupt
Anrede	varchar	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	utf8	utf8_general_ci	Herrn, Frau, Firm
Briefanrede	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	utf8	utf8_general_ci	Briefanrede
Titel	varchar	25		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	utf8	utf8_general_ci	Akademische oder
Name	varchar	35		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	utf8	utf8_general_ci	Nachname der Perso
Vorname	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	utf8	utf8_general_ci	Vorname der Perso
RZusatz	varchar	100		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	utf8	utf8_general_ci	ggf. weitere Info:
Rabatt	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Grund für Rabatt
Jahrgang	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Geburtsjahr
KfzKenntnisse	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	utf8	utf8_general_ci	Umfang der kaufm
Idee	varchar	100		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	utf8	utf8_general_ci	vorläufige Geschä
Kontakt von	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Erstkontakt
Kontakt bis	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			letzter Kontakt
telefonisch	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Anzahl telefonisch
persoenlich	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Anzahl persönlich
schriftlich	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Anzahl schriftlich
Interview	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Anzahl Interviews
Dauer	int	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Gesamtdauer aller
WvAnz	smallint	5		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Anzahl der aktive
WarnStat	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	utf8	utf8_general_ci	Besondere Hinweis
Verzeichnis	text			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	utf8	utf8_general_ci	Kundenspezifische
SemKal	bit	1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Seminarkalender
Eingabedatum	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
Korrekturdatum	timestamp			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			

Bullzip Access To MySQL

Mit **Bullzip** wurde die Personentabelle aufgrund von Problemen mit mehreren Datumsfeldern und Memo-Defaultwerten erst gar nicht übernommen – das Tool ist mit aktuellem Stand daher nicht zu empfehlen!

Datentypen Access <=> MySQL

Bei der Migration von Access nach MySQL stellt sich die Frage, wie die einzelnen Datentypen von Access in MySQL am sinnvollsten abgebildet werden. Die nachstehende Tabelle gibt dazu Auskunft.

Access	MySQL	Kommentar
Autowert (Long Integer) Not Null Index: J Duplikate: N	INT (10) AUTO_INCREMENT NOT NULL PRIMARY KEY	Mit Acc als Client wird für <u>jede</u> MySQL Tabelle ein PK benötigt (auch zusammengesetzte Schlüssel sind ok)! PKs können aktualisiert werden, wenn UNIQUE gewährleistet bleibt (bei händischem Import wichtig!)
Text (max. 255 Z) Null zulässig Duplikate: J/N Index: J/N	CHAR (n) fixe Länge, max. 255 Z VARCHAR (n) flexible Länge, max. 65.535 Z	CHAR(0) ist zulässig (Werte NULL oder "") <u>Achtung</u> : „Silent Column Changes“: VARCHAR(<4)=>CHAR(<4); CHAR(>3)=>VARCHAR(>3) Wichtig: Latin1 + Latin1_german2_ci (ä=ae, ö=oe;ü=ue;ß=ss); ohne Angabe gilt der Default-Zeichensatz/Sortierung von Tabelle > Datenbank > Server. MySQL Toolkit-Migration liefert utf8 + utf8_general_ci und nutzt ausschließlich VARCHAR -> Korrektur empfohlen!
Memo (max. 65.535 Z) Null zulässig Default zulässig Index: N	VARCHAR (n) flexible Länge, max. 65.535 Z und zwar als <u>Summe aller VARCHARs</u> einer Tabelle! TEXT (reiner Text) flexible Länge, max. 65.535 Z MEDIUMTEXT (z.B. RTF) (Längen der TEXT-Varianten s.u. BLOB)	Für VARCHAR muss bereits bei Deklaration die maximale Länge vorgegeben werden; für TEXT sind <u>keine</u> Längenbegrenzung oder <u>Defaults</u> möglich Memo wird bei Migration zu LONGTEXT gewandelt -> korrigieren! Bei Import-Problemen mit TEXT oder MEDIUMTEXT: den Parameter max_allowed_packet in My.ini prüfen und hochsetzen: 16M
Byte (ganze Z.) 0 bis 255	TINYINT , Unsigned Wichtig: NULL führt zu Problemen; immer mit „0“ oder „-1“ als Default ausstatten	Ohne Unsigned nur -127 bis 128
Integer (ganze Z.) -32.768 bis 32.767 Index: J/N	SMALLINT (2 Byte)	
Long Integer (ganze Z.) -2.147.483.648 bis 2.147.483.647 Index: J/N	INT / INTEGER (4 Byte)	In Acc für Fremdschlüssel genutzt: fk Not Null <u>MySQL</u> : INT = INTEGER Verwendung von INT Unsigned in Acc nicht empfohlen
Single (Gleitkommawerte) -3,4 x 10 ³⁸ bis 3,4 x 10 ³⁸	FLOAT (4 Byte)) DOUBLE nutzen!	In Acc für Dezimalzahlen genutzt wird durch Migration in DOUBLE(7,2) gewandelt

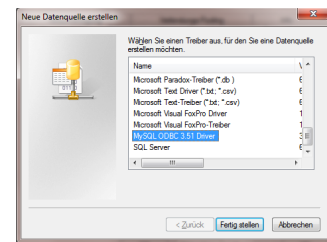
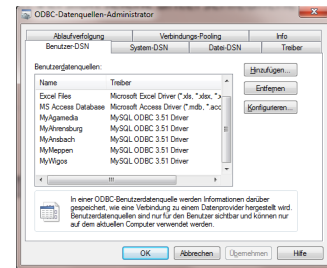
Double (Gleitkommawerte) -1,797 x 10 ³⁰⁸ bis 1,797 x 10 ³⁰⁸ (bis 15 Nachkommastellen)	DOUBLE (8 Byte) Vorkommastellen >= Nachkommastellen Double 7,2 -> 7 Stellen insgesamt: 5 vor, 2 nach dem Komma	<u>Kofler</u> : Double statt Float verwenden REAL = DOUBLE GEO-Koordinaten: Double 10,6
Dezimal (numerische Werte) -9,999... x 10 ²⁷ bis 9,999... x 10 ²⁷	DECIMAL	
Datum/Uhrzeit	<u>DATETIME</u> (YYYY-MM-TT hh:mm:ss) >01.01.1000 <31.12.9999 <u>TIMESTAMP</u> >1970 <2038 0000-00-00 00:00:00 auf Server zulässig; wird in Access ignoriert (NULL)	<u>MySQL</u> : 1 Timestamp-Feld pro Tabelle ist für Access unbedingt erforderlich: ts, CURRENT_TIMESTAMP, NOT NULL. Das Feld ist nach Einrichtung noch NULL, d.h. über den Query Browser (nicht über Acc-Abfrage!) oder über Pass Through kann das Feld gefüllt werden (z.B. Wert: 20080101000000) <u>Kofler</u> : DateTime statt Date verwenden; eigene Datums-/Uhrzeit-Angaben in DateTime speichern, NICHT in Timestamp (dient der internen Verwaltung) <u>Tipp</u> für Anzeige: In Abfrage die Funktion DateValue (dt. DatWert) verwenden: <i>DatumOhneUhrzeit: DatWert([MyDate])</i>
Währung	DECIMAL(19,4) Speicherung der Zahlen als Zeichen	<u>MySQL</u> : Dezimalzahl mit vier Nachkommastellen; für Währungen empfohlen <u>Access</u> : Im Entwurf der verknüpften (!) Tabelle unter Format "Euro" auswählen, dann wird der Wert korrekt mit € angezeigt NUMERIC, DEC = DECIMAL
Ja/Nein (Boole) 0 = Nein -1 = Ja Index: J/N	TINYINT(1) (BOOL : NULL = Nein, NOT NULL = Ja) Default = 0!	<u>Access</u> : Ja = -1; <u>MySQL</u> : Ja = 1 BIT nicht für Boolean verwenden TINYINT wird zu NOT NULL konvertiert -> Korrigieren!
OLE-Objekt	Medium-BLOB	Tinyblob = 255 B Blob = 64 KB Mediumblob = 16 MB Longblob = 4 GB (Jet max. 2 GB!)
Hyperlink	TEXT flexible Länge, max. 65.535 Z	Keine Entsprechung in MySQL
[Keine Entsprechung!]	MEDIUMINT (24 Bit) BIGINT (64 Bit) LONGTEXT (4 GB – Jet max. 2 GB!) ENUM SET GEOMETRY-Objekte	<u>Kofler</u> : Bigints, Sets und Enums meiden!
<u>Defaultwerte</u>	Datums-Defaults nur bei Timestamps möglich:	Defaults gehen bei der Migration verloren!

	<p>CURRENT_TIMESTAMP; bei DateTime kann man sich mit einem Codeschnipsel helfen, der das aktuelle Datum beim Anlegen setzt. Defaults immer mit NOT NULL kombinieren!</p>	<p><u>Anderes Verhalten:</u> Defaults werden erst beim Speichern auf dem Server gesetzt; kann durch Feld-Standardwerte im Formular wie bei Access simuliert werden</p>															
<u>Indizes</u>	<p>Indizes können über 1 oder mehrere Spalten festgelegt werden (optional UNIQUE). Bei n:m-Verknüpfungstabellen sind zur Erstellung von Constraints <u>zwei</u> Indizes erforderlich (idx_fkUnt_fkAdr ist allein nicht ausreichend, sondern erfordert noch idx_fkAdr; s. Abbildung). Bei Textspalten (CHAR, VARCHAR, TEXT, BLOB u.a.) kann die Länge der Strings, die für den Index genutzt werden soll, festgelegt werden (NULL = Gesamter Inhalt) <u>Wichtig:</u> InnoDB-Tabellen unterstützen (bisher; das ändert sich evt. in naher Zukunft) keine FULLTEXT-Indizes!</p>	<p>Tabellen mit mehreren UNIQUE-Indizes werden in Access nach dem alphabetisch obersten sortiert angezeigt <u>Empfehlung:</u> Sprechende Benennung mit Präfix „idx_...“</p> <table border="1"> <thead> <tr> <th>Indexes</th><th>Columns</th><th>Index Type</th></tr> </thead> <tbody> <tr> <td>* PRIMARY</td><td>relID</td><td>Unique</td></tr> <tr> <td>idx_fkUnt_fkAdr</td><td>fkUnt, fkAdr</td><td>Unique</td></tr> <tr> <td>idx_fkAdr</td><td>fkAdr</td><td></td></tr> <tr> <td>idx_AdrTyp</td><td>AdrTyp</td><td></td></tr> </tbody> </table>	Indexes	Columns	Index Type	* PRIMARY	relID	Unique	idx_fkUnt_fkAdr	fkUnt, fkAdr	Unique	idx_fkAdr	fkAdr		idx_AdrTyp	AdrTyp	
Indexes	Columns	Index Type															
* PRIMARY	relID	Unique															
idx_fkUnt_fkAdr	fkUnt, fkAdr	Unique															
idx_fkAdr	fkAdr																
idx_AdrTyp	AdrTyp																
<u>Benennungskonventionen</u>	<p>Feldnamen sollten keine Leerzeichen, Umlaute oder Trennstriche enthalten (Probleme beim Dump-Import!) Weitere Probleme tauchen auf, wenn SQL-Befehle als Feldnamen verwendet wurden.</p>	<p>Vor Umbenennungen von Feldern müssen ggf. vorhandene Constraints und dann die Indizes (für diese Felder) gelöscht werden! Nach Umbenennung das Ganze wieder aufbauen: erst Index, dann Constraint.</p>															

ODBC-Datenquelle (DSN) erstellen

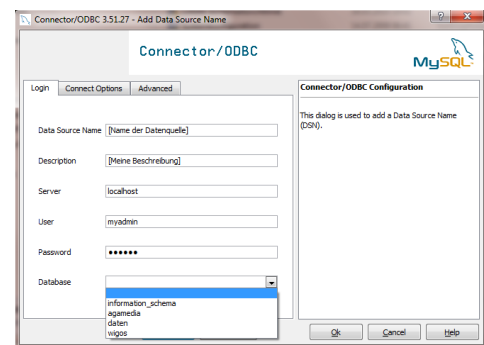
Eine ODBC-Datenquelle (DSN) dient der dauerhaften Speicherung von Verbindungsinformationen. Insbesondere für ein Frontend, das während der Entwicklung mit unterschiedlichen MySQL-Backends auf dem Server verknüpft werden soll, sind DSNs sehr hilfreich. DSNs werden am besten in der Variante „Benutzer-DSN“ nach Installation eines MySQL-ODBC-Treibers (s. an anderer Stelle) in *Systemsteuerung / Verwaltung / ODBC-Datenquellen* mit Admin-Rechten erstellt. Zunächst soll hier eine Anleitung zur Erstellung und Parametrierung von Benutzer-DSNs vorgestellt werden; weiter hinten wird die „Königsklasse“ DSN-lose-Verbindung mit Code-Beispielen vorgestellt.

1. Rufen Sie unter Windows den Menüpunkt *Systemsteuerung / Verwaltung / ODBC-Datenquellen* auf (je nach Version leicht unterschiedliche Benennungen!)
2. Mit Start des ODBC-Datenquellen-Administrators erhalten Sie mehrere Möglichkeiten, eine DSN zu erstellen, die sich lediglich im Ort der Speicherung der Verbindungsinformationen unterscheiden.
3. Wählen Sie den Reiter „Benutzer-DSN“ und dann „Hinzufügen“.
4. Im nächsten Dialogschritt wählen Sie den installierten ODBC-Treiber von MySQL aus und klicken auf den etwas seltsam formulierten Button „Fertig stellen“, woraufhin der eigentliche Dialog zur Parametrierung der ODBC-Datenquelle erscheint.
5. Füllen Sie den Dialog über die einzelnen Reiter wie folgt aus (ersetzen Sie dabei die Einträge in „[]“ durch eigene Angaben).

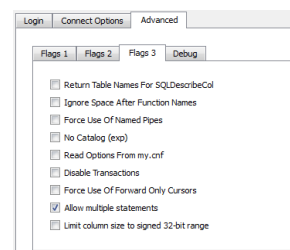
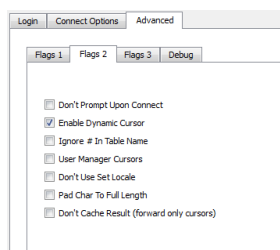
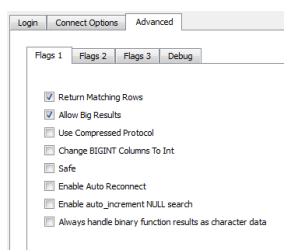


Anstelle von „localhost“ können Sie auch eine konkrete IP-Adresse angeben. Als User können Sie Ihren neu angelegten Admin-Account nutzen (root sollte ja, wie erwähnt, aus Sicherheitsgründen gelöscht werden).

Wenn alles richtig konfiguriert ist, dann werden schon beim Öffnen der Combobox „Database“ die auf dem Server vorhandenen Datenbanken (information_schema und mysql sind interne Datenbanken von MySQL) angezeigt. Wählen Sie eine vorhandene Datenbank aus. Über den Button „Test“ (in der Abbildung unter der geöffneten Combobox) können Sie prüfen, ob die Verbindung aufgebaut wird.



6. Klicken Sie anschließend auf den Reiter „Connect Options“. Hier können Sie ggf. einen anderen TCP/IP-Port angeben, wenn der Standard-Port 3306 belegt ist. Alle übrigen Einstellungen erfolgen im Reiter „Advanced“ wie folgt (Empfehlung bewährter Einstellungen):



7. Im Anschluss speichern Sie die erstellte Datenquelle mit Klick auf den Button „OK“.

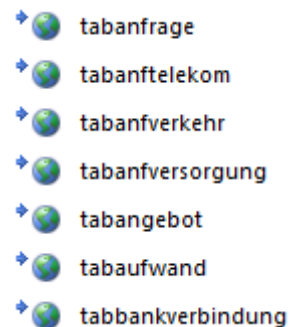
Bleibe noch zu erwähnen, dass die Summe der verwendeten Parameter den Wert **67108906** ergibt. Dieser Wert sollte auch als Konstante für Connectionsstrings bspw. in VBA verwendet werden (der öfter genannte Parameterwert 43 macht keinen Sinn mehr, wenn der ODBC-Treiber > 3.51.17 ist).

Zukünftig können Sie in Access die Tabellen der ausgewählten Datenbank verknüpfen, indem Sie im Dialog den vergebenen Namen für die Datenquelle aufrufen. Wie das geht, zeigt der folgende Abschnitt.

MySQL-Tabellen per Hand in Access einbinden

Damit nach der ganzen Vorrede/-arbeit nun endlich die Entwicklung in Access starten kann, müssen lediglich noch die Tabellen der Datenbank vom MySQL-Server eingebunden werden. Es gibt hierzu zwei Möglichkeiten, die sich bewährt haben. Solange noch kein ausgefeilter Dialog zur DB-Auswahl und automatischen Aktualisierung der Tabellenverknüpfungen besteht, können Sie die Tabellen über einen Wizard in Access einbinden. Hierzu ist eine bereits angelegte ODBC-Datenquelle sehr hilfreich (s.o.), die die Verbindungsparameter dauerhaft speichert. Um die Tabellen in einer leeren Access-DB einzubinden, wählen Sie

- in Access 2007: **Externe Daten / Weitere / ODBC-Datenbank.** Klicken Sie anschließend auf „Erstellen Sie eine Verknüpfung zur Datenquelle, ...“ und „OK“
- Wählen Sie im auftauchenden Dialog die zuvor angelegte Computerdatenquelle (DSN) aus und klicken dann „OK“
- Wählen Sie alle gewünschten Tabellen der Datenbank aus und starten den Verknüpfungsprozess wiederum mit „OK“
- Die derart über ODBC verknüpften Tabellen werden mit einem speziellen Symbol im Objektbereich angezeigt (s. Abbildung).



Für alle anderen Access-Versionen verfahren Sie analog bzw. nach den Angaben der Onlinehilfe.

Bei der Arbeit mit MySQL-Tabellen in Access sind zwei wichtige Punkte zu beachten:

- Nach jeder **Änderung** an einer Tabelle auf dem Server muss die zugehörige Tabellenverknüpfung in Access gelöscht und auf dem oben beschriebenen Weg neu eingebunden werden. Access erstellt bei der Verknüpfung nicht nur den Connectionstring, sondern speichert zahlreiche weitere Informationen, die zur Zusammenarbeit **JET/ACE <-> ODBC <-> MySQL** erforderlich sind. Auch für diese schnelle Neuverknüpfung ist eine DSN sehr hilfreich.
- Wollen Sie während der Entwicklung die **Access-Objekte** in eine leere Datenbank **importieren**, muss die Verknüpfung der Tabellen unbedingt vor dem Import der anderen Access-Objekte erfolgen, ansonsten reagiert Access „verschnupft“ und alle Abfragen sind unbrauchbar!

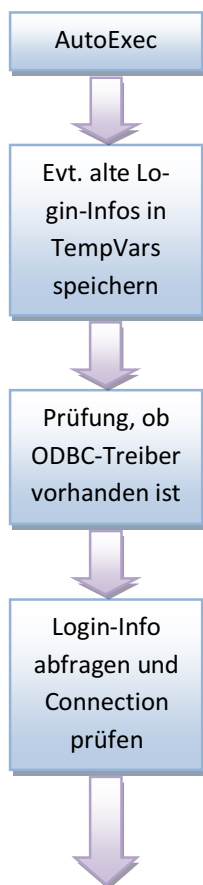
Ansonsten war's das schon – Access arbeitet in fast allen Fällen mit dem MySQL-Backend wie mit einem MDB/ACCDB-Backend zusammen. Einige Beispiele, wo es nicht 1:1 funktioniert, habe ich im Vortrag vorgestellt und problematisiert.

Natürlich ist für den produktiven Einsatz beim Kunden einiges mehr an Komfort erforderlich, z.B. ein Access-Login für MySQL, ggf. Lizenzprüfungen sowie das Aktualisieren der verknüpften Tabellen und Pass-Through-Abfragen (auch diese haben einen Connectionstring, der gepflegt sein will). Wie das funktioniert, beschreibt der folgende Abschnitt.

Login via Access mit automatischer Verknüpfung der Tabellen (DSN-less)

Die für mich zum Beginn der Migration größte Herausforderung war der Login via Access am MySQL-Server. Denn neben der reinen Anmeldung eines Users am Server war die Auswahl einer (von mehreren) Datenbank(en), die Lizenzprüfung (steht freie Lizenz für bestimmtes Modul zur Verfügung?), das Login-Protokoll, die Aktualisierung von Tabellenverknüpfungen und Pass-Through-Abfragen (PT) – um nur die wichtigsten Punkte zu nennen –, von Bedeutung.

Dies soll im Folgenden aus Übersichtsgründen jedoch nur eine untergeordnete Rolle spielen, wir werden uns auf den Login am Server und die Neuverknüpfung von Tabellen und PTs konzentrieren. Hierbei hat mir die MySQL-Artikelserie von Sascha Trowitzsch in „Access im Unternehmen“ sehr weitergeholfen (s. Links). Auf dieser Grundlage habe ich die folgenden Routinen entwickelt/verfeinert. An dieser Stelle kann ich natürlich nur eine Übersicht über den Ablauf geben, den vollständigen Code finden Sie im Modul modODBC.bas gespeichert im ZIP-File (die Bereitstellung erfolgt mit freundlicher Erlaubnis von Sascha Trowitzsch, Berlin; Achtung: TempVars gibt es erst ab Acc2007!).



Der Login startet mit dem AutoExec-Makro von Access, das die AutoExec-Funktion aufruft. Es folgen zunächst diverse Prüfungen, die hier keine Rolle spielen sollen. Dann werden die vorherigen Login-Infos aus einer gesicherten, lokalen MDB ausgelesen und als TempVars gespeichert (gehen bei Error nicht verloren).

```

Function AutoExec()
' *****
' Funktion: Startprozedur der Datenbank (Aufruf durch AutoExec-Makro)
' *****

'TempVars-Auflistung initialisieren
'(liest u.a. LastDB, LastServer, LastPort aus)
If InitTempVars = False Then
    SendStatus ("ERROR - TempVars nicht initialisiert!")
    ShutDownDB
Else
    SendStatus "OK - TempVars sind initialisiert ..."
End If
    
```

Anschließend wird geprüft, ob ein geeigneter MySQL-ODBC-Treiber auf dem Client installiert ist (SendStatus ist eine Routine zur Protokollierung des Login-Vorgangs):

```

'ODBC-Treiber vorhanden?
If CheckDriver = False Then
    SendStatus ("ERROR - ODBC-Treiber fehlt!")
    ShutDownDB
Else
    SendStatus "OK - ODBC-Treiber ist vorhanden ..."
End If
    
```

Bei positiver Prüfung wird der ODBC-Connectionstring ermittelt
*Driver={MySQL ODBC 3.51 Driver};;DATABASE=agamedia;SERVER=127.0.0.1;
 PORT=3306;UID=bartels;PWD=geheim;OPTION=67108906;*
 und eine Test-Verbindung zum Server aufgebaut. Hat das funktioniert, dann wird der Anmeldedialog präsentiert, der nur noch die Eingabe des PWs erfordert (der Connectionstring wird ebenfalls als TempVar gespeichert).



'Connection zum Server prüfen und Relinking der Tabellen/PT-Abfragen


```

If StartDB = False Then
    SendStatus ("ERROR - Keine Verbindung zum Server!")
    ShutDownDB
Else 'alles in Ordnung
    'Login protokollieren
    If SaveLogin("Modul", GetTVar("tvarLastUser"), GetTVar("tvarLastPwd"), GetT-
Var("tvarLastDB"), GetTVar("tvarLastServer"), GetTVar("tvarLastPort")) = True Then
        SendStatus "OK - Login am Server erfolgreich ..."
    End If
End If

```



Verknüpfte
Tabellen und
PTs prüfen
und neu
verlinken

In der Routine RefreshLinks werden die unterschiedlichen Tabellen geprüft (Tabellen mit den Präfixen hl-, wz- und xx- sind lokal verknüpfte ACCDB-Tabellen) und je nach Präfix weiterverarbeitet; nur die ODBC-Tabellen – zu erkennen am „ODBC“-String im Connectionstring – werden mit dem MySQL-Server verbunden. Im Anschluss werden auch die vorhandenen Pass-Through-Abfragen in gleicher Weise bearbeitet (s. folgender Auszug).

```

'Passthrough-Abfragen neu verlinken
For Each qdf In CurrentDbC.QueryDefs
    'läuft die abf über ODBC?
    If left(qdf.Connect, 4) = "ODBC" Then
        'ist der Connectionstring auf den aktuellen User eingestellt?
        If (qdf.Connect <> strBase) Or ForceRefresh Then
            qdf.Connect = strBase
        End If
        qdf.Close
        If Err <> 0 Then 'Err = 3011, wenn Tabelle nicht gefunden
            RefreshLinks = False
            MsgBox "Fehler bei PT-Abfrage " & qdf.Name & " in Modul " & conModulName & "!" &
vbNewLine _
                & "Bitte kontaktieren Sie den Hersteller.", vbCritical, "preConSulT 3"
            Exit Function
        End If
    End If
Next qdf

```

Dieser Vorgang dauert je nach Netzwerkdurchsatz (100 MBit) weniger als 20 Sekunden bei rund 360 Tabellen- und rund 25 PT-Abfragen.

Im Code finden sich auch sonst noch einige Besonderheiten, wie bspw. der Einsatz von *CurrentDBC*, auf die hier nicht weiter eingegangen werden kann – bei Interesse finden Sie weitere Infos in alten AEK-Skripten oder auch in Internet-Foren.

Diverse nützliche Routinen

Pass-Through-Abfragen aus VBA erzeugen

```
Public Sub CreatePT(strQueryName As String, Optional strSQL As String, Optional strConnect As String, Optional bolReturnsRecords As Boolean = True)
```

```
' *****
```

```
' Sub: CreatePT
```

```
' Parameter:
```

```
' Autor: AGA
```

```
' Erstellt am: 22.06.2009
```

```
' Geändert am: 20.02.2010
```

```
' Funktion: erzeugt Pass-through-Abfrage aus übergebenen Parametern
```

```
' Aufruf: call CreatePT(abfName_pt, strSQL, [connectionstring], True),
```

```
' danach Aufruf der erzeugten Abfrage mit: currentdb.execute "abfName_pt"
```

```
' Achtung : Der Connection-String muss im Nachhinein angegeben werden, weil
```

```
' DAO sonst davon ausgeht, dass es sich um eine lokale ABF handelt
```

```
' *****
```

```
Dim qdf As DAO.QueryDef
```

```
On Error Resume Next
```

```
DoCmd.DeleteObject acQuery, strQueryName
```

```
If Err.Number = 7874 Then 'Objekt nicht gefunden
```

```
Err.Clear
```

```
End If
```

```
On Error GoTo LocalErrorHandler
```

```
'Abfrage anlegen
```

```
Set qdf = CurrentDbC.CreateQueryDef(strQueryName)
```

```
'den SQL-String übergeben
```

```
If Len(strSQL) > 0 Then
```

```
qdf.SQL = strSQL
```

```
End If
```

```
'den Connectionstring übergeben
```

```
If Len(strConnect) > 0 Then
```

```
'Neuen Connectionstring nutzen
```

```
qdf.Connect = strConnect
```

```
Else
```

```
'Standardconnection nutzen
```

```
qdf.Connect = GetTVAR("tvarODBCString")
```

```
End If
```

```
'liefert die ABF Daten?
```

```
qdf.ReturnsRecords = bolReturnsRecords
```

```
qdf.Close
```

```
Cleanup:
```

```
'On Error Resume Next
```

```
Set qdf = Nothing
```

```
Exit Sub
```

```
LocalErrorHandler:
```

```
'Ausgewählte Fehler hier behandeln
```

```
Select Case Err.Number
```

```
'case xxx
```

End Select

'unbehandelte Fehler an GEH durchreichen

ErrEx.CallGlobalErrorHandler

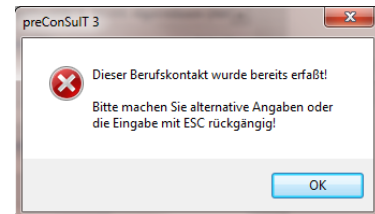
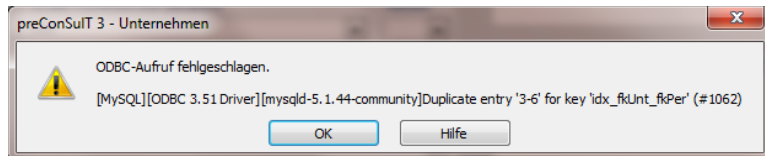
Resume Cleanup

End Sub

ODBC-Error in Access abfangen

Ein häufiger auftauchendes Problem besteht darin, dass auf Server-Seite ein Fehler auftaucht, der zwar auch durch das VBA-Error-Objekt abgefangen werden kann – allerdings kann nicht individuell auf den Fehler reagiert werden, weil Access den Fehler nur in Form eines längeren Strings aus Err.Description

mitgeteilt bekommt (verschiedene Versuche, das Error-Objekt von DBEngine o.ä. auszuwerten, gingen leider gründlich schief). Die entscheidende Info steckt dabei in der Ziffer am Ende des Error-Strings: „#1062“ – ist die MySQL-Fehlernummer für den doppelten Eintrag in einen Unique-Index. Die nachstehende (lokale) Fehleroutine fängt zunächst den Access-Fehler 3146 ab, um dann zu prüfen, welcher Fehler eigentlich vom ODBC-Treiber gemeldet wird – im vorliegenden Fall von „1062“ soll der User die nebenstehende Fehlermeldung erhalten. Mit Hilfe von GetODBCErrNr sind Sie zukünftig in der Lage, selektiv auf MySQL-Fehler zu reagieren.



Public Function GetODBCErrNr(ByVal strErrDescription As String) As String

' *****

' Function: GetODBCErrNr

' Parameter: erwartet Err.Description; liefert Fehlernummer als String

' Autor: AGA

' Erstellt am: 26.07.2010

' Geändert am: 26.07.2010

' Funktion: liefert (letzte) ODBC-Fehler-Nr. als String zurück

' (MS selbst empfiehlt dieses Vorgehen in MSDN ... [129165])

' *****

Dim intPosStart As Integer

Dim intLength As Integer

'Anfang der Fehlernummer suchen (beginnt mit "#")

intPosStart = InStr(1, strErrDescription, "#") + 1

'Länge der Fehlernummer bestimmen (endet mit ")")

intLength = InStr(intPosStart, strErrDescription, ")") - intPosStart

'ODBC-Error-Nummer auslesen

GetODBCErrNr = Mid(strErrDescription, intPosStart, intLength)

End Function

Beispiel für Aufruf (aus lokalem Error-Handler):

Select Case Err.Number

Case 3146 'Access-ODBC-Fehler (z.B. wg. doppeltem Schlüssel in n:m-Tabelle)

If GetODBCErrNr(Err.Description) = "1062" Then

MsgBox "Diese Information wurde bereits erfaßt!" & vbCrLf & vbCrLf _
& "Bitte machen Sie alternative Angaben oder" & vbCrLf _
& "die Eingabe mit ESC rückgängig!", vbCritical, "preConSulT 3"

End If
Exit Sub

GetLastInsertID (mit ADOExecuteSQL)

Gerade bei intensivem Einsatz von n:m-Beziehungen besteht häufiger der Bedarf, dass man zwei Masterdatensätze über eine Zwischentabelle verknüpfen möchte, ohne dem User größere Formular-Unterformular-Combobox-Ungeheuer zumuten zu wollen. Im gegebenen Fall soll im Dokumentenmanagement „automatisch“ ein Mitarbeiter (oder Unternehmen, Projekt usw.) mit dem soeben per Automation erstellten und in einem Ablageordner gespeicherten Word-Dokument verknüpft werden. Das übergebene „Word-Dokument“ wurde zuvor in Pfad (strDateiPfad) und Dateiname (strDateiName) getrennt aufbereitet.

Der nachfolgende Codeauszug zeigt gleich mehrere Problemlösungen auf einmal:

1. Pfadangaben müssen zwingend mit doppelten Backslashes an den MySQL-Server übergeben werden, sonst werden diese gelöscht! (auch ein Hochkomma führt zu Problemen ...)
2. Das aktuelle Datum von Access (Date()) muss aufbereitet werden, so dass es in der Form 2009-03-10 übergeben wird (das erledigt MySQLDate, s.u.)
3. Dann wird ein SQL-String in MySQL-Syntax generiert, der über eine ADO-Connection direkt auf dem Server ausgeführt wird und einen Dokumentendatensatz anlegt. Die Access-Funktion GetLastInsertID nutzt dabei die MySQL-Funktion Last_Insert_ID(), die den zuletzt in der Tabelle auf dem Server erzeugten PK derselben Session (d.h. keine Verwechslung mit anderen Usern möglich!) zurückliefert und in einer Variablen (IngpkDok) zwischenspeichert.
4. Damit ist die halbe Miete im Sack! Die zweite Hälfte wird durch das nachfolgende SQL-Statement beigetragen, das die Zwischentabelle relMADok mit der zwischengespeicherten IngpkDok sowie der ausgelesenen IngpkMA füllt und somit Mitarbeiter mit Dokument verknüpft.

```
'Pfad für MySQL aufbereiten
strDateiPfad = Replace(strDateiPfad, "\", "\\")
'Aktuelles Datum umwandeln
strEinDat = MySQLDate(Date)
'User ermitteln
strBearbeiter = GetUserName 'Aktuellen Windows-User ermitteln

'Neuen Eintrag für Dokumente schreiben
strSQL = "INSERT INTO tabDokumente (Betreff, Autor, EinDat, Datei, Pfad, DokDat, Bearbeiter) " _
        & "VALUES ('" & strBetreff & "', '" & strBearbeiter & "', '" & strEinDat & "', '" & strDateiName & "', '" &
strDateiPfad & "', '" & strEinDat & "', '" & strBearbeiter & "');"
IngpkDok = GetLastInsertID(strSQL)

'Dokument mit Mitarbeiter verknüpfen
strSQL = "INSERT INTO relMADok (fkMA, fkDok) VALUES ('" & IngpkMA & "', '" & IngpkDok & "');"
ADOExecuteSQL(strSQL)
```

```
Public Function GetLastInsertID(strSQL As String) As Long
' *****
' Function: GetLastInsertID
' Parameter: erwartet Insert-Statement; liefert Last_ID (long)
' Autor: AGA
' Erstellt am: 31.07.2009
' Geändert am: 31.07.2009
' Funktion: Ermittelt nach Insert die letzte vergebene ID der Tabelle
' Aufruf: IngpkWZ = GetLastInsertID(strSQL)
' *****

Dim con As ADODB.Connection
```

```
Dim rs As ADODB.Recordset

On Error GoTo LocalErrorHandler

Set con = New ADODB.Connection

con.ConnectionString = GetADOConString
con.Open
con.Execute strSQL

Set rs = New ADODB.Recordset

With rs
    .Open "SELECT Last_Insert_ID()", con, adOpenForwardOnly, adLockReadOnly
    GetLastInsertID = .Fields(0)
    .Close
End With

con.Close

Cleanup:
'On Error Resume Next
Set rs = Nothing
Set con = Nothing
Exit Function

LocalErrorHandler:
'Ausgewählte Fehler hier behandeln
Select Case Err.Number
'case xxx
End Select

'unbehandelte Fehler an GEH durchreichen
ErrEx.CallGlobalErrorHandler
Resume Cleanup

End Function
```

ADOExecuteSQL (aus VBA direkt auf Server)

```
Public Function ADOExecuteSQL(strSQL As String)
' *****
' Function: ADOExecuteSQL
' Parameter: Variant
' Autor: AGA
' Erstellt am: 06.08.2009
' Geändert am: 06.08.2009
' Funktion: führt SQL-Statements direkt auf dem Server aus
' Wichtig: SQL-Syntax muss MySQL entsprechen!
' *****

Dim con As ADODB.Connection

On Error GoTo LocalErrorHandler

Set con = New ADODB.Connection

With con
    .ConnectionString = GetADOConString
    .Open
    .Execute strSQL
    .Close
End With
```

```
Set con = Nothing
```

```
Cleanup:
```

```
'On Error Resume Next
Set con = Nothing
Exit Function
```

```
LocalErrorHandler:
```

```
'Ausgewählte Fehler hier behandeln
Select Case Err.Number
'case xxx
End Select
```

```
'unbehandelte Fehler an GEH durchreichen
ErrEx.CallGlobalErrorHandler
Resume Cleanup
```

```
End Function
```

Datum aufbereiten für MySQL

```
Public Function MySQLDat(datDatum As Date) As String
```

```
' *****
```

```
' Function: MySQLDat
```

```
' Parameter: erwartet: MySQLDat("01.01.2009"); liefert String: 2009-01-01 oder 0000-000-00
```

```
' Autor: AiU, AGA
```

```
' Erstellt am: 13.07.2009
```

```
' Geändert am: 13.07.2009
```

```
' Funktion: Bringt ein Datum in MySQL-taugliche Form: 2009-03-10
```

```
' Einsatz: "SELECT ... WHERE Datum=" & MySQLDat("01.01.2009")
```

```
' WICHTIG: das Datum muss im SQL-Statement als String in ' ' eingebunden werden!
```

```
' *****
```

```
If Not IsNull(datDatum) Then
    MySQLDat = format(datDatum, "yyyy-mm-dd")
Else
    MySQLDat = "0000-00-00"
End If
```

```
End Function
```

MySQL-Datumsfunktionen

```
'Datum im Format TT.MM.JJJJ
```

```
SELECT
DATE_FORMAT(KonDatum,GET_FORMAT(DATE,'EUR'))
FROM tabKontakt;
```

```
'Zeit im Format hh:mm:ss
```

```
SELECT
DATE_FORMAT(KonDat,GET_FORMAT(TIME,'EUR'))
FROM tabKontakt;
```

```
'Datum in TT.MM.JJJJ und String wandeln, dann mit weiterem String konkatenieren:
```

```
'liefert: "15.09.2009 Treffen mit Herrn Schmidt; ID"
```

```
SELECT CONCAT(CAST(DATE_FORMAT(KonDat,GET_FORMAT(DATE,'EUR'))AS CHAR), ' ', Betreff) AS
Kontakt, pkKon
```

```
FROM tabkontakt  
ORDER BY KonDat DESC;
```

Diese etwas komplexere Schachtelung soll kurz erläutert werden: Die innere Funktion DATE_FORMAT wandelt das Kontaktdatum von bspw. „2010-10-23“ in das im deutschen Raum genutzte Format „23.10.2010“ um; dabei wird als zweiter Parameter die Hilfsfunktion GET_FORMAT zur Definition der korrekten Datumsansetzung genutzt. Die mittlere Funktion CAST(...) AS CHAR wandelt das resultierende Ergebnis in eine Textvariable (CHAR) um, die dann durch die äußere Funktion CONCAT mit einem Leerschritt und dem Inhalt des Feldes „Betreff“ verkettet wird.

'Mit SQL Differenz in Tagen aus zwei Daten errechnen
(TO_DAYS(CURDATE()) - TO_DAYS(`tabsemtermin`.`SemDat`)) AS `SemDatDiff`

Beispiele für MySQL-SQL-Syntax (Insert, Update, Delete)

```
INSERT INTO tabWZ (WZAbs, WZCode, WZ, WZAbt, WZG, WZBez)  
VALUES ('" & strWZAbs & "', '" & strWZCode & "', '" & strWZ & "', '" & strWZAbt & "', '" & strWZG & "',  
'" & strWZBez & "');
```

Beispiel für rels (Zwischentabellen zur n:m-Verknüpfung zweier Mastertabellen):
ADOExecuteSQL ("INSERT INTO relKonAnf (fkKon, fkAnf) VALUES ('" & lngpkKon & "', '" & Me!pkAnf & "');")

'Datum einfügen:
1. In Ansi-Datum wandeln (eigene Funktion in Access, s.o.)
2. Als String einfügen
strEinDat = MySQLDat(Date)
INSERT INTO ... VALUES ('" & strEinDat & "', ...

'Integer-Zahlen einfügen:
VALUES ('" & lngpkID & "', ...
(geht direkt auf Server sowohl mit ' ' als auch ohne!)

'Dezimal-Zahl einfügen:
1. Komma gegen Punkt austauschen: Replace(Me!GesPreis, ",", ".")
2. Als String einfügen (s.o.)

'Dateipfade mit "\":
das "\" wird gelöscht, daher alle "\" verdoppeln:
strVerzeichnis = Replace(strVerzeichnis, "\", "\\")

```
UPDATE tabThema  
SET WZG='47.11', Branche='Langer Text'  
WHERE pkThe=347;
```

```
UPDATE tabunternehmen  
SET KontaktVon = '2009-09-01',KontaktBis = '2009-09-15',Dauer = '60',GesKon = '5'  
WHERE pkUnt = 3;
```

```
UPDATE tabperkom
```


SET tabperkom.NurText = tabperkom.Kommentar

DELETE FROM tabWZ
WHERE pkWZ=3;

Beispiele: Strings mit MySQL konkatenieren (Concat, IsNull, If, As, View, CAST)

'SQL-Statement, das unter Verwendung einer View folgendes Ergebnis liefert:

'[wenn Person MA eines Unternehmens]

'Neutzler, Regina - Leitung Buchhaltung (AGA Media - Joachim Bartels)

[wenn Privatperson]

'Poulheim, Regina - Privat

```
SELECT
  CONCAT(tabperson.Nachname, ' ',
        tabperson.Vorname, ' - ',
        IF(ISNULL(vw_maunt.Funktion),'Privat',vw_maunt.Funktion), ' ',
        IF(ISNULL(vw_maunt.Bereich),'',vw_maunt.Bereich),
        IF(ISNULL(vw_maunt.UntName),'',CONCAT('(',vw_maunt.UntName,')'))
  ) AS Person,
  tabperson.pkPer
FROM agamedia.vw_maunt
  RIGHT JOIN agamedia.tabperson
    ON (vw_maunt.fkPer = tabperson.pkPer);
```

'einfache Variante

```
SELECT
  reluntma.pkMA,
  CONCAT(tabperson.Nachname, ' ', tabperson.Vorname, ' (', reluntma.Funktion, ') ', tabunterneh-
  men.UntName) AS Person
FROM tabperson INNER JOIN (tabunternehmen INNER JOIN reluntma ON tabunternehmen.pkUnt =
  reluntma.fkUnt) ON tabperson.pkPer = reluntma.fkPer
ORDER BY tabperson.Nachname, tabperson.Vorname;
```

Probleme:

CONCAT(Datum, ' ', Betreff) AS Kontakt

funktioniert nicht unter Access, da eine explizite Typumwandlung stattfinden muss:

-> CONCAT(CAST(Datum AS CHAR), ' ', Betreff) AS Kontakt

'Datum in TT.MM.JJJJ und String wandeln, dann mit weiterem String konkatenieren;

'liefert: "15.09.2009 Treffen mit Herrn Schmidt; ID"

```
SELECT CONCAT(CAST(DATE_FORMAT(KonDat,GET_FORMAT(DATE,'EUR'))AS CHAR), ' ', Betreff) AS
  Kontakt, pkKon
FROM tabkontakt
ORDER BY KonDat DESC;
```

Systemabfragen mit MySQL-Funktionen

'Versionsnummer des Servers

```
SELECT VERSION();
```

'Aktueller User

```
SELECT USER();
```

'Name der Datenbank

```
SELECT DATABASE();
```

'Connection-ID

```
SELECT CONNECTION_ID();
```

View auf dem Server erstellen

Eine View, die auf dem Server erstellt wurde, kann ebenso wie die „normalen“ Tabellen als verknüpfte Tabelle in Access eingebunden werden. Im Folgenden gibt es ein (komplexes) Beispiel eines Joins über 8 Tabellen, um die Teilnehmer eines Seminars anzuzeigen. Darüber hinaus werden einzelne Funktionen (s.o.) eingesetzt, um die Differenz zwischen zwei Datumsangaben oder konkatenierte Personenstrings zu erzeugen. Ich habe Views anfänglich erst genutzt, dann aber darauf verzichtet, da eine gleiche Access-Abfrage ähnlich performant erschien und weniger problematisch bezüglich der Aktualisierung des Users und seiner Rechte ist.

```
CREATE ALGORITHM=UNDEFINED DEFINER=`supervisor`@`localhost`  
SQL SECURITY DEFINER VIEW `vw_seminarteilnehmer` AS  
SELECT `tabsemreihe`.`pkRei` AS `pkRei`,  
       `tabsemreihe`.`Reihe` AS `Reihe`,  
       `tabseminar`.`pkSem` AS `pkSem`,  
       `tabseminar`.`SemStat` AS `SemStat`,  
       `tabseminar`.`SemTitel` AS `Seminar`,  
       `tabseminar`.`KZF` AS `KZF`,  
       `tabseminar`.`SemLevel` AS `SemLevel`,  
       `tabseminar`.`SemInhalt` AS `SemInhalt`,  
       `tabsemkalender`.`fkZei` AS `fkZei`,  
       `tabsemzeitraum`.`Zeitraum` AS `Zeitraum`,  
       `tabsemzeitraum`.`Jahr` AS `Jahr`,  
       `tabsemkalender`.`pkKal` AS `pkKal`,  
       `tabsemorga`.`Veranstalter` AS `Veranstalter`,  
       `tabsemorga`.`Einheiten` AS `Einheiten`,  
       `tabsemorga`.`Intervall` AS `Intervall`,  
       `tabsemorga`.`Organisator` AS `Organisator`,  
       `tabsemtermin`.`pkTer` AS `pkTer`,  
       `tabsemtermin`.`Status` AS `Status`,  
       `tabsemtermin`.`Einheit` AS `Einheit`,  
       `tabsemtermin`.`SemDat` AS `SemDat`,  
       (TO_DAYS(CURDATE()) - TO_DAYS(`tabsemtermin`.`SemDat`)) AS `SemDatDiff`,  
       `tabsemtermin`.`Beginn` AS `Beginn`,  
       `tabsemtermin`.`Ende` AS `Ende`,  
       `tabsemanmeldung`.`pkAnm` AS `pkAnm`,
```

```

        `tabsemanmeldung`.`AnmDat` AS `AnmDat`,
        `tabperson`.`Nachname` AS `Nachname`,
        `tabsemanmeldung`.`fkPer` AS `fkPer`,
        `tabperson`.`Anrede` AS `Anrede`,
        `tabperson`.`Vorname` AS `Vorname`,
        IF(ISNULL(`tabperson`.`Vorname`),`tabperson`.`Nachname`,CONCAT(`tabperson`.`Nachname`
            ,',',`tabperson`.`Vorname`)) AS `Teilnehmer`,
        `tabsemanmeldung`.`AnmStat` AS `AnmStat`,
        `tabsemanmeldung`.`AbsDat` AS `AbsDat`,
        (SELECT COUNT(`tabsemanmeldung`.`fkPer`) AS `TeilnehmerAnz`
FROM `tabsemanmeldung`
WHERE ((`tabsemanmeldung`.`fkTer` = `tabsemtermin`.`pkTer`)
    AND (`tabsemanmeldung`.`AnmStat` = 'Teilnehmer')))) AS `AnzTeilnehmer`,
        (SELECT COUNT(`tabsemanmeldung`.`fkPer`) AS `TeilnehmerAnz`
FROM `tabsemanmeldung`
WHERE ((`tabsemanmeldung`.`fkTer` = `tabsemtermin`.`pkTer`)
    AND (`tabsemanmeldung`.`AnmStat` = 'Abwesend')))) AS `AnzAbsagen`
FROM ((`tabsemreihe` JOIN `tabseminar` ON((`tabsemreihe`.`Reihe` =
    `tabseminar`.`SemReihe`)))
JOIN ((`tabsemzeitraum` JOIN `tabsemkalender` ON((`tabsemzeitraum`.`pkZe` =
    `tabsemkalender`.`fkZe`)))
JOIN ((`tabsemtermin` JOIN `tabsemorga` ON((`tabsemtermin`.`pkTer` =
    `tabsemorga`.`fkTer`)))
JOIN (`tabperson` JOIN `tabsemanmeldung` ON((`tabperson`.`pkPer` =
    `tabsemanmeldung`.`fkPer`)))
ON((`tabsemtermin`.`pkTer` = `tabsemanmeldung`.`fkTer`)))
ON((`tabsemkalender`.`pkKal` = `tabsemorga`.`fkKal`))) ON((`tabseminar`.`pkSem` =
    `tabsemkalender`.`fkSem`)))

```

Professionalisierung

1. MapPoint-Control

Standort für Unternehmen geokodieren

Görlitzer Str. 51 90473 Nürnberg

Standard Länge 11,137335 Breite 49,399787 Entf. Luftlinie 327,55 km

Ort oder Adresse eingeben Straßensuche

Zeit	km	Anweisung	Teilstrecke	Richtung
Übersicht: 463,8 Kilometer (3 Stunden, 56 Minuten)				
09:00	0,0	1 Abfahrt Start auf Zum Muttergottesbild (Nord 0,1 km		
09:00	0,1	Neuer Straßenname: Ab hier Kranzweg	0,6 km	
09:02	0,8	Neuer Straßenname: Ab hier Kolpingstraße	0,5 km	
09:04	1,3	RECHTS abbiegen (Ost) auf L479 [Hildesheimer Straße]	0,6 km	
09:06	1,9	LINKS abbiegen (Ost) auf L479	1,9 km	

Route zum Standort anzeigen Geokodierung löschen Beenden

Das MapPoint-Control, das mit der MS-MapPoint-Anwendung im System-Verzeichnis installiert wird, stellt diverse Funktionen auf der Oberfläche (einfach mal das Kontextmenü aufrufen) sowie über das Objektmodell in VBA bereit. Um die im Vortrag vorgeführten Beispiele zu rekapitulieren, reichen wenige Zeilen Code; im Folgenden werden die postalischen Daten aus dem Formular ausgelesen und geokodiert.

[objMapCtrl ist im Kopf des Formular-Klassenmoduls definiert:

Dim objMapCtrl As MappointControl

und wird im Form_Unload wieder zerstört (s.u.)]:

```
Private Sub btnGeoKodieren_Click()
```

```
' *****
```

```
' Sub: btnAdrAnzeigen_Click
```

```
' Parameter:
```

```
' Autor: AGA
```

```
' Erstellt am: 20.07.2009
```

```
' Geändert am: 20.07.2009
```

```
' Funktion: zeigt den aktuellen Standort auf der Karte an und ermittelt Geo-Koordinaten
```

```
' *****
```

```
Dim objLocation As MapPoint.Location
```

```
Dim strName As String
```

```
Dim dblDistance As Double
```

```
Set objMapCtrl = Me.mapStandort.Object
```

```
'Suche nach der angegebenen Adresse und Übernahme des 1. Treffers
```

```
Set objLocation = objMapCtrl.ActiveMap.FindAddressResults(Me!Strasse, Me!Ort, , Me!PLZ,  
Nz(Me!cbLKZ, "Deutschland")).Item(1)
```

```
Select Case left(Me.OpenArgs, 1) 'dient der Beschriftung des Pins auf der Karte
```

```
Case "u" 'Unternehmensnamen ermitteln
```

```
strName = GetUntName(Mid(Me.OpenArgs, 2, Len(Me.OpenArgs)))
```

```
Case "p" 'Person ermitteln
```

```
strName = GetPerName(Mid(Me.OpenArgs, 2, Len(Me.OpenArgs)))
```

```
End Select
```

```
'Pin-Symbol geokodieren und mit Unternehmensnamen versehen
```

```
objMapCtrl.ActiveMap.AddPushpin objLocation, strName
```

```
'objMapCtrl.ActiveMap.DataSets(2).Name = "AGA Media" '(1) funktioniert nicht, wenn schon  
eigene Karte geladen: dann werden Gebietsüberschriften überschrieben
```

```
'Symbol zuordnen
```

```
objMapCtrl.ActiveMap.DataSets(1).Symbol = 1
```

```
'Auf Kartenausschnitt mit geokodiertem Standort zoomen
```

```
objMapCtrl.ActiveMap.DataSets.ZoomTo
```

```
'Längen- und Breitengrad ermitteln (6 Dezimalstellen anzeigen) und in Form schreiben
```

```
Me!Longitude = format(objLocation.Longitude, "###.#####")
```

```
Me!Latitude = format(objLocation.Latitude, "###.#####")
```

```
'Distanz zum Lizenznehmer ermitteln (anderer Datensatz, der bereits geokodiert ist)
```

```
dblDistance = GetDistance(objLocation) 'Hilfsfunktion, s.u.
```

```
If dblDistance = 0 Then 'Standort Lizenznehmer
```

```
Me!Luftlinie = "0"
```

```
Else 'jeder andere Standort (2 Dezimalstellen anzeigen)
```



```
Me!Luftlinie = format(dblDistance, "#####.##")
End If
```

Hierzu gehört die Hilfsfunktion GetDistance:

```
Public Function GetDistance(objLockKunde As MapPoint.Location) As Double
' *****
' Function: GetDistance
' Parameter: erwartet Referenz auf Karte und Location; liefert Double
' Autor: AGA
' Erstellt am: 27.07.2009
' Geändert am: 27.07.2009
' Funktion: ermittelt Längen- und Breitenkoordinaten zur übergebenen
'           Location und dann die Entfernung vom eigenen Standort
' *****
Dim objMap As MapPoint.Map
Dim objLocLizNehmer As MapPoint.Location
Dim strSQL As String
Dim rs As DAO.Recordset
Dim dblLong As Double
Dim dblLat As Double

' Standortkoordinaten des Lizenznehmers der aktuellen DB ermitteln
strSQL = "SELECT tabadresse.Longitude, tabadresse.Latitude " _
        & "FROM tabadresse INNER JOIN ((tabunternehmen INNER JOIN (tabdb INNER JOIN tabli-
zenz ON tabdb.pkDB = tablizenz.fkDB) ON tabunternehmen.pkUnt = tablizenz.fkUnt) INNER JOIN
reluntadr ON " _
        & "tabunternehmen.pkUnt = reluntadr.fkUnt) ON tabadresse.pkAdr = reluntadr.fkAdr " _
        & "WHERE (((tabdb.DBName)='" & GetTVar("tvarLastDB") & "')));"

Set rs = CurrentDbC.OpenRecordset(strSQL)

dblLong = Nz(rs!Longitude, "0")
dblLat = Nz(rs!Latitude, "0")

Set objMapCtrl = Me.mapStandort.Object
Set objMap = objMapCtrl.ActiveMap

' Location des Lizenznehmers ermitteln
Set objLocLizNehmer = objMap.GetLocation(dblLat, dblLong)
' Luftliniendistanz zum Kunden ermitteln
GetDistance = objLockKunde.DistanceTo(objLocLizNehmer)

End Function
```

Die Spalten in der Adresstabelle für „Longitude“ und „Latitude“ sind als Zahl vom Typ Double definiert (10,6). Auch die Ermittlung der Route vom Lizenznehmer zum betreuten Kunden ist dann schnell ermittelt (es wird noch ein Meldeformular „Die Route wird berechnet ...“ eingeblendet):

```
Private Sub btnRoute_Click()
' *****
```

```
' Sub: btnRoute_Click
' Parameter:
' Autor: AGA
' Erstellt am: 04.09.2009
' Geändert am: 04.09.2009
' Funktion: zeigt Routenplanung auf Karte an
' *****

Dim objLocation As MapPoint.Location
Dim strName As String
Dim dblDistance As Double

'Meldung ausgeben
DoCmd.OpenForm "frmDlgRoute"
Set objMapCtrl = Me.mapStandort.Object
'Suche nach der angegebenen Adresse und Übernahme des 1. Treffers
Set objLocation = objMapCtrl.ActiveMap.FindAddressResults(Me!Strasse, Me!Ort, , Me!PLZ,
Nz(Me!cbLKZ, "Deutschland")).Item(1)
'Route Lizenznehmer -> Kunde berechnen
Call ShowRoute(objLocation)
DoCmd.Close acForm, "frmDlgRoute"
```

Die oben aufgerufene Funktion ShowRoute sieht wie folgt aus:

```
Private Sub ShowRoute(objLocKunde As MapPoint.Location)
' *****
' Function: ShowRoute
' Parameter: erwartet Referenz auf Karte und Location
' Autor: AGA
' Erstellt am: 27.07.2009
' Geändert am: 27.07.2009
' Funktion: errechnet Route vom eigenen Standort zum Kunden
' *****

Dim objMap As MapPoint.Map
Dim objLocLizNehmer As MapPoint.Location
Dim objRoute As MapPoint.Route
Dim strSQL As String
Dim rs As DAO.Recordset
Dim dblLong As Double
Dim dblLat As Double

'Standortkoordinaten des Lizenznehmers der aktuellen DB ermitteln
strSQL = "SELECT tabadresse.Longitude, tabadresse.Latitude " _
        & "FROM tabadresse INNER JOIN ((tabunternehmen INNER JOIN (tabdb INNER JOIN tabli-
zenz ON tabdb.pkDB = tablizenz.fkDB) ON tabunternehmen.pkUnt = tablizenz.fkUnt) INNER JOIN
reluntadr ON " _
        & "tabunternehmen.pkUnt = reluntadr.fkUnt) ON tabadresse.pkAdr = reluntadr.fkAdr " _
        & "WHERE (((tabdb.DBName)='" & GetTVar("tvarLastDB") & "')));"

Set rs = CurrentDbC.OpenRecordset(strSQL)

dblLong = Nz(rs!Longitude, "0")
dblLat = Nz(rs!Latitude, "0")
```



```
Set objMapCtrl = Me.mapStandort.Object
Set objMap = objMapCtrl.ActiveMap
'Set objRoute = objMapCtrl.ActiveRoute

'evt. vorhandene Route löschen
objMap.ActiveRoute.Clear

'Location des Lizenznehmers ermitteln
Set objLocLizNehmer = objMap.GetLocation(dblLat, dblLong)

'Start und Zielpunkt hinzufügen und Route berechnen
objMap.ActiveRoute.Waypoints.Add objLocLizNehmer, "Start"
objMap.ActiveRoute.Waypoints.Add objLocKunde, "Ziel"
objMap.ActiveRoute.Calculate
```

End Sub

Wenn eine Adresse bereits geokodiert wurde (Longitude NOT Null) dann kann beim Aufruf des Adressformulars der Standort auf der Karte automatisch eingeblendet werden.

Private Sub Form_Current()

```
' *****
' Sub: Form_Current
' Parameter:
' Autor: AGA
' Erstellt am: 15.07.2009
' Geändert am: 15.07.2009
' Funktion: Standort anzeigen, wenn bereits geokodiert
' *****
```

```
If ((Not IsNull(Longitude)) And (Longitude <> 0)) Then
```

```
Dim objMap As MapPoint.Map
Dim objLoc As MapPoint.Location
Dim objPin As MapPoint.Pushpin
Dim strName As String
```

```
Set objMapCtrl = Me.mapStandort.Object
Set objMap = objMapCtrl.ActiveMap
Set objLoc = objMap.GetLocation(Me!Latitude, Me!Longitude)
```

```
Select Case left(Me.OpenArgs, 1)
Case "u" 'Unternehmensnamen ermitteln
    strName = GetUntName(Mid(Me.OpenArgs, 2, Len(Me.OpenArgs)))
Case "p" 'Person ermitteln
    strName = GetPerName(Mid(Me.OpenArgs, 2, Len(Me.OpenArgs)))
End Select
```

```
'Suche nach den Koordinaten und zeige Standort an
Set objPin = objMap.AddPushpin(objMap.GetLocation(Me!Latitude, Me!Longitude), strName)
'auf Kartenausschnitt mit geokodiertem Standort zoomen
objMap.Location.Goto
objMap.DataSets.ZoomTo
```

```
'Info-Fenster mit Namen anzeigen  
objPin.BalloonState = geoDisplayBalloon  
End If
```

```
End Sub
```

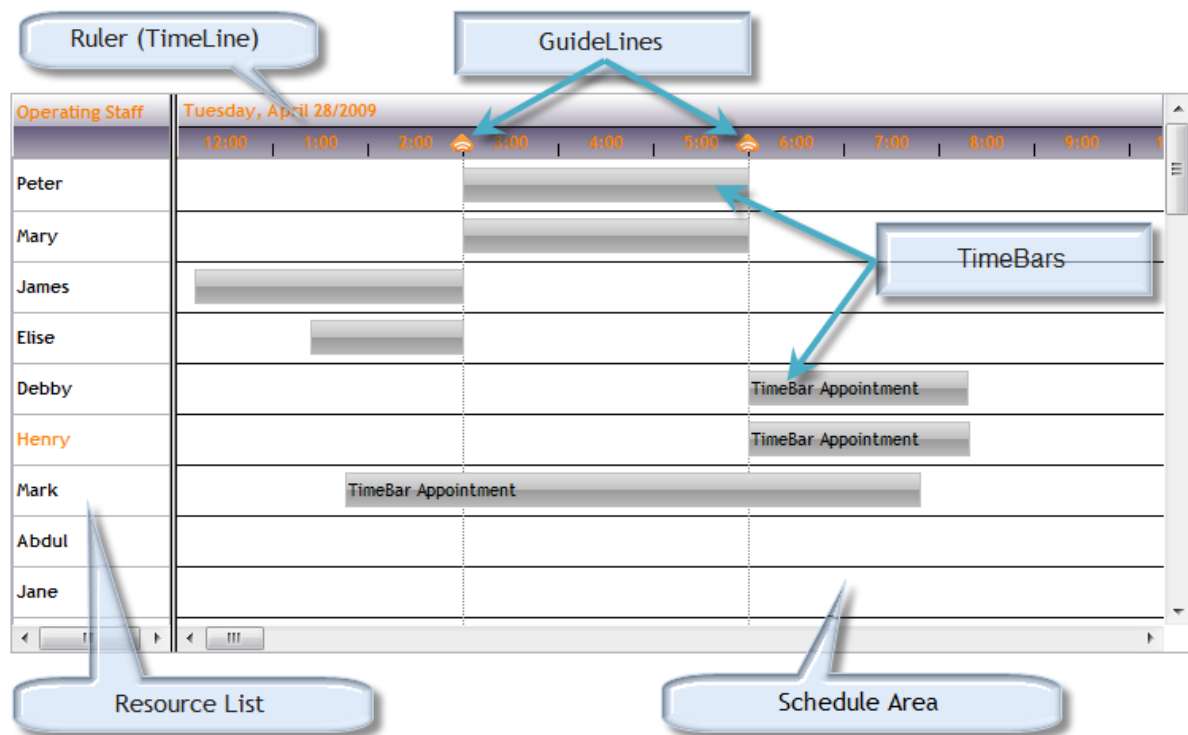
So, zum Schluss die Objektvariable wieder zerstören und die Nachfrage, ob die erzeugte Karte gespeichert werden soll, unterbinden.

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    objMapCtrl.ActiveMap.Saved = True  
    Set objMapCtrl = Nothing
```

```
End Sub
```

2. ctSchedule



The screenshot shows the ctSchedule application window. On the left, there is a calendar view for September, October, and November 2010. The main area displays a schedule grid for Friday, 24. September 2010, with columns for hours from 7:00 to 16:00. The grid shows appointments for various resources, including:

- Adamski, Matthias (n/a)
- Bartels, Joachim (EDV-Dozent)
- Beese, Clas (n/a)
- Fischer, Wolfram (n/a)
- Günter, Carsten (n/a)
- Langnau, Axel (n/a)
- Lüders, Jörg (Betriebswirt)
- Reiter, Andreas (Steuerberater)
- Schmidt, Tobias (Betriebswirt)
- Weber, Kai (Betriebswirt)
- Zimmermann, Michael (Mathematiker)

Appointments shown in the grid include:

- Franchising - Aktive Podiumsdiskussion (13:00-14:00)
- Franchising - Aktive Die besten (11:00-12:00)
- Kurzseminar "Businessplan" (Einzeltermin) Alles ist Marketing! (10:00-11:00)
- Franchising - Aktive Rechtliche Hintergründe (11:00-12:00)
- Franchising Begrüßung (10:00-11:00)

At the bottom left, there are settings for the interface, including "Navigationsleiste anzeigen", "Vertikalen Wechsel erlauben", and "Zeitachse skalieren (+/-): 60".

Das Control ctSchedule ist in mehrere Bereiche untergliedert, so dass auf der linken Seite in einer (hierarchisierbaren) Liste die Ressourcen (bspw. Dozenten oder Räume für Seminare) angezeigt wer-

den können (mit Datenbindung über ein Recordset und bedarfsweiser Identifizierung durch Speicherung der pkID im Listeneintrag-Tag). Auf der rechten Seite befindet sich der Schedule- (Zeitplanungs-) Bereich, der im Stunden- (eines Tages), Tages-, Wochen-, Monats-, Quartals- oder Jahresformat angezeigt werden kann (insofern auch für Projektplanung geeignet). Im Schedule-Bereich können Tageszeiten (z.B. vor 7:00 und nach 19:00 Uhr) oder ganze Tage (Wochenende, Feiertage, Betriebsferien u.a.) für die Planung gesperrt werden, Konflikte zwischen Zeitbalken werden automatisch optisch gekennzeichnet oder programmatisch unterbunden. Durch das Aufziehen von Zeitbalken (Timebars) mit der Maus erfolgt die konkrete Planung zu einer Ressource, die ebenfalls über ein Recordset mit Datenbindung ausgestattet werden kann. Zeitbalken können innerhalb des angezeigten Schedules per Drag&Drop verschoben werden, sie können mehrzeilige Informationen anzeigen, unterschiedliche Füllungen, Farben oder Icons aufweisen usw. usw.

Leider fehlen für ctSchedule seit der Version 9 die Access-Beispiele. Wer mit dem Tool unter Access entwickeln möchte, kann sich diesbezüglich an mich wenden; die Beispiele der Version 8 funktionieren auch mit der aktuellen Version 10, wenn man zuvor das Control im Formular ersetzt (E-Mail-Adresse s. Einleitung).

3. sevDataGrid2

Unternehmensstandorte - Übersicht

Sämtliche Unternehmen

untern

Art	Unternehmensname	Kundentyp	Zuständig	NWP	Web	Warnu...	Status	Gegr...	Rechtsform	ABC	M
Unternehmen	AGA Media - Joachim Bartels	Kleinunternehmen	menkesu	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	2005	GmbH	A	2-
Institution	Amtsgericht Elmshorn	Mittl. Unternehm...	AGA Media	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	n/a	KöR	X	n/
Institution	Amtsgericht Hildesheim	Mittelstand (KMU)	menkesu	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1949	KöR	X	n/
Institution	Amtsgericht Osnabrück	Mittelstand (KMU)	schmidtt	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1949	KöR	X	n/
Unternehmen	Arvena Park - Hotel am Frank...	Mittl. Unternehm...	AGA Media	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	n/a	GmbH	X	n/
Institution	Beschäftigungsförderung Göt...	Kleinunternehmen	pohlman	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1998	KöR	X	n/
Unternehmen	BNW - Bildungswerk der Nied...	Kleinunternehmen	menkesu	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1969	gGmbH	A	11
Institution	Bürgerhaus Kaltenkirchen	n/a	AGA Media	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	n/a	n/a	X	n/
Unternehmen	Commerz Kontor GmbH	Kleinunternehmen	menkesu	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1999	GmbH	X	n/
Unternehmen	CUT - Centrum für Umwelt un...	Kleinunternehmen	pohlman	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	n/a	n/a	X	n/
Unternehmen	Dieter Otter Software-Entwick...	Kleinunternehmen	menkesu	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1992	Einzelunte...	X	n/
Unternehmen	Emsland GmbH	Kleinunternehmen	Joachim Bartels	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1959	GmbH	A	2-
Unternehmen	Enigma Company Builders G...	Mittelstand (KMU)	menkesu	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1998	GmbH	X	n/
Institution	Gemeinde Algermissen	Mittelstand (KMU)	Joachim Bartels	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1949	KöR	X	11
Unternehmen	Gewerbebau Amberg GmbH	Kleinunternehmen	Joachim Bartels	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1990	GmbH	B	11
Unternehmen	Gründerhaus Osnabrück.Osn...	Kleinunternehmen	schmidtt	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	2006	n/a	X	n/
Unternehmen	H+H Gründerzentrum Wittenb...	n/a		<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	n/a	n/a	X	n/
Institution	Handelskammer Hamburg	Mittelstand (KMU)	schmidtt	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1949	KöR	X	n/
Unternehmen	HIREG - Wirtschaftsförderung...	Kleinunternehmen	schmidtt	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	2003	GmbH	A	6-
Unternehmen	human-data - Bruno Koch	Kleinunternehmen	menkesu	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Normal	Aktiv	2003	Einzelunte...	A	1
Unternehmen	IHK Osnabrück-Emsland	Mittelstand (KMU)	schmidtt	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1949	KöR	X	n/
Unternehmen	IHK Schleswig-Holstein	n/a		<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	n/a	KöR	X	n/
Unternehmen	IMB - Institut für Mittelstand...	Kleinunternehmen	schmidtt	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	2002	GmbH	X	n/
Institution	Industrie- und Handelskamm...	Mittelstand (KMU)	schmidtt	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1950	KöR	X	n/
Unternehmen	JBL Softwareentwicklung GmbH	Kleinunternehmen	rneutzler	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	2010	GmbH	X	2-
Bank	KfW Bankengruppe	Großunternehmen	AGA Media	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1948	KöR	X	n/
Institution	Landkreis Amberg-Sulzbach	Mittelstand (KMU)	Joachim Bartels	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	1949	KöR	A	51
Unternehmen	Landkreis Verden	n/a		<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	n/a	KöR	X	n/
Bank	NBank - Investitions- und För...	Mittl. Unternehm...	AGA Media	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	2004	KöR	X	n/
Unternehmen	OLEG - Osnabrücker Land-Ent...	Kleinunternehmen	pohlman	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	n/a	GmbH	X	n/
Unternehmen	raumbezug Kruse & Teubert G...	Kleinunternehmen	menkesu	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	2006	GbR/BGB	X	n/
Unternehmen	Reinhold Neufeldt, Geograph...	Kleinunternehmen	menkesu	<input type="checkbox"/>	<input type="checkbox"/>	Normal	Aktiv	2000	Einzelunte...	X	n/

Anzeige

Suchzeile

Anzahl Treffer

Unt. bearbeiten

Neues Unternehmen

Daten an Excel

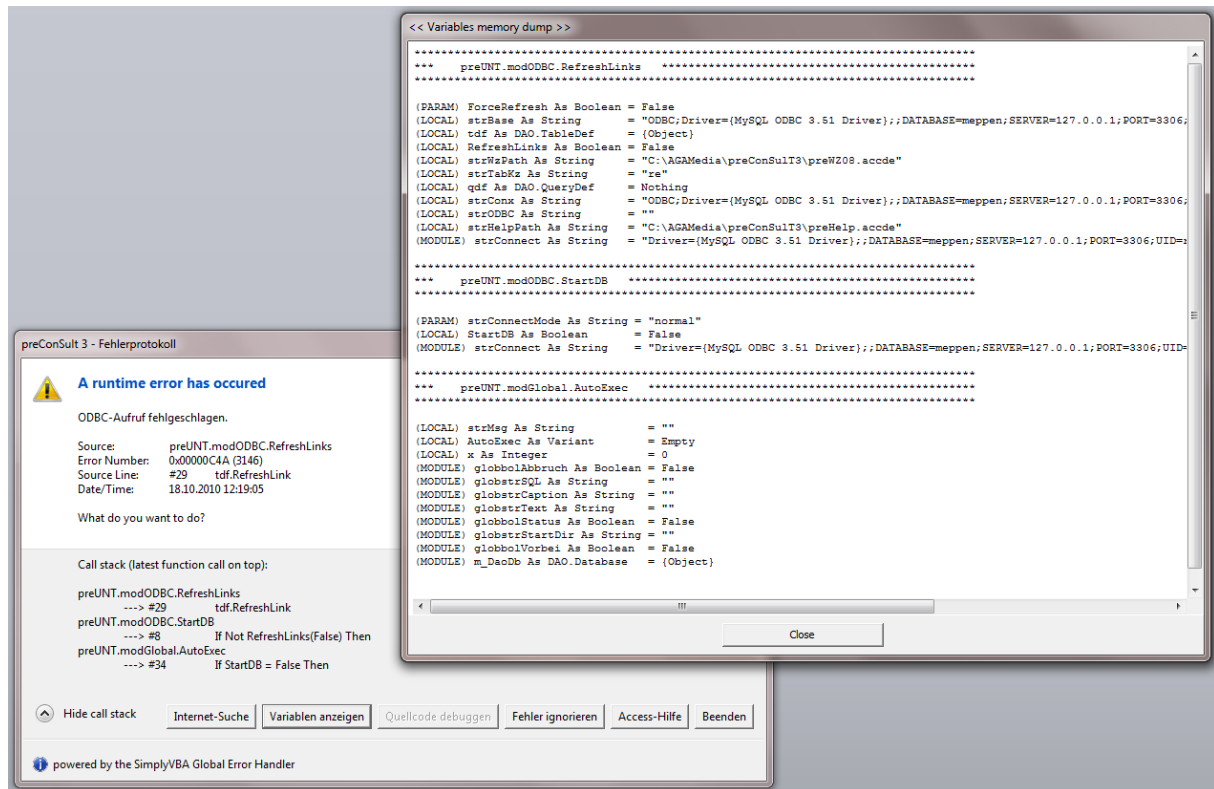
Web-Report

Auswertungen

Beenden

sevDataGrid2 funktioniert einwandfrei mit Access – einziger Wermutstropfen ist, dass nach Installation und Registrierung einer neuen Version das Steuerelement im Formular gelöscht und erneut integriert werden muss (das gilt übrigens für viele OCXe). Da der Hersteller Dieter Otter eine exzellent dokumentierte Beispieldatenbank im Download anbietet und auch die Onlinehilfe keine Wünsche offen lässt, spare ich mir an dieser Stelle weitere Beispiele.

4. Global Error Handler (vbWatchDog)



Der SVGEH kann als DLL auf den Userrechner kopiert/registriert und aus VBA heraus initialisiert werden (in der neuen Version vbWatchDog benötigt man nur noch den Import von vier Klassenmodulen), woraufhin sämtliche Fehler in Access oder der Runtime auf den SVGEH laufen und dort abgefangen bzw. zur weiteren Entscheidung präsentiert werden (s. unteres Formular, das die Entwicklerversion der Fehleranzeige zeigt; der End-User könnte eine kleinere Maske angezeigt bekommen, mit der Möglichkeit, diese Infos per Mail an den Entwickler zu senden). Bei Bedarf kann sowohl die Abfolge, die zum Fehler führte (von unten nach oben gelesen: erst AutoExec, dann StartDB bis zum Crash in RefreshLinks) als auch der jeweilige Status der Variablen angezeigt bzw. in einer Tabelle mit weiteren Infos (User, Datum, Software usw.) protokolliert werden.

Auch Wayne Phillips stellt eine sehr gut dokumentiert Access-Datenbank und Hilfefunktion bereit, mit der man das Tool vor Einsatz testen kann.