

Microsoft Access Projekte in der Praxis

(c) Uwe Ricken
MCP for SQL Server 2000 Database Implementation

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

db-Berater

Wer bin ich? Was mache ich?

- Seit 1991 als freiberuflicher Dozent und Entwickler tätig
- Von 1998 – 2007 als geschäftsführender Gesellschafter der GNS GmbH für den Bereich „Entwicklung“ tätig
- Datenbankentwicklung auf Basis Microsoft SQL-Server seit 1999 (SQL-Server 6.5 / 7.0 / 2000 / 2005)
- 2001 Zertifizierung zum „MCP for SQL Server 2000 Database Implementation“
- Seit 2007 mit db-Berater wieder selbständig
 - Dienstleistungen rund um den SQL-Server (Entwicklung/Optimierung)
 - Netzwerkintegration
- Autor mehrerer Fachpublikationen im Bereich Microsoft Windows / Microsoft Access
- Entwicklung von zwei Standardprodukten
 - conferenceadminVerwaltung von Besprechungen und Protokollen (D / CH / AU)
 - memberadminVeranstaltungen / Mitgliederverwaltung (D / CH / FR / BE / PL)

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

db-Berater

Agenda

- Unterschied zwischen ADP und MDB
 - Was ist ein Microsoft Access Database Project
 - Zugriff auf die Daten des Microsoft SQL-Server
 - Einschränkungen bei der Arbeit mit einer ADP-Datei
 - Vor- und Nachteile der Arbeit mit ADP
 - Bearbeiten von Objekten des SQL-Servers (V. 2000 / 2005)
- Verbindungen zwischen ADP und SQL-Server
 - statische Verbindung
 - dynamische Verbindungszeichenfolgen
 - Verbindung mittels UDL (universal data link)
 - Qualifizierter Zugriff auf Objekte des SQL-Servers
- Das Kreuz mit den Kreuztabellen
 - statische Kreuztabellen (V. 2000 / 2005)
 - dynamische Kreuztabellen
 - Dynamik kennt Grenzen (Änderungen an Objekten registrieren)

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

db-Berater

Agenda

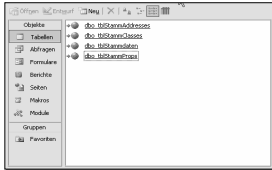
- **Automatik oder Hand(schaltung)**
 - Verwendung von Autowerten (Identity)
 - Tücken der Autowerte bei der Verwendung von Triggern
 - Trigger und Primary Keys/Unique Indizes
- **Temporäre Objekte**
 - Welche Arten von temporären Objekte gibt es
 - Unterschied zwischen „lokalen“ und „globalen“ Temporärobjekten
- **Dynamisches SQL**
 - Dynamik und EXEC() – Vor- und Nachteile
 - Eine echte Alternative: sp_executeSQL
- **Das Problem der „eindeutigen Tabelle“ - UniqueTable**
- **Das Datum: Fluch und Segen**
 - Probleme bei der Verwendung von Datumsangaben
 - Viele Formate aber eine Lösung
- **Sonstiges**
 - Warum SET NOCOUNT
 - Multiple Recordsets

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

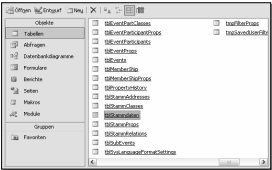
cb-Berater

Unterschied zwischen MDB und ADP

Eingebundenen Objekte in MDB



Direkter Zugriff auf Objekte in ADP



- Tabellen werden mittels ODBC eingebunden
- Relationen können wie ACCESS-Tabellen behandelt aber nicht bearbeitet werden
- VIEWS werden wie Relationen behandelt
- Zugriff auf Relationen/Views mittels JET

- Unmittelbarer Zugriff auf SQL-Objekte mittels OLEDB
- Auf Relationen kann direkt zugegriffen werden und Objekte können bearbeitet werden (gilt nicht für SQL2005)
- Relationen werden von Views/Stored Procedures/Functions (Abfragen) getrennt angezeigt
- Zugriff auf Objekte mittels ADO

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Unterschied zwischen MDB und ADP

- **Zugriffsmöglichkeiten (ADP)**
 - Technologie ist ausschließlich auf den SQL-Server anwendbar
 - Zugriff auf SQL-Server erfolgt direkt (keine Einbindung von Objekten)
 - Zugriff auf Objekte erfolgt mittels ADO (Active Data Objects)
- **Klare Trennung von Daten- und Benutzerschnittstelle**
 - Daten (Relationen / Abfragen (Views) / Stored Procedures / Funktionen) sind Objekte des SQL-Server
 - Formulare / Berichte / Makros / Module sind Objekte der Benutzerschnittstelle
 - Sicherheitseinstellungen für Daten und Datenbankobjekte werden in SQL-Server verwaltet
- **Keine Möglichkeit der Kombination von Microsoft Access Datenbanken und SQL-Server-Objekte**
 - temporäre Objekte (Relationen) MÜSSEN auf dem SQL-Server angelegt werden
 - Verwendung von SQL-Relationen UND JET-Objekten ist nicht möglich.
- **Bearbeitung von SQL-Server Objekte**
 - Bis SQL-Server 2000 war eine Bearbeitung von Objekten mittels Microsoft Access möglich
 - Ab SQL-Server 2005 muß das SQL-Server Management Studio oder Microsoft Access 2007 verwendet werden.

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007


cb-Berater

SEK1, Nürnberg, 21./22.4.2007

2

Verbindungen zwischen ADP und SQL-Server

Demonstration für die Erstellung eines dynamischen Connectionstrings



1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Verbindungen zwischen ADP und SQL-Server

Option Compare Database
Option Explicit

```
Private Const CONNECTION_STRING As String = "Provider=SQLOLEDB;Data
Source=$Servername;User ID=$Username;Password=$Password;Initial
Catalog=MemberAdminPL; "
```


```
Private Sub cmdLogin_Click()
Dim ConnectionString As String
ConnectionString = Replace(CONNECTION_STRING, "$Servername,, Me!txtSRVname)
ConnectionString = Replace(ConnectionString, "$Username", Me!txtUserName)
ConnectionString = Replace(ConnectionString, "$Password", Me!txtPassword)
CurrentProject.OpenConnection ConnectionString
End Sub
```

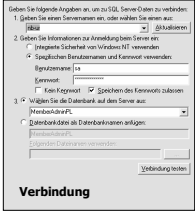
1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Verbindungen zwischen ADP und SQL-Server

- **Verbindung mittels UDL (universal data link)**
 - UDL ist eine reine Textdatei
 - UDL wird über „Dateieigenschaften“ bearbeitet
- **Konfigurieren einer UDL**
 - Anlegen einer neuen Datei mit der Dateieindung *.udl
 - Auswahl von [Eigenschaften] aus dem Kontextmenü für Dateien
 - **Verbindungszeichenfolge wird in Klartext gespeichert (wie DSN!)**



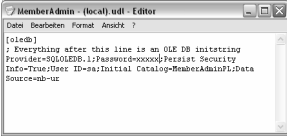


1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Verbindungen zwischen ADP und SQL-Server

- **Verbindung mittels UDL (universal data link)**
 - UDL ist eine reine Textdatei
 - UDL wird über „Dateieigenschaften“ bearbeitet
- **Konfigurieren einer UDL**
 - Anlegen einer neuen Datei mit der Dateiendung *.udl
 - Auswahl von [Eigenschaften] aus dem Kontextmenü für Dateien
 - **Verbindungszeichenfolge wird in Klartext gespeichert (wie DSN!)**




1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Verbindungen zwischen ADP und SQL-Server

Demonstration für die Verbindung zu SQL-Datenbank mittels UDL-Datei



1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Verbindungen zwischen ADP und SQL-Server

```
Option Compare Database
Option Explicit

Private cnn As Object
Private Sub cmdLogin_Click()
    Dim ConnectionString As String
    If Me!txtServername.Caption <> vbNullString Then
        Set cnn = CreateObject("ADODB.Connection")
        With cnn
            ConnectionString = "File Name=" & Me!UDLFileName.Caption
            .ConnectionTimeout = 0
            .CommandTimeout = 0
            .Open ConnectionString
            CurrentProject.OpenConnection .ConnectionString
        End With
    End If
End Sub
```

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Qualifizierter Zugriff auf Objekte des SQL-Server (<=2000)

- Objekte sollten **IMMER** voll qualifiziert sein [Eigentümer].[Objekt]

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Qualifizierter Zugriff auf Objekte des SQL-Server (2005)

- Besonderheiten bei der Verwendung von Schemas**
 - Sofern für einen Benutzer (Principal) kein Schema benannt wird, ist das Standardschema [dbo]
 - Bei Verwendung eines Schemas ist keine Referenzierung des Schemas erforderlich
 - Sollten Objekte mit gleichen Namen vorhanden sein, gilt folgende Suchreihenfolge:

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Qualifizierter Zugriff auf Objekte des SQL-Server

- Prozeduren möglichst nie mit dem Präfix **sp_** beginnen lassen

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Das Kreuz mit den Kreuztabellen

- Kreuztabellen a' la Microsoft Access JET gibt es im SQL-Server nicht
- Kreuztabellen im SQL-Server sind „by design“ nicht dynamisch
- Syntaxunterschiede zwischen Microsoft SQL-Server <= 2000 und Microsoft SQL-Server 2005

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Das Kreuz mit den Kreuztabellen

```
/*      Beispiel für die Erstellung einer PIVOT-Tabelle
gemäß Konventionen von Microsoft Access (JET)
*/

-- Ohne vordefinierte Spaltenangabe
TRANSFORM Sum(IsMember) AS SummevonIsMember
SELECT  Year([MemberSince]) AS Jahr
FROM    dbo.view_Members_And_MemberTypes
GROUP BY Year([MemberSince])
PIVOT   TOM;

-- mit vordefinierter Spaltenangabe
TRANSFORM Sum(IsMember) AS SummevonIsMember
SELECT  Year([MemberSince]) AS Jahr
FROM    dbo.view_Members_And_MemberTypes
GROUP BY Year([MemberSince])
PIVOT   TOM IN ('Corporate', 'Individual', 'Honorary')
```

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Das Kreuz mit den Kreuztabellen

```
/*      Beispiel für die Erstellung einer PIVOT-Tabelle
mit Syntax des SQL-Server <= 2000
*/

SELECT  YEAR(MemberSince) AS Year,
SUM(CASE WHEN [TOM] = 'Corporate' THEN IsMember ELSE 0 END) AS C,
SUM(CASE WHEN [TOM] = 'Individual' THEN IsMember ELSE 0 END) AS I,
SUM(CASE WHEN [TOM] = 'Honorary' THEN IsMember ELSE 0 END) AS H
FROM    dbo.view_Members_And_MemberTypes
GROUP BY
    Year(MemberSince)
```

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Das Kreuz mit den Kreuztabellen

```
/*
    Beispiel für die Erstellung einer PIVOT-Tabelle
    mit Syntax des SQL-Server 2005
*/

SELECT  YEAR(MemberSince),
        SUM (pvt.Corporate) AS C,
        SUM (pvt.Individual) AS I,
        SUM (pvt.Honorary) AS H
FROM    dbo.view_Members_And_MemberTypes
PIVOT (
    SUM(ISMember)
    FOR TOM IN ([Corporate], [Individual], [Honorary])
) AS PVT
GROUP BY
    YEAR(MemberSince)
```

dynamische Kreuztabellen ...

- ... gibt es nicht!
- ... können nur simuliert werden
- ... Verwendung von T-SQL-Syntax unabdingbar
- ... Kapselung der Erstellung in Stored Procedure
- ... Möglichkeiten von Views und oder Functions sehr stark eingeschränkt

dynamische Kreuztabellen (ein Beispiel) #1

```
CREATE PROC dbo.proc_sek1_CreateDynamicPivotTable
AS
    SET NOCOUNT ON

    DECLARE @DynSQL      varchar(4000)
    DECLARE @Year        int
    DECLARE @ColumnName  varchar(100)
    DECLARE @Counter     int
    DECLARE @ReturnValue  int

    -- Existiert die Ausgabetable bereits
    IF EXISTS (
        SELECT  *
        FROM    dbo.sysObjects
        WHERE   Id = OBJECT_ID('tmp_DynamicPivotTable', 'U')
    )
    BEGIN
        SET @DynSQL = 'DROP TABLE tmp_DynamicPivotTable'
        EXEC (@DynSQL)
    END
```

dynamische Kreuztabellen (Ein Beispiel)

#2

```
-- Erstellen des Grundgerüsts mittels SQL
SET @DynSQL = 'CREATE TABLE tmp_DynamicPivotTable([YEAR] int, '

DECLARE c CURSOR
FOR
    SELECT DISTINCT TOM
    FROM    dbo.view_Members_And_MemberTypes

OPEN c
FETCH NEXT FROM c INTO @ColumnName
WHILE (@@FETCH_STATUS <> -1)
BEGIN
    SET @DynSQL = @DynSQL + @ColumnName + ' int DEFAULT (0),'
    FETCH NEXT FROM c INTO @ColumnName
END

CLOSE c
DEALLOCATE c
```

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

dynamische Kreuztabellen (Ein Beispiel)

#3

```
-- Entfernen des letzten Fieldseparators
SET @DynSQL = LEFT(@DynSQL, LEN(@DynSQL) - 1)
SET @DynSQL = @DynSQL + ' '

-- Anlegen der Tabelle
EXEC (@DynSQL)

-- Befüllen der Tabelle
EXEC ('INSERT INTO tmp_DynamicPivotTable ([YEAR]) SELECT DISTINCT
YEAR(MemberSince) FROM dbo.view_Members_And_MemberTypes')

DECLARE c CURSOR FOR
SELECT  YEAR(MemberSince),
        TOM,
        SUM(IsMember)      AS Members
FROM    dbo.view_Members_And_MemberTypes
GROUP BY
        YEAR(MemberSince),
        TOM
```

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

dynamische Kreuztabellen (Ein Beispiel)

#3

```
OPEN c
FETCH NEXT FROM c INTO @Year, @ColumnName, @Counter
WHILE (@@FETCH_STATUS <> -1)
BEGIN
    SET @DynSQL = 'UPDATE tmp_DynamicPivotTable SET ' + @ColumnName
+ ' = ' + CONVERT(varchar, @Counter) + ' WHERE [YEAR] = ' +
CONVERT(varchar, @Year)
    EXEC (@DynSQL)
    FETCH NEXT FROM c INTO @Year, @ColumnName, @Counter
END
CLOSE c
DEALLOCATE c

SET    @ReturnValue = 1

RETURN ISNULL(@ReturnValue, 0)
SET NOCOUNT OFF

GO
```

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

dynamische Kreuztabellen (Ein Beispiel)

#4

Analyse

Mitgliederanalyse

analyse durchführen

Drucken

Exportieren

Jahr	Corporate	Individual
1991	26	0
1992	17	1
1993	15	1
1994	17	0
1995	13	0
1996	10	1
1997	12	0
1998	6	1
1999	10	1
2000	7	1
2001	13	2

1. SQL-Server Entwickler Konferenz

Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Dynamische Elemente und deren Tücken

- Szenario:**
 - Objekt (wird später zur Laufzeit erstellt!) wird temporär erstellt und an Microsoft Access Objekt gebunden (z. B. Bericht)
 - Nachdem der Bericht fertiggestellt ist, wird das temporäre Objekt wieder gelöscht
 - Bei Ausführung des Berichts erfolgt Fehlermeldung:
- Ursache:**
 - Microsoft Access aktualisiert den Objektcontainer (Tables) nur in Intervallen
- Lösung**
 - Temporäre Objekte immer mittels `SELECT * FROM [Eigentümer].[Objekt]` binden!

Format | Daten | Ereignis | Anders | Abk

Datenherkunft: ...Temp_DynamicPivotTable

Datenherkunftsberechner: ...

Filter: ...

Filter aktiv: ...

Sortiert nach: ...

Die auf diesem Formular oder Bericht angegebene Datenherkunft Temp_DynamicPivotTable ist nicht vorhanden.

Die Daten Ihres Berichts sind gestrichelt, oder Sie wurde in der aktuellen Datenbank gelöscht oder umbenannt. Überprüfen Sie die Datenbank, um sicherzustellen, dass die Datenherkunft der Formeln oder Berichte den Namen auf der Schaltfläche Eigenschaften der Datenherkunftsberechner aus und geben Sie die Datenherkunftsberechner den entsprechenden Namen an.

OK

Format | Daten | Ereignis | Anders | Abk

Datenherkunft: ...SELECT * FROM Temp_DynamicPivotTable

Datenherkunftsberechner: ...

Filter: ...

Filter aktiv: ...

Sortiert nach: ...

1. SQL-Server Entwickler Konferenz

Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Autowerte / Primary Keys und Unique Keys in Verbindung mit Triggern

- Änderung von eindeutigen Attributen über Trigger verursachen Fehlermeldung in Microsoft Access**
 - Autowerte

```
CREATE TABLE dbo.myFoo (Id IDENTITY (1, 1))
```
 - Clustered Indexes

```
CREATE TABLE dbo.myFoo (SortName PRIMARY KEY CLUSTERED)
```
 - Unique Indexes

```
CREATE TABLE dbo.myFoo (AddressMatchCode UNIQUE CLUSTERED)
```
- mögliche Fehlermeldungen**
 - Ändern von Primär- und Uniqueschlüsseln

Die Daten wurden zur Datenbank hinzugefügt, aber sie können nicht im Formular angezeigt werden, weil sie den Kriterien der Datenquelle nicht entsprechen.

Hilfe anzeigen >>

OK
 - Ändern von Identity-Attributen in Referenztabellen
 - durch Trigger

1. SQL-Server Entwickler Konferenz

Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

SEK1, Nürnberg, 21./22.4.2007

10

Autowerte / Primary Keys und Unique Keys in Verbindung mit Triggern

	Id	Sid	Description	RequestType	RequestNo	RequestDate	Requestor	SendDate	SendBy	ConfirmedBy
F	(AutoWert)		#Gelöscht Das ist ein Test #Gelöscht	#Gelöscht	#Gelöscht	#Gelöscht	#Gelöscht	#Gelöscht	#Gelöscht	#Gelöscht
F	(AutoWert)		AN	0000-0000	06.04.2007					

Microsoft Office Access

Die Daten wurden zur Datenbank hinzugefügt, aber sie können nicht in Formulare angezeigt werden, weil sie den Kriterien der Datenabfrage nicht entsprechen.

Hilfe anzeigen >>>

OK

1. SQL-Server Entwickler Konferenz

Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Autowerte / Primary Keys und Unique Keys in Verbindung mit Triggern

```
CREATE TABLE dbo.tbl_app_ServiceRequest
(
    Id            int            NOT NULL IDENTITY(1, 1),
    Sid           int            NOT NULL,
    Description    varchar(2000) NOT NULL,
    RequestType   char(2)        NOT NULL DEFAULT ('AN'),
    RequestNo     varchar(9)     NOT NULL DEFAULT ('0000-0000'),
    ...,

    CONSTRAINT pk_tbl_app_ServiceRequest PRIMARY KEY NONCLUSTERED
        (Id),

    CONSTRAINT fk_tbl_app_Stammdaten FOREIGN KEY (Sid)
        REFERENCES dbo.tblStammdaten (Sid),

    CONSTRAINT idx_tbl_app_ServiceRequest UNIQUE CLUSTERED
        (RequestType, RequestNo)
)
```

1. SQL-Server Entwickler Konferenz

Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Autowerte / Primary Keys und Unique Keys in Verbindung mit Triggern

```
CREATE TRIGGER dbo.trg_tbl_app_ServiceRequest_Insert
ON dbo.tbl_app_ServiceRequest
FOR INSERT
AS
    DECLARE @RequestCounter int
    DECLARE @RequestNumber  varchar(9)

    -- Ermittlung der nächsten Requestnummer (basierend auf RequestType)
    SELECT @RequestCounter = ISNULL(COUNT(r.Id), 0) + 1
    FROM    inserted i INNER JOIN dbo.tbl_app_ServiceRequest r
        ON (i.RequestType = r.RequestType)
    WHERE   SUBSTRING(r.RequestNo, 6, 4) = YEAR(getdate())

    SET @RequestNumber = REPLICATE('0', 4 - LEN(@RequestCounter)) +
        CONVERT(varchar(4), @RequestCounter)
    SET @RequestNumber = @RequestNumber + '-' + CONVERT(varchar(4),
        YEAR(getdate()))

    UPDATE r
    SET    RequestNo = @RequestNumber
    FROM    inserted i INNER JOIN dbo.tbl_app_ServiceRequest r
        ON (i.Id = r.Id)
```

1. SQL-Server Entwickler Konferenz

Nürnberg 21.04.2007 – 22.04.2007

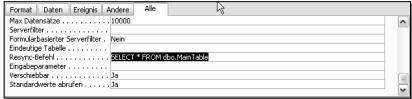
cb-Berater

SEK1, Nürnberg, 21./22.4.2007

11

Lösung: ResyncCommand (Formulareigenschaft)

- ResyncCommand führt nach Aktualisierung der Daten ein Requery durch



1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Dynamisches SQL und seine Tücken

- Dynamisches SQL bedeutet ein Sicherheitsrisiko
 - Berechtigungen auf Objekte müssen vergeben werden, da der Prozeß immer im Benutzerkontext abläuft
 - Beeinflussung von SQLStatements durch „SQL-Injection“
Artikel von Erland Sommarskog (ins Deutsche übersetzt von Frank Kalis)
http://www.insidesql.de/beltraege/dev_basics/es_dynamisches_sql_fluch_und_seggen.html
- Dynamisches SQL braucht Zeit
 - Es werden keine Abfragepläne/Ausführungspläne gespeichert
- Zwei Möglichkeiten der Verwendung von „Dynamisches SQL“
 - EXEC()
 - sp_ExecuteSQL (ab SQL-Server 7.0!)

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Die Eigenschaft „UniqueTable“ und ihre Tücken

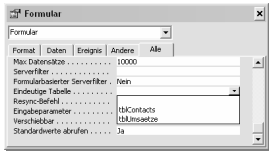
- Formulare, denen als Datengrundlage eine Abfrage mit mehr als einer Relation zu Grunde liegen, können nur bearbeitet werden, wenn eine eindeutige Relation angegeben wird
- Die Eigenschaft „UniqueTable“ des Formulars beinhaltet den Namen der zu bearbeitenden Datengrundlage
- Problem:
 - Je nach angemeldeten Benutzer MUSS der volle Qualifier des Eigentümers der Relation mit angegeben werden
- Es ist nicht möglich, einheitlich Owner.Objekt als UniqueTable-Eigenschaft zu definieren!

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Die Eigenschaft „UniqueTable“ und ihre Tücken

- Beispiel für eine Auswahl bei Anmeldung als „dbo“



Die Eigenschaft „UniqueTable“ und ihre Tücken

- Beispiel für eine Auswahl bei Anmeldung als „Benutzer“



Lösung mittels Stored Procedure #1

```
CREATE PROC dbo.sysGetUniqueTableString
    @ObjectName varchar(40),
    @UniqueTable varchar(100) OUTPUT
AS
    SET NOCOUNT ON
    DECLARE @ObjectId int
    SET @ObjectId = OBJECT_ID(@ObjectName)

    IF (@ObjectId IS NULL)
    BEGIN
        RAISERROR ('Objekt %s existiert nicht', 16, 1, @ObjectName)
        GOTO Move_ToExit
    END

    IF EXISTS (SELECT TOP 1 o.Name
        FROM dbo.sysObjects o INNER JOIN dbo.sysUsers u
            ON (o.uid = u.uid)
        WHERE o.id = @ObjectId AND u.Name = user_name()
    )
        SET @UniqueTable = @ObjectName
```

Lösung mittels Stored Procedure

#1

```
CREATE PROC dbo.sysGetUniqueTableString
    @ObjectName varchar(40),
    @UniqueTable varchar(100) OUTPUT
AS
    SET NOCOUNT ON
    DECLARE @ObjectId int
    SET @ObjectId = OBJECT_ID(@ObjectName)

    IF (@ObjectId IS NULL)
    BEGIN
        RAISERROR ('Objekt %s existiert nicht', 16, 1, @ObjectName)
        GOTO Move_ToExit
    END

    IF EXISTS (SELECT TOP 1 o.Name
        FROM dbo.sysObjects o INNER JOIN dbo.sysUsers u
            ON (o.uid = u.uid)
        WHERE o.id = @ObjectId AND u.Name = user_name()
    )
        SET @UniqueTable = @ObjectName
```

1. SQL-Server Entwickler Konferenz

Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Lösung mittels Stored Procedure

#2

```
ELSE
    SELECT @UniqueTable = user_name(o.uid) + '.' + @ObjectName
    FROM dbo.sysObjects o
    WHERE id = @ObjectId

Move_ToExit:
    SET NOCOUNT OFF
```

1. SQL-Server Entwickler Konferenz

Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Das Datum – Fluch und Segen

- Datum als Parameter ist abhängig von Ländereinstellungen des Benutzers auf der Clientseite
- Datum als Parameter ist abhängig von Spracheinstellungen des Benutzers auf der Serverseite
- Ziel: Suchen einer „gemeinsamen“ Einstellung, die unabhängig von Einstellungen ist
 - ISO-Format (yyyymmdd hh:mm)

1. SQL-Server Entwickler Konferenz

Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Beispiele für die generische Verwendung von ISO-Formaten

```
DECLARE @SearchDate datetime
SET @SearchDate = '19940401'

SELECT @@LANGUAGE

SET LANGUAGE 'deutsch'
-- Einstellungen DEUTSCH
SELECT @SearchDate, * FROM dbo.tblStammProps
WHERE Pid = 54 AND
      PDate = @SearchDate

SET LANGUAGE 'us_english'
SELECT @SearchDate, * FROM dbo.tblStammProps
WHERE Pid = 54 AND
      PDate = @SearchDate

SET LANGUAGE 'polski'
SELECT @SearchDate, * FROM dbo.tblStammProps
WHERE Pid = 54 AND
      PDate = @SearchDate
```

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Multiple RecordSets

```
CREATE PROC dbo.proc_app_MultiRecords
AS
    SELECT * FROM dbo.tbl_app_Projects

    SELECT * FROM dbo.tbl_app_ServiceRequest

    SELECT * FROM dbo.tbl_app_ServiceRequest_Details
GO
```

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater

Multiple RecordSets – Zugriff mittels VB(A)

```
Public Sub fn_app_MultiRecords()
    On Error GoTo ErrorHandler

    Set cmd = New ADODB.Command
    With cmd
        Set .ActiveConnection = CurrentProject.Connection
        .CommandType = adCmdStoredProc
        .CommandText = "dbo.proc_app_MultiRecords"

        Set rst = .Execute()
    End With

    Do While Err = 0
        While Not rst.EOF
            Debug.Print rst.Fields(0)
            rst.MoveNext
        Wend
        Set rst = rst.NextRecordset
    Loop
```

1. SQL-Server Entwickler Konferenz
Nürnberg 21.04.2007 – 22.04.2007

cb-Berater
