# Migration von A2K nach A07/MySQL + Professionalisierung durch OCXe

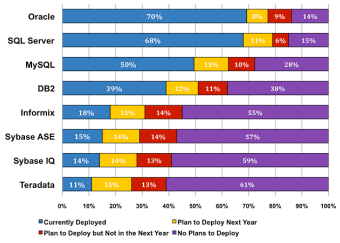
Ein Praxisbericht von Joachim Bartels
www.agamedia.de

## Persönliche Migrationsgründe

- Ausgangssituation: „Finsteres Mittelalter“ [Demo]
- Gründe für die Migration
  - Professionalisierung der bisherigen Entwicklung
  - Skalierbarkeit, Zukunftssicherheit (COM vs .NET)
  - Hauptproblem: Constraint-/Index-Limit in Access (max. 32)
  - Datenschutz verbessern
  - Produktpositionierung für größere Kunden
- Zielsetzung: „Strahlende Zukunft“ [Demo]
  - Attraktives und modernes GUI
  - Entwicklung mit „beherrschbarem“ Server
  - Geringer Administrationsaufwand für Server beim Kunden
  - Wachstumspfad mit Rücksicht auf eigene Fertigkeiten

## Status von MySQL im DB-Markt

- Ca. 25% Marktanteil (2007 – „now powered by Oracle“)




<--- Gartner Group 2008
Datenbank-Installationen und geplante Bereitstellung

**MySQL-Marketing 2009:**
ca. 70.000 Downloads/Tag
ca. 12 Mio. Installationen
ca. 16.000 zahlende Kunden
WIN-Development: 72%
WIN-Deployment: 59%

**“Why Move to MySQL from Microsoft SQL Server?” [Anhang]**

## MySQL OEM-Versionen, Kosten

- GPL vs Closed-Source: ein kontroverses Thema [Anhang]
  - *„Wer Software zusammen mit MySQL als Paketlösung vertreibt und den eigenen Quellcode nicht unter der GPL lizenzieren will, muss eine kommerzielle MySQL-Lizenz erwerben.“*
  - *„Eine Verbreitung liegt erst vor, wenn das Programm öffentlich, also gegenüber unbeteiligten Dritten, angeboten wird.“*

- Versionen + Kosten [www.mysql.de/oem/products.html]
  - MySQL Classic (nur MyISAM/HEAP-Engines) – 499 USD
  - **MySQL Pro (alle Engines inkl. InnoDB) – 899 USD (für Access!)**
  - MySQL Advanced (Pro + Partitionierung) – 1.199 USD
  - Enthalten sind der MySQL-Server 5.1.x inkl. ODBC-Treiber-Lizenz
  - Server darf nur im Paket, nicht Stand-alone eingesetzt werden!
  - Diverse (unübersichtliche) Rabattoptionen ab 25 Lizenzen

## Subjektiver Blick auf Highlights von MySQL

- Einfache Installation [Anhang]:
  - Server mit DB nach 15 Minuten einsatzbereit (ca. 225 MB)
  - 32-Bit/64-Bit-Versionen, diverse Plattformen
- Administration:
  - Lokale Administration: SQLyog („Access-de-luxe“)
  - Beim Kunden: SQLyog in Kombination mit bspw. TeamViewer
- Backup (erzeugt Dump aus InnoDB):
  - `mysqldump –u USER -pPASSWORD --skip-opt --single-transaction --add-drop-table --create-options --quick --extended-insert --set-charset --disable-keys DB > LW:\...\Backup\backup.sql`
- Einfache Replikation (auch für Backup geeignet) [Anhang]
- Nachteile: Lizenzpreis + „fragliche Zukunft“? (Oracle)

## Migration von Access nach MySQL 1

- Texte + Tools zur Migration Access -> MySQL [Anhang]
- Ergebnisse bei Migration einer Access-DB (198 Tabellen):
  - **MySQL Migration Toolkit** (7 DB-Quellsysteme ab) [Demo]
    - Constraints (Beziehungen) werden nicht übertragen – auch nicht, wenn man die Systemtabellen wie empfohlen freischaltet (82/198)!
    - Index-Namen unbrauchbar (GUIDs?)

| Indexes | Columns | Index Type |
|---|---|---|
| PRIMARY | fkPer, fkThe | Unique |
| {1792B84C-772B-4F29-9D93-4CF14C | fkThe | |
| {6DD34FEA-1DBD-4DC9-8F17-EA1241 | fkPer | |

    - Collation (Sortierung) = *latin1_swedish_ci* anstatt *latin1_german2_ci*, wenn man nicht aufpasst; *utf8* vermeiden! [Anhang]
    - Feldkommentare werden nicht übertragen
  - **SQLyog**-Migrationsassistent – wie zuvor, aber:
    - Constraints (Beziehungen) werden übertragen
    - Feldkommentare werden übertragen

## Migration von Access nach MySQL 2

- **Bullzip** Access To MySQL
  - Große Probleme bei mehr als einem Datumsfeld + Default pro Tabelle (Hersteller empfiehlt Ausschluss) sowie bei Default-Eintrag für Kommentar-Memo -> Tabellen werden nicht importiert!
  - Constraints (Beziehungen) werden nicht übertragen
  - Indizes werden nicht übertragen
- **MS-Access** – Einzelne Tabellen per Anfügeabfrage übernehmen (MySQL-PKs sind aktualisierbar, Fremdschlüssel abschaltbar!)
  - Größtmögliche Kontrolle
  - Höchster Zeitaufwand
- **Empfehlung:** Struktur + Daten mit SQLyog übernehmen + ggf. überarbeiten. Alternativ „per Hand“ direkt auf Server erstellen, dann Daten-Übernahme einzelner Tabellen mit Access

## Datentypen: Access ⇔ MySQL [Anhang]

| Autowert | INT(10) |
|---|---|
| Text | (CHAR), VARCHAR |
| Memo | (VARCHAR !),TEXT/MEDIUMTEXT |
| Byte | TINYINT unsigned |
| Integer | SMALLINT |
| Long Integer | INT / INTEGER |
| Single | FLOAT |
| Double | DOUBLE |
| Dezimal | DECIMAL |
| Datum/Uhrzeit | DATETIME,TIMESTAMP |
| Währung | DECIMAL(19,4) |
| Ja/Nein | TINYINT(1) |
| OLE-Objekt | Medium-BLOB |
| Hyperlink | TEXT |

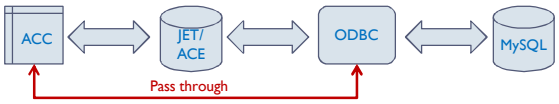
## Admin- bzw. Monitoring-Tools [Anhang]

- **AnySQL Maestro**: ab 99 USD, 5 Produkte (Trial)
- **DBACentral for MySQL**: interessant, leider veraltet (2007)
- **dbForge Studio for MySQL**: von 50 bis 200 USD; kostenlose Express-Version und Professional-Trial
- **EMS SQL Manager for MySQL**: ab 85 Euro, 9 Produkte (Trial)
- **HeidiSQL**: Open-Source-Projekt
- **MySQL:** (GUI-Tools), Workbench, (Enterpr. Monitor > 600 USD)
- **MySQL Front**: auf der Suche nach einer neuen Heimat …
- **Navicat for MySQL**: ab 95 USD, auch für MAC (30-Tage-Trial)
- **SQLyog**: von 99 bis 179 USD; 30-Tage-Trial; TIPP! [Demo]
- **MonYog**: von 139 bis 299 USD; Monitoring und Optimierungen im laufenden Betrieb (kostengünstige Alternative zum MySQL-Enterprise-Monitor)

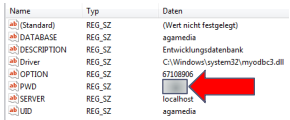
## Wichtige Voraussetzungen für Access 1

- Damit die Zusammenarbeit zwischen Access und MySQL reibungslos funktioniert, gelten folgende Bedingungen:
  - Jede Tabelle braucht einen Primärschlüssel
  - Jede Tabelle benötigt Timestamp (nicht NULL!) [Anhang]
  - **Tabellen**namen grundsätzlich in Kleinbuchstaben (für UNIX-Kompatibilität; Server erzwingt Lower Case)
  - **Feld**namen ohne Umlaute/ß oder Leerzeichen, reservierte Wörter (SQL) vermeiden, Groß-/Kleinschreibung erlaubt
  - Nach Änderung an einer Tabelle muss die betreffende Verknüpfung in Access gelöscht und neu eingebunden werden
  - Vor dem Import von Access-Objekten müssen die MySQL-Tabellen-Verknüpfungen erstellt werden

## Wichtige Voraussetzungen für Access 2



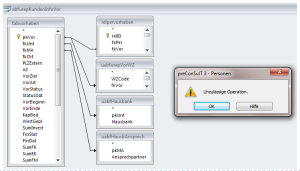

- Entwicklung: Eine ODBC-Datenquelle (DSN) [Anhang]
- Produktiv: Login via Access: DSN-less mit automatischer Tabellenverknüpfung [Anhang]
- User auf Server anlegen [Demo] (ggf. Gruppenkonzept in Access abbilden)



| Name | Typ | Daten |
|---|---|---|
| (Standard) | REG_SZ | (Wert nicht festgelegt) |
| DATABASE | REG_SZ | agamedia |
| DESCRIPTION | REG_SZ | Entwicklungsdatenbank |
| Driver | REG_SZ | C:\Windows\system32\myodbc3.dll |
| OPTION | REG_SZ | 67108906 |
| PWD | REG_SZ | [redacted] |
| SERVER | REG_SZ | localhost |
| UID | REG_SZ | agamedia |

## In der Praxis – „Wo liegt der Hase im Pfeffer?“

- Formulare zur Abbildung von n:m-Relationen
  - GetLastInsertID mit ADOExecuteSQL [Anhang]
  - (fk)Control.DefaultValue = xxx (in ufrm per Code vorbesetzen)
- Defaultwerte (Problem: erst auf Server geschrieben)
  - Als „Standardwert“ für Controls im Formular (Eigenschaften)
  - (Als „Standardwert“ für Feld in verknüpfter Tabelle)
- Abfrage mit mehrfachen Right-Joins auf Unterabfragen (als Datenbasis eines Reports wird sie dennoch ausgeführt!)



## Tipps & Tricks

- Pass-Through-Abfragen (aus VBA) [Anhang]
- Interessante MySQL-Funktionen [Anhang]
- Lösungen für Formate in Abfragen (Listboxen, Grid, TreeView)
  - Datum (z.B. Timestamp)

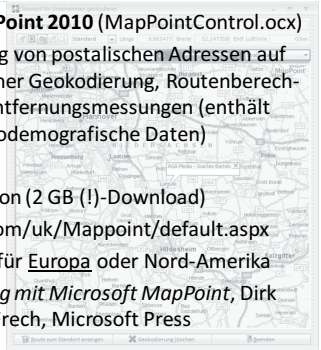
| ts |
|---|
| 2008-12-11 16:31:56 |
| 2008-12-11 16:32:11 |
| 2008-12-11 16:32:22 |
| CURRENT_TIMESTAMP |

    - In Acc-Abfrage Funktion DateValue (=> dt. DatWert): *DatumOhneZeit: DatWert([ts]) -> 10.03.2010*
    - FormatDateTime("10.03.1964",vbGeneralDate) -> 10.03.1964
    - FormatDateTime("10.03.1964",vbLongDate) -> Dienstag, 10. März 1964
    - FormatDateTime("10.03.1964 18:52:00",vbShortDate) -> 10.03.1964
    - FormatDateTime("10.03.1964 18:52:00",vbShortTime) -> 18:52
  - Währung

| Brutto | Mwst | Netto |
|---|---|---|
| 160.0000 | 25.5474 | 134.4500 |

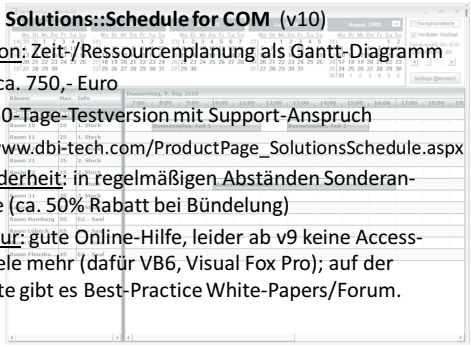
    - ABF: FormatCurrency(123.456, 2, vbTrue, vbFalse, vbTrue) -> 123,46 €
    - VBA – SQL-String: format([Brutto], '#,##0.00 €,)
    - ABF-Eigenschaft-Anzeigeformat: #.##0 €
    - Formular-Control-Format: Euro

## Professionalisierung 1: MapPoint [Demo]



- Name: **Microsoft MapPoint 2010** (MapPointControl.ocx)
- Funktion: Visualisierung von postalischen Adressen auf Karten mit automatischer Geokodierung, Routenberechnung, Fahrleitzonen, Entfernungsmessungen (enthält 750.000 POIs bzw. soziodemografische Daten)
- Preis: ca. 315,- Euro
- Trial: 60-Tage-Testversion (2 GB (!)-Download)
- Info: www.microsoft.com/uk/Mappoint/default.aspx
- Besonderheit: Version für Europa oder Nord-Amerika
- Literatur: *Geomarketing mit Microsoft MapPoint*, Dirk Grasekamp, Iris u. Kai Frech, Microsoft Press

## Professionalisierung 2: ctSchedule [Demo]



- Name: **Solutions::Schedule for COM** (v10)
- Funktion: Zeit-/Ressourcenplanung als Gantt-Diagramm
- Preis: ca. 750,- Euro
- Trial: 30-Tage-Testversion mit Support-Anspruch
- Info: www.dbi-tech.com/ProductPage_SolutionsSchedule.aspx
- Besonderheit: in regelmäßigen Abständen Sonderangebote (ca. 50% Rabatt bei Bündelung)
- Literatur: gute Online-Hilfe, leider ab v9 keine Access-Beispiele mehr (dafür VB6, Visual Fox Pro); auf der Website gibt es Best-Practice White-Papers/Forum.

## Professionalisierung 3: sevDataGrid2 [Demo]

- Name: **sevDataGrid2**
- Funktion: Flexibles DataGrid und ListView-Control mit XLS-Exportfunktionen und integriertem WebReport
- Preis: ca. 139,- Euro
- Trial: Shareware als Download (enthält Beispiel-MDBs)
- Info: www.tools4vb.com/software/sevdatagrid.php
- Besonderheit: Für AEK-Teilnehmer mit Gutscheincode **25% Rabatt** vom 25.09.-31.12.2010! Exzellenter Support!
- Literatur: sehr gute Online-Hilfe mit vielen Acc-Beispielen

## Professionalisierung 4: GlobalErrorHandler

- Name: **Simply VBA Global Error Handler** (vbWatchDog)
- Funktion: Fängt + protokolliert sämtliche VBA-Fehler (auch) ohne lokalen Error-Handler
- Preis: ca. 180,- Euro (neu: per Entwickler)
- Trial: Beispiel-MDB mit kommentiertem Code
- Info: www.everythingaccess.com
- Besonderheit: vbWatchDog läuft ohne DLLs auf Basis von 4 Klassenmodulen
- Literatur: sehr gute Onlinehilfe, guter Support