db Berater GmbH
Datenbanken • Netzwerke • Beratung • Service

Sicherheitskonzept von SQL Server 2005 SQL Server 2008

db Berater GmbH

Uwe Ricken

Microsoft CERTIFIED IT Professional
Database Administrator
Database Administrator 2008

Microsoft CERTIFIED Technology Specialist
SQL Server 2005
SQL Server 2008, Implementation and Maintenance

Agenda

Sicherheitskonzept des SQL Server

SQL Server Dienste	20 min
Serverzugriff	10 min
Datenbankzugriff	10 min

Sicherheitskonzept in Datenbanken

Serverrollen (Logins)	10 min
Datenbankrollen (User)	10 min
Schemata	15 min
Principals	15 min
Proxies	15 min

Verschiedenes

xp_cmdshell	15 min
sp_executeSQL / EXEC()	10 min
Identitätswechsel	10 min
Linked Server	15 min

Sicherheitsdefinition

Authentication

Unter [**Authentication**] wird der Prozess verstanden, mit dem Microsoft SQL Server die Identität einer Anforderung überprüft und und einrichtet.

Authorization

Unter [**Authorization**] wird der Prozess verstanden, unter dem Microsoft SQL Server einer authentifizierten Identität Zugriff auf Ressourcen gewährt oder verweigert.

SQL Server Dienste

Welche Dienste verwendet SQL Server?

SQL Server:	Database Engine (Verwaltung von Datenbanken)
SQL Server Agent:	Job Engine (Ausführung von zeitgesteuerten Aufträgen)
SQL Server Browser:	Listener für Anfragen in Multi-Instanzen-Umgebung Weiterleitung von Anfragen an richtige Instanz Verbindung von DAC (Dedicated Admin Connections
SQL Server Integration Serv.:	Datenmigration
SQL Server Reporting Serv.:	Webbasiertes Reporting
SQL Server Analysis Serv.:	Data-Mining / OLAP

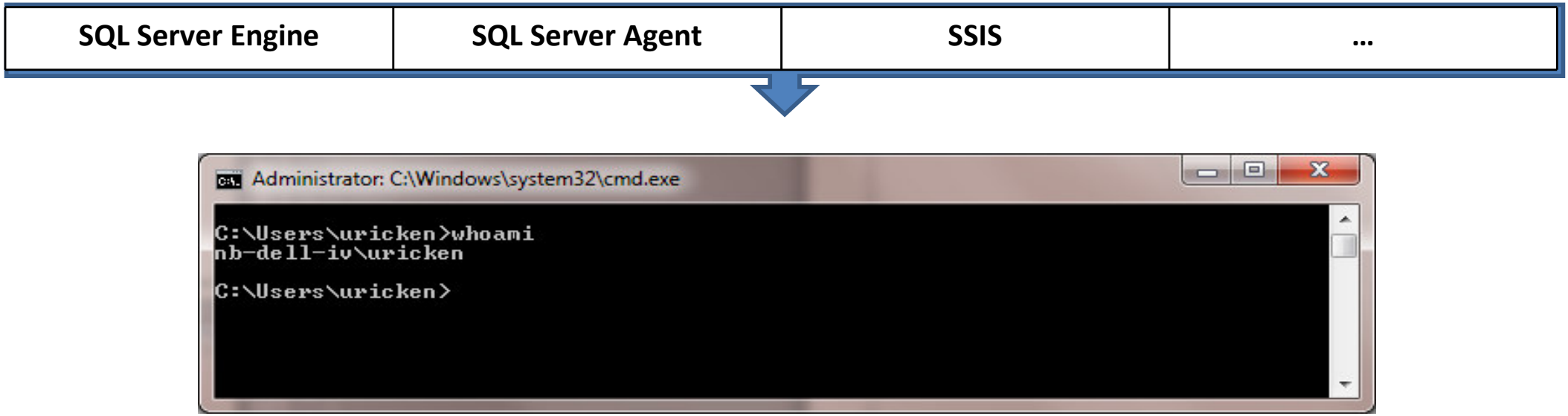
Name	Status	Startmodus	Anmelden als	Prozess-ID	Diensttyp
SQL Server Integration Services 10.0	Wird ausgeführt	Automatisch	._sqlserver	1392	SSIS Server
SQL Full-text Filter Daemon Launcher (MSSQLSERVER)	Wird ausgeführt	Manuell	NT AUTHORITY\LOCALSERVICE	3316	
SQL Server (MSSQLSERVER)	Wird ausgeführt	Automatisch	._sqlserver	2072	SQL Server
SQL Server Analysis Services (MSSQLSERVER)	Beendet	Manuell	._sqlserver	0	Analysis Server
SQL Server Reporting Services (MSSQLSERVER)	Wird ausgeführt	Automatisch	._sqlserver	2516	ReportServer
SQL Server Browser	Beendet	Andere (Boot, Syste...	NT AUTHORITY\LOCALSERVICE	0	SQL Browser
SQL Server-Agent (MSSQLSERVER)	Beendet	Manuell	._sqlserver	0	SQL Agent

Warum Dienstkonten?

- Dienstkonten steuern die **kontrollierte** Zugriffsberechtigung des Prozesses auf Ebene des Betriebssystems

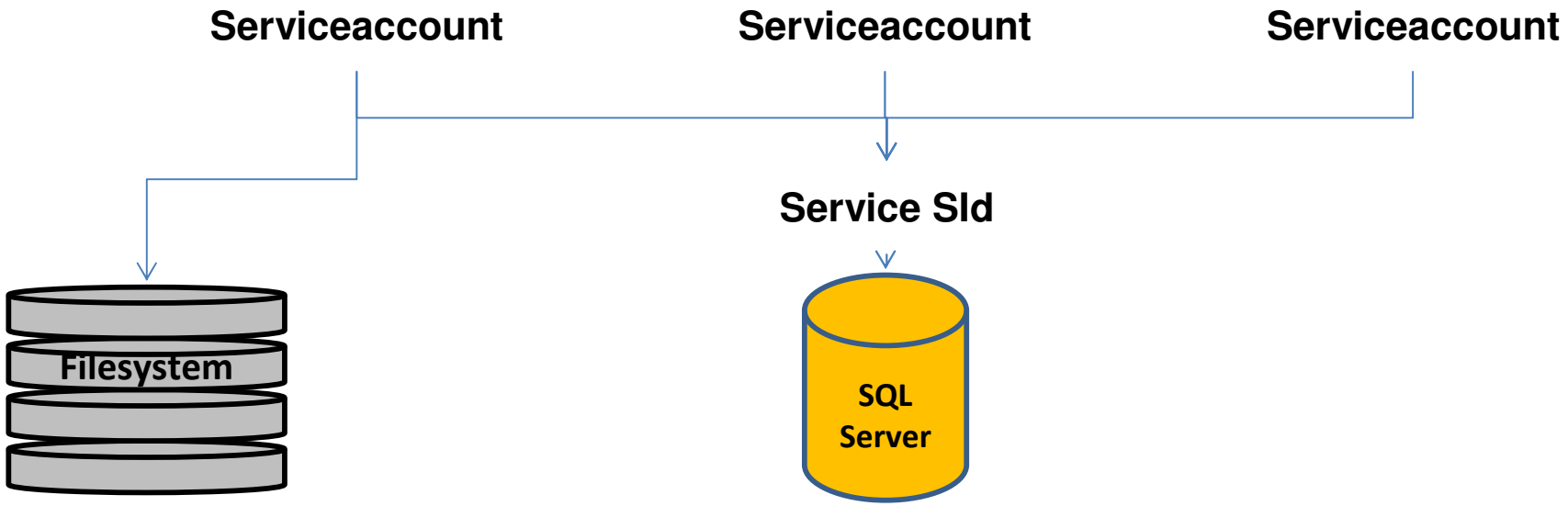
- Berechtigungen auf Ebene des Betriebssystems können ausschließlich an Computer-/User-/Gruppenobjekte erteilt werden

- Getrennte Verwaltung von Logins und Applikationen (SQL Server)



Neu ab SQL Server 2008

- Per Standard wird die lokale Windows-Gruppe BUILTIN\Administrators nicht mehr der festen Servergruppe „sysadmin“ zugewiesen
- Die Windowsgruppen für die Verwendung von SQL Server (siehe nächste Folie) sind nicht mehr der festen Servergruppe „sysadmin“ zugewiesen
- Dem „Dienstkonto“ werden explizit „sysadmin“-Berechtigungen zugeteilt
- Wird SQL Server unter dem Betriebssystem „Windows Server 2008“ oder Vista installiert, wird eine „Dienst-Sid“ als Element der festen Serverrolle „sysadmin“ bereitgestellt.



Standardgruppen

Bei der Installation von Microsoft SQL Server werden Standardgruppen anlegt

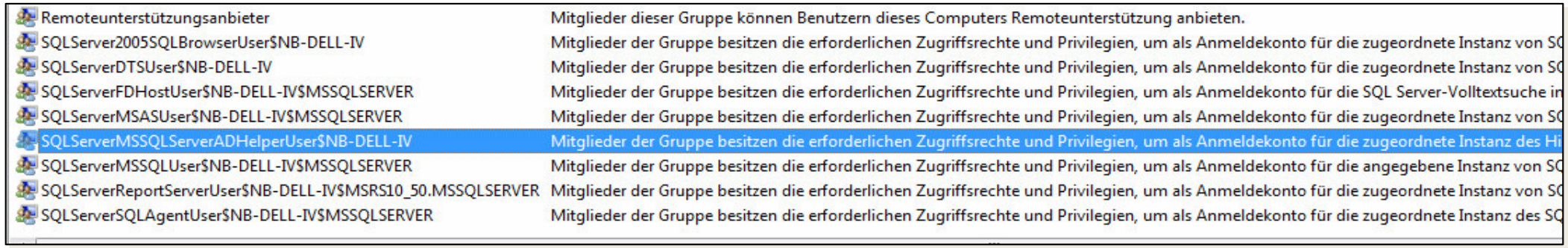
Die Standardgruppen werden IMMER auf dem lokalen Server angelegt

Die Nomenklatur unterliegt festen Regeln:

SQLServer[Service]$[ComputerName]$[InstanceName]

Ausnahmen:

Instanzunabhängige Dienste (SQL Browser, AD Zugriff) werden nur einmal gelistet!



Remoteunterstützungsanbieter	Mitglieder dieser Gruppe können Benutzern dieses Computers Remoteunterstützung anbieten.
SQLServer2005SQLBrowserUser$NB-DELL-IV	Mitglieder der Gruppe besitzen die erforderlichen Zugriffsrechte und Privilegien, um als Anmeldekonto für die zugeordnete Instanz von SQ
SQLServerDTSUser$NB-DELL-IV	Mitglieder der Gruppe besitzen die erforderlichen Zugriffsrechte und Privilegien, um als Anmeldekonto für die zugeordnete Instanz von SQ
SQLServerFDHostUser$NB-DELL-IV$MSSQLSERVER	Mitglieder der Gruppe besitzen die erforderlichen Zugriffsrechte und Privilegien, um als Anmeldekonto für die SQL Server-Volltextsuche in
SQLServerMSASUser$NB-DELL-IV$MSSQLSERVER	Mitglieder der Gruppe besitzen die erforderlichen Zugriffsrechte und Privilegien, um als Anmeldekonto für die zugeordnete Instanz von SQ
SQLServerMSSQLServerADHelperUser$NB-DELL-IV	Mitglieder der Gruppe besitzen die erforderlichen Zugriffsrechte und Privilegien, um als Anmeldekonto für die zugeordnete Instanz des Hi
SQLServerMSSQLUser$NB-DELL-IV$MSSQLSERVER	Mitglieder der Gruppe besitzen die erforderlichen Zugriffsrechte und Privilegien, um als Anmeldekonto für die angegebene Instanz von SQ
SQLServerReportServerUser$NB-DELL-IV$MSRS10_50.MSSQLSERVER	Mitglieder der Gruppe besitzen die erforderlichen Zugriffsrechte und Privilegien, um als Anmeldekonto für die zugeordnete Instanz von SQ
SQLServerSQLAgentUser$NB-DELL-IV$MSSQLSERVER	Mitglieder der Gruppe besitzen die erforderlichen Zugriffsrechte und Privilegien, um als Anmeldekonto für die zugeordnete Instanz des SQ

Verwendung von Dienstkonten

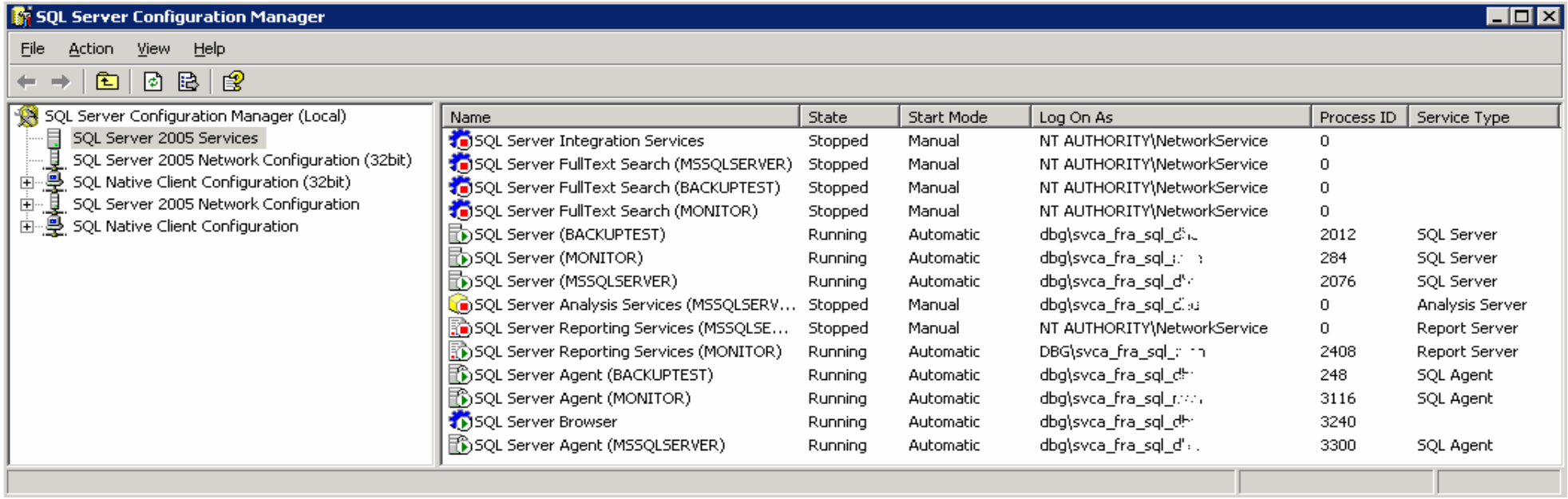
- Dienste über den SQL Server Configuration Manager konfigurieren
- Dienstkonten werden automatisch den Computergruppen hinzugefügt

DEMO:

Erstellung eines Serviceaccounts

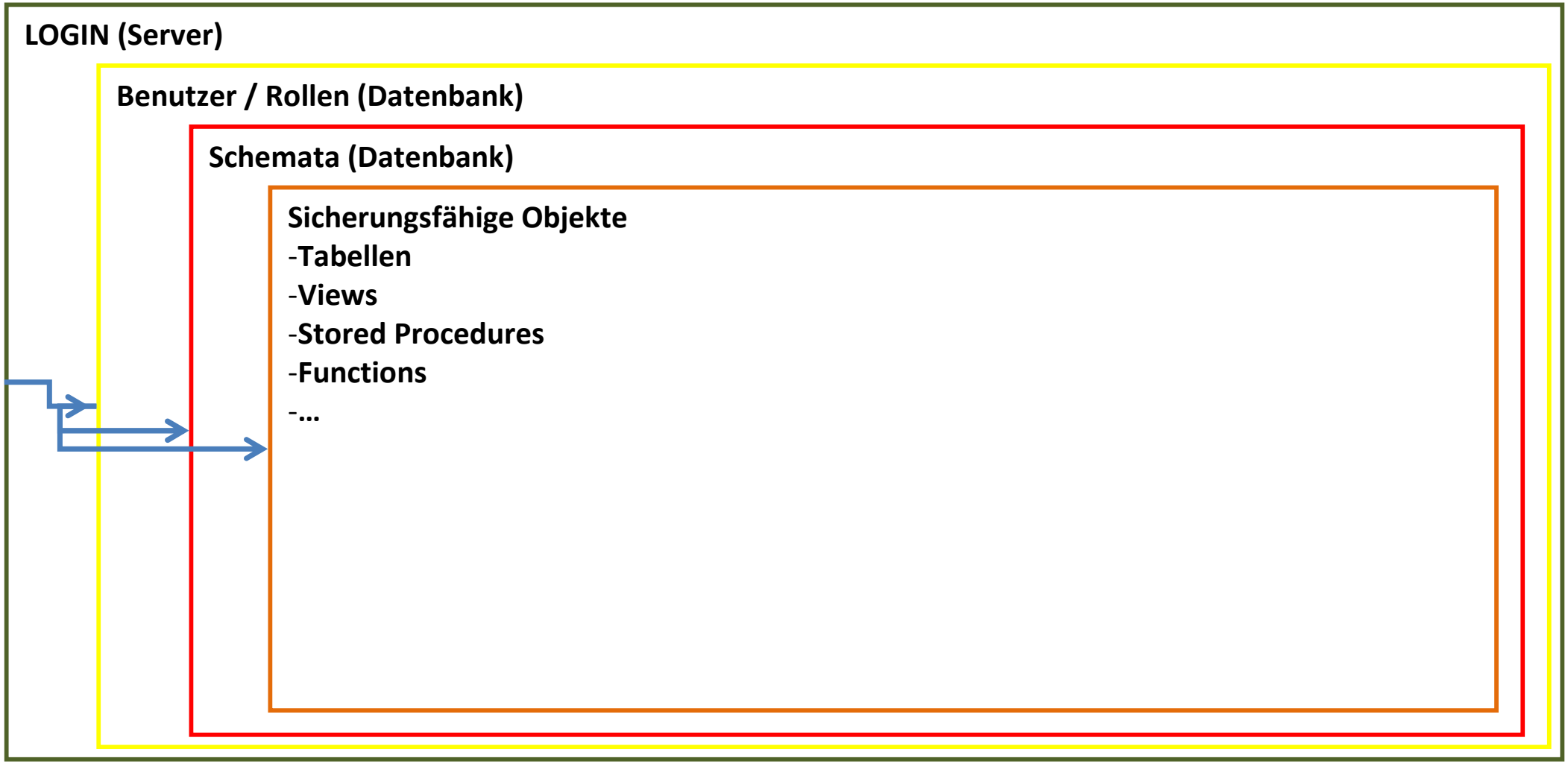
Konfiguration des SQL Server Engine Dienstes

Konfiguration der Berechtigung in Datenbankverzeichnis



Sicherheitshierarchie

Die Sicherheitseinschränkungen für einen zugriffsberechtigten Benutzer können auf verschiedene Sicherheitselemente (sicherungsfähige Elemente) angewendet werden.



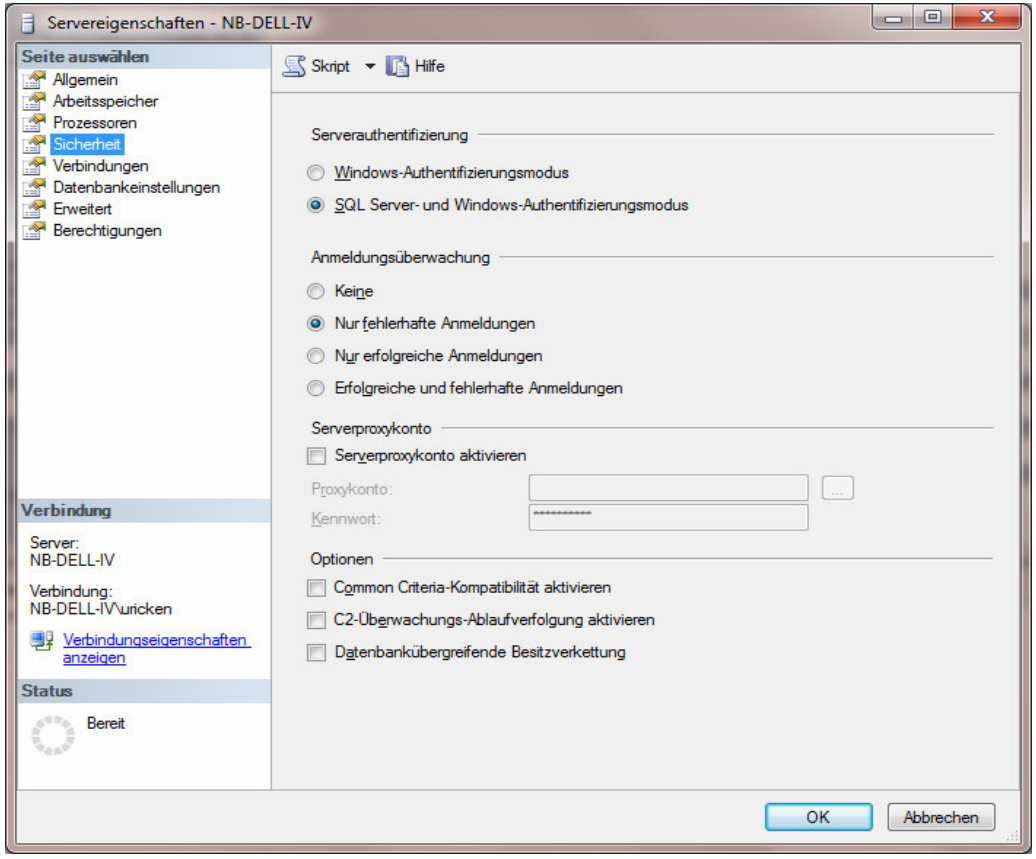
Zugang zu SQL Server (Authentication)

Berechtigung zum Zugriff auf den SQL Server über [Logins]

- Integrierte Sicherheit (Windows Konto)
- SQL Sicherheit (Von SQL Server verwaltetes Konto)

Konfiguration ist [Eigenschaft] des Servers

- Integrierter Modus (Windows only)
- Gemischter Modus (Windows und SQL)

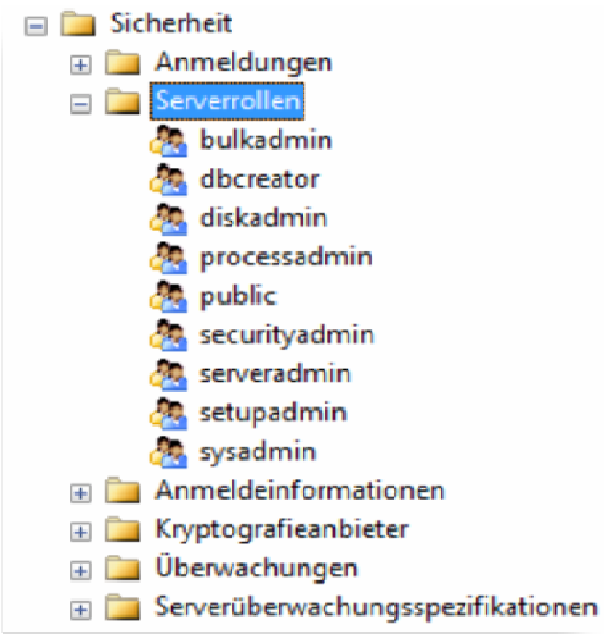


Berechtigungen für Server

Interne Serverrollen sind für die Verwaltungsebene des SQL Server vorgesehen

Es können keine selbstdefinierten Rollen hinzugefügt werden

Serverrollen sind unabhängig von Berechtigungen in den Datenbanken



Rolle	Beschreibung
Bulkadmin	Können BULK INSERT Befehle ausführen
Dbcreator	Können Datenbanken erstellen, verwalten und löschen
Diskadmin	Können die Datenbankdateien verwalten (vergrößern / verkleinern / hinzufügen / löschen)
Processadmin	Können Prozesse (SPID), die auf dem Server aktiv sind, beenden.
Securityadmin	Können Logins und Benutzer (Server / Datenbanken) verwalten.
Serveradmin	Können Serverkonfigurationen verändern und die Dienste stoppen.
Sysadmin	Können ALLE Aktivitäten auf dem SQL Server durchführen.
Setupadmin	Können [Linked Server] hinzufügen / löschen

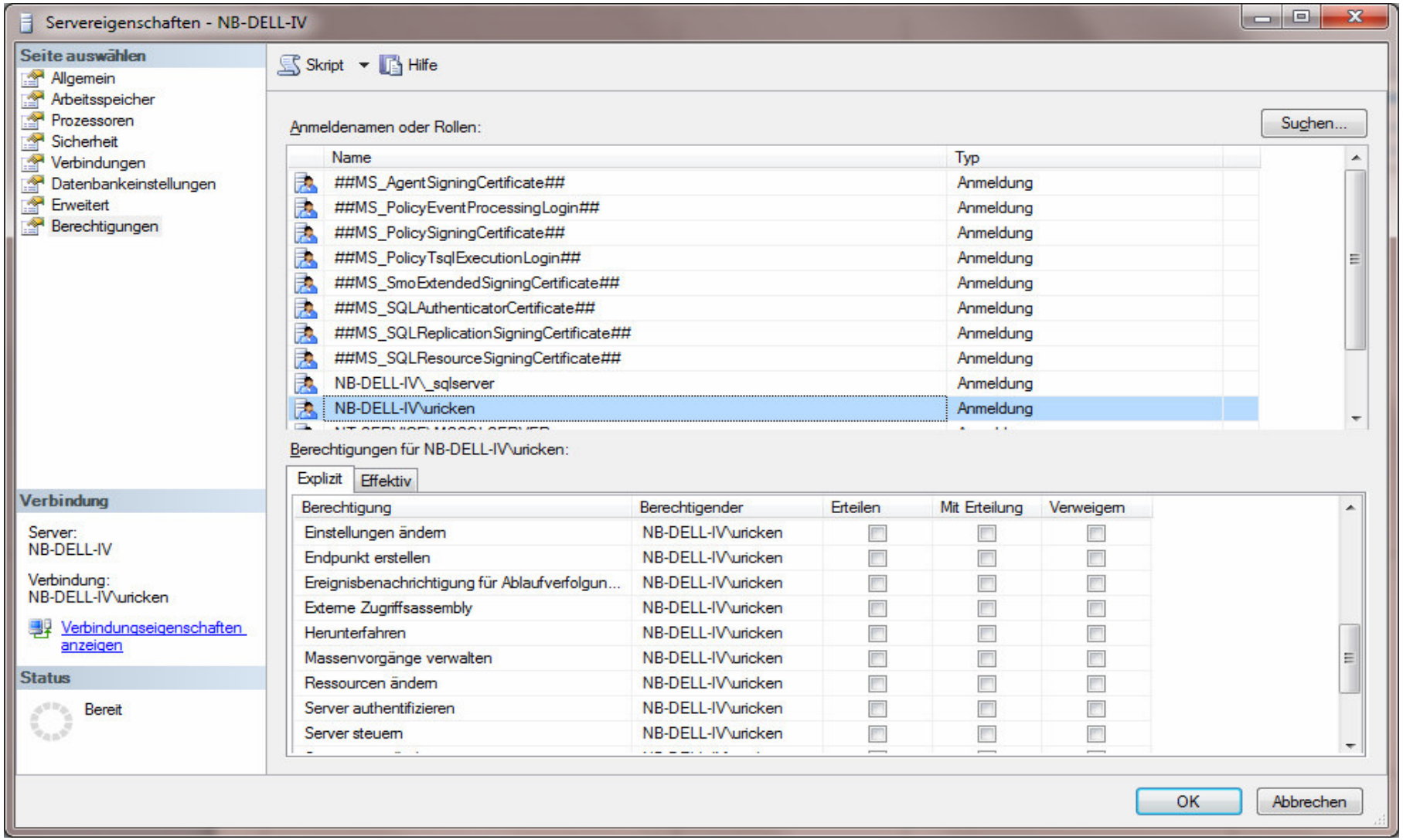
Berechtigungen für Server

Systembefehle /-objekte für die Verwaltung von Serverrollen

Prozedur	Beschreibung
sp_helpsrvrole	Liefert eine Liste aller verfügbaren Serverrollen
sp_helpsrvrolemember	Liefert eine Liste aller Logins, die Mitglied einer Serverrolle sind
sp_srvrolepermission	Zeigt Berechtigungen einer Serverrolle
IS_SRVROLEMEMBER()	Ermittelt, ob ein Login Mitglied einer bestimmten Serverrolle ist
sys.server_role_members	Liste aller Logins und deren Mitgliedschaften in Serverrollen
sp_addsrvrolemember	Prozedur zum Hinzufügen eines Logins zu einer Serverrolle
sp_dropsrvrolemember	Prozedur zum Entfernen eines Logins von einer Serverrolle

Individuelle Berechtigungen

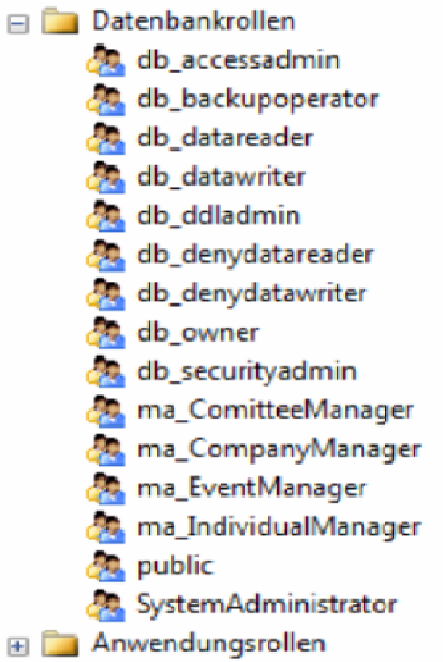
Explizite Berechtigungsstufen (unabhängig von Serverrollen)



Berechtigungen für Datenbanken

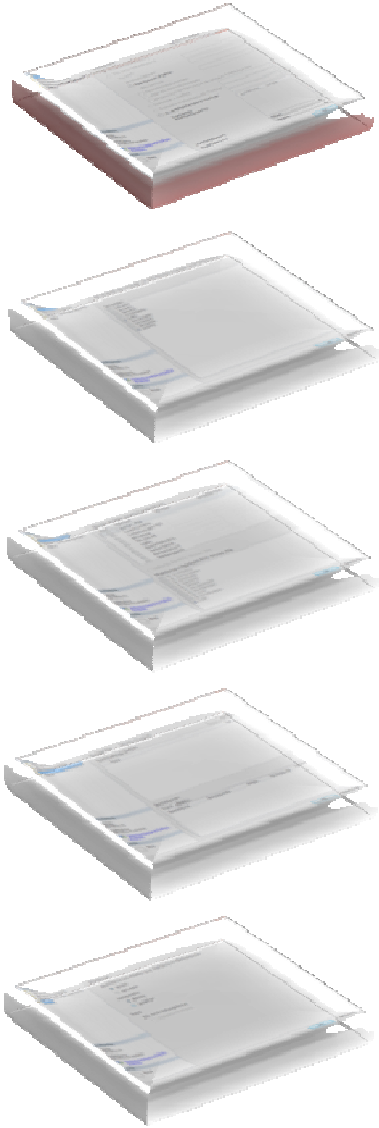
Zugriff auf Datenbanken erfolgt nur für authentifizierte Anmeldungen

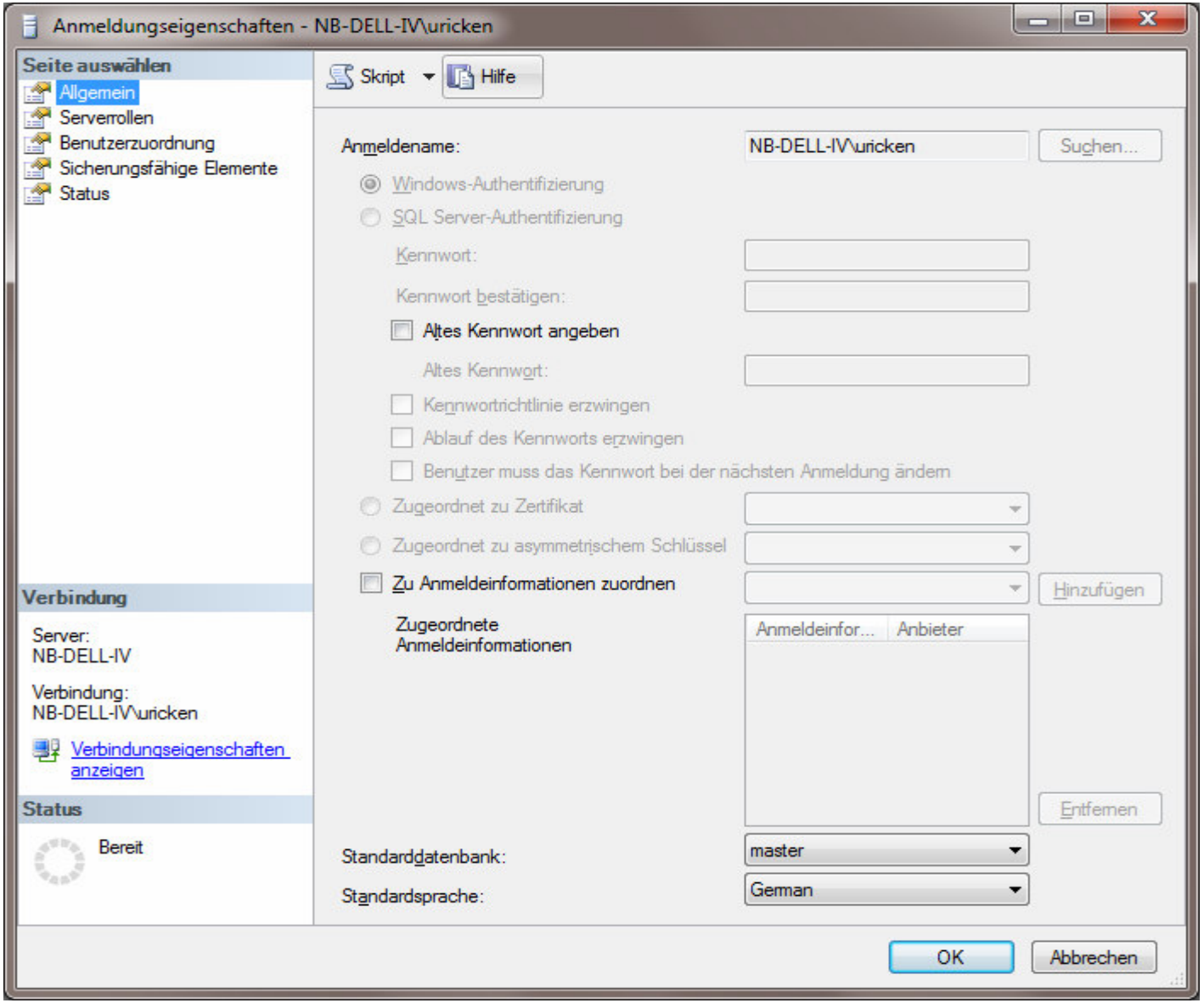
Berechtigungen in Datenbanken werden auf Benutzer übertragen



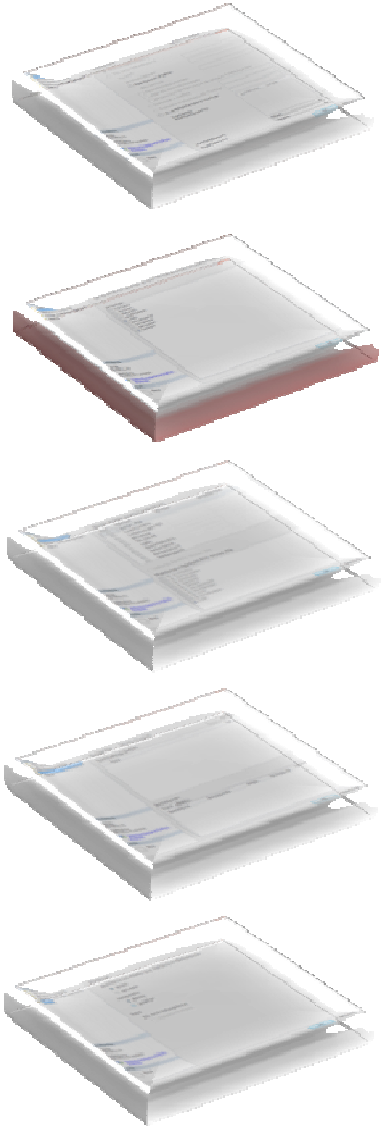
Rolle	Beschreibung
db_accessadmin	können den Zugriff für Windows-Anmeldungen, Windows-Gruppen und SQL Server-Anmeldungen hinzufügen und entfernen.
db_backupoperator	können eine Sicherung der Datenbank durchführen.
db_datareader	können für **jede** Tabelle bzw. Sicht in der Datenbank eine **SELECT-Anweisung** ausführen.
db_datawriter	können Daten in **allen** Benutzertabellen **hinzufügen**, **löschen** oder **ändern**.
db_ddladmin	können in einer Datenbank sämtliche DDL-Befehle (Data Definition Language) ausführen.
db_denydatareader	können keine Daten in den Benutzertabellen in einer Datenbank lesen.
db_denydatawriter	können keine Daten in den Benutzertabellen in einer Datenbank hinzufügen, ändern oder löschen.
db_owner	können **alle** Aktivitäten zur Konfiguration und Wartung an der Datenbank ausführen.
db_securityadmin	können die Rollenmitgliedschaft ändern und Berechtigungen verwalten.

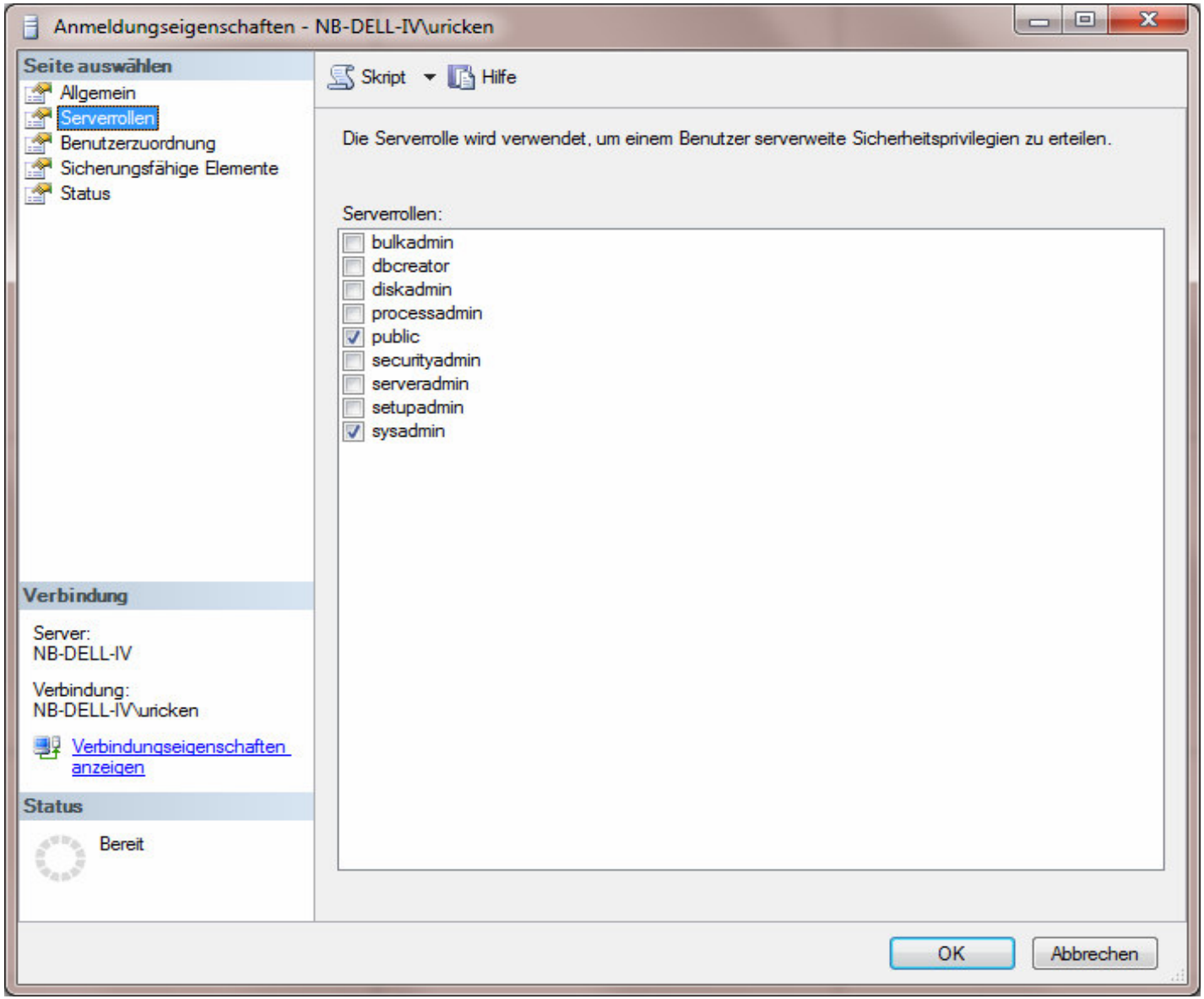
Eintrittskarte (Login)



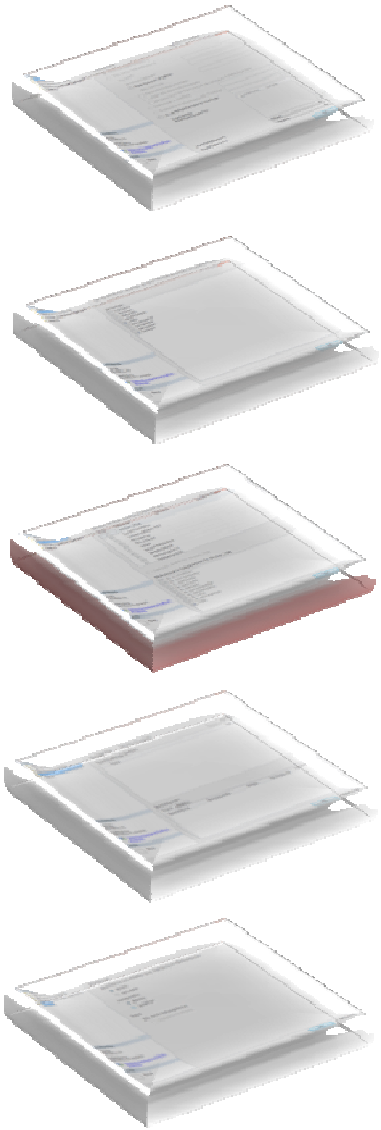


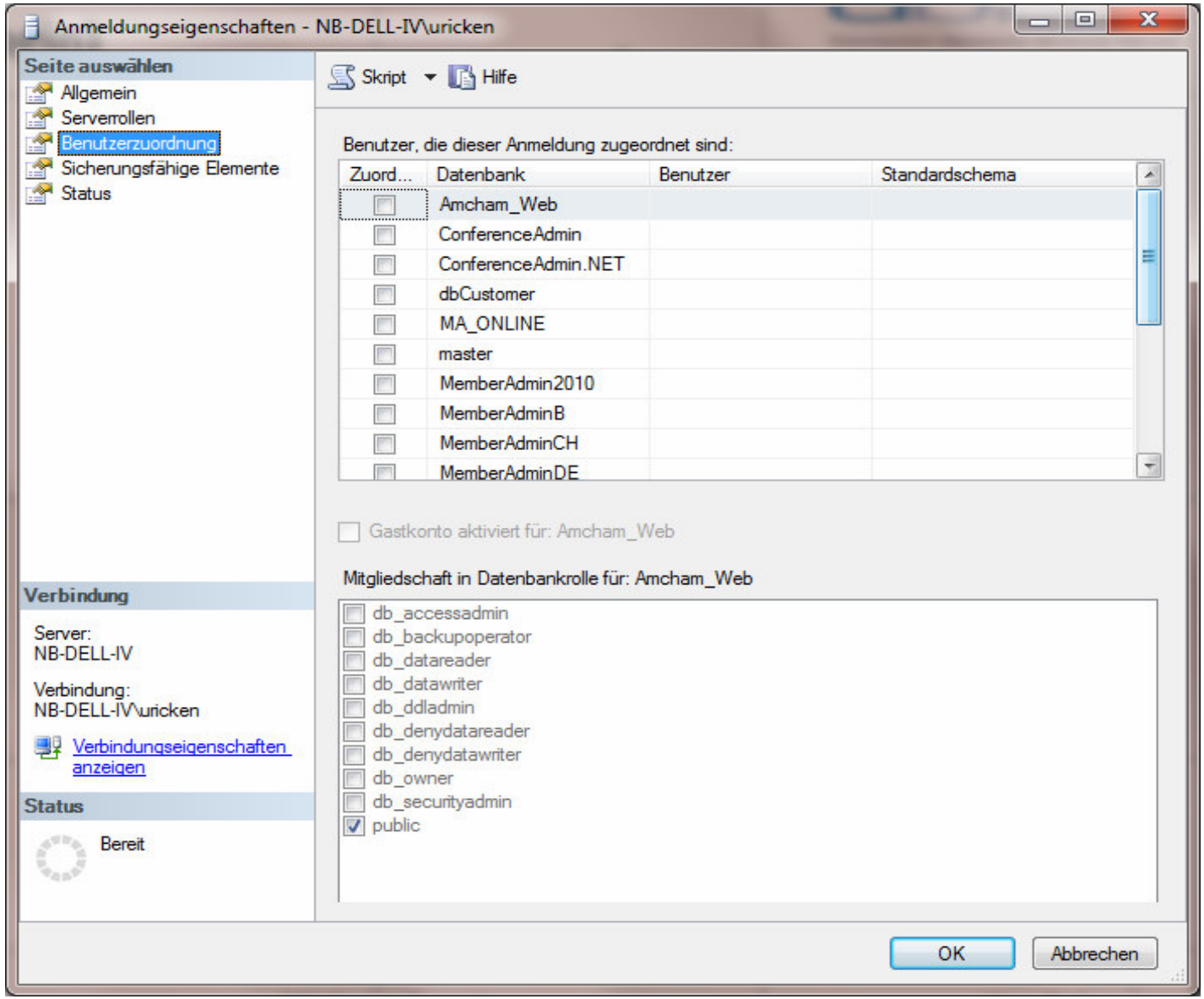
Eintrittskarte (Login)



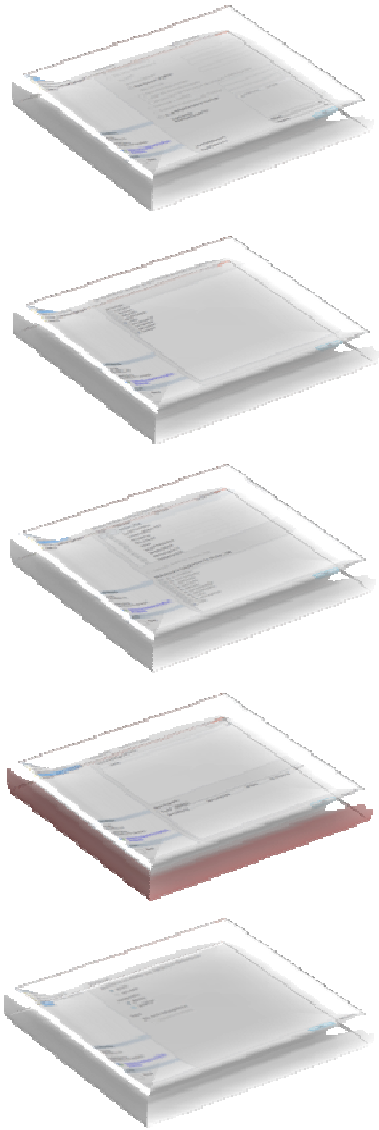


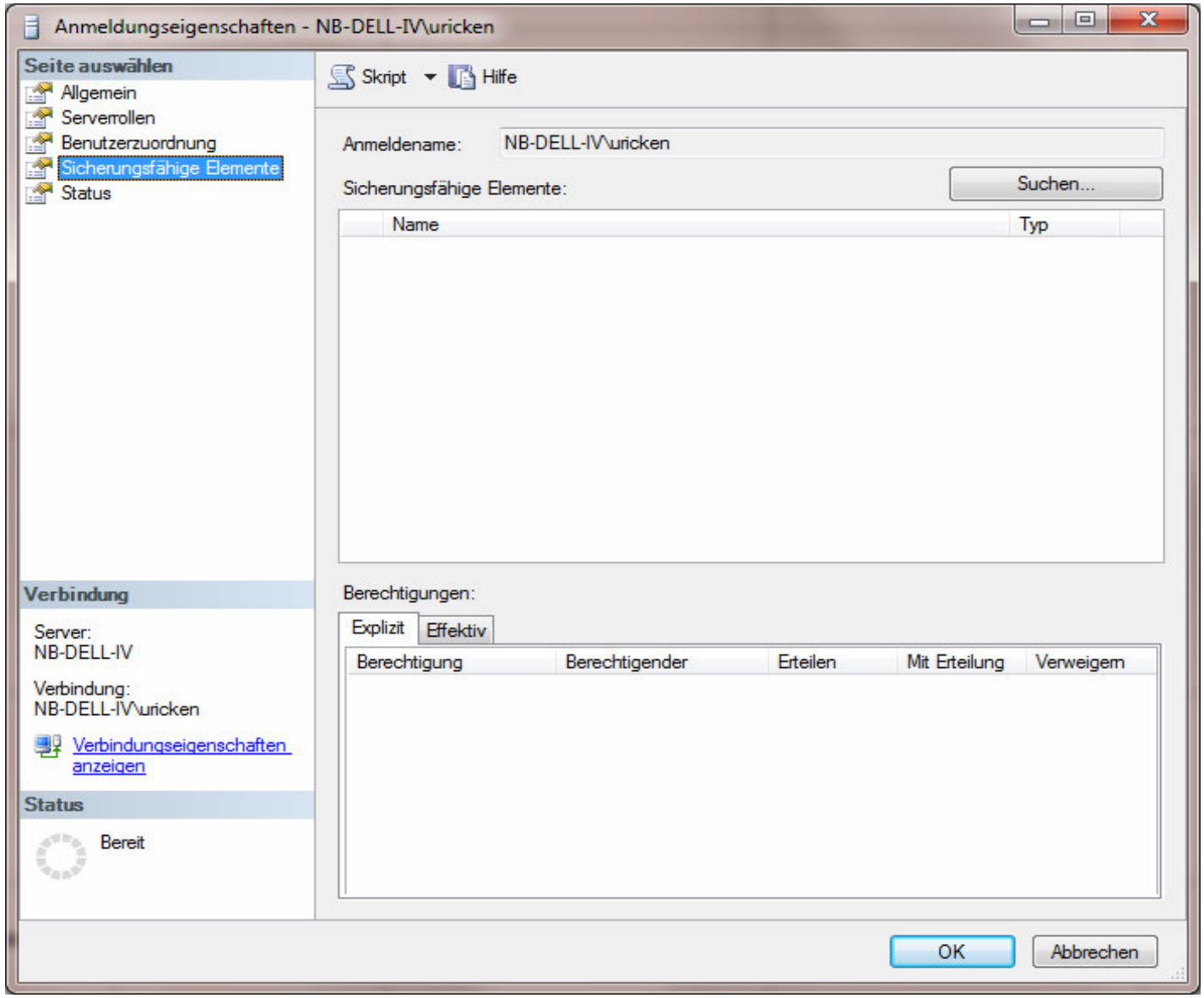
Eintrittskarte (Login)



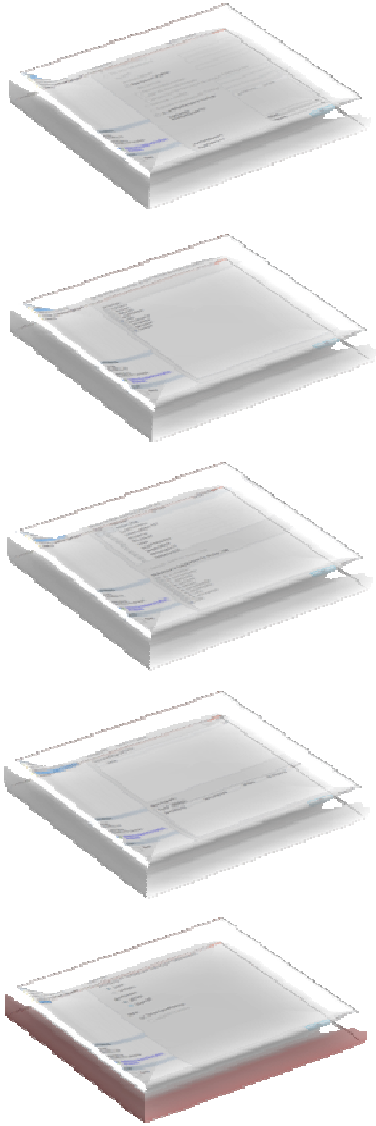


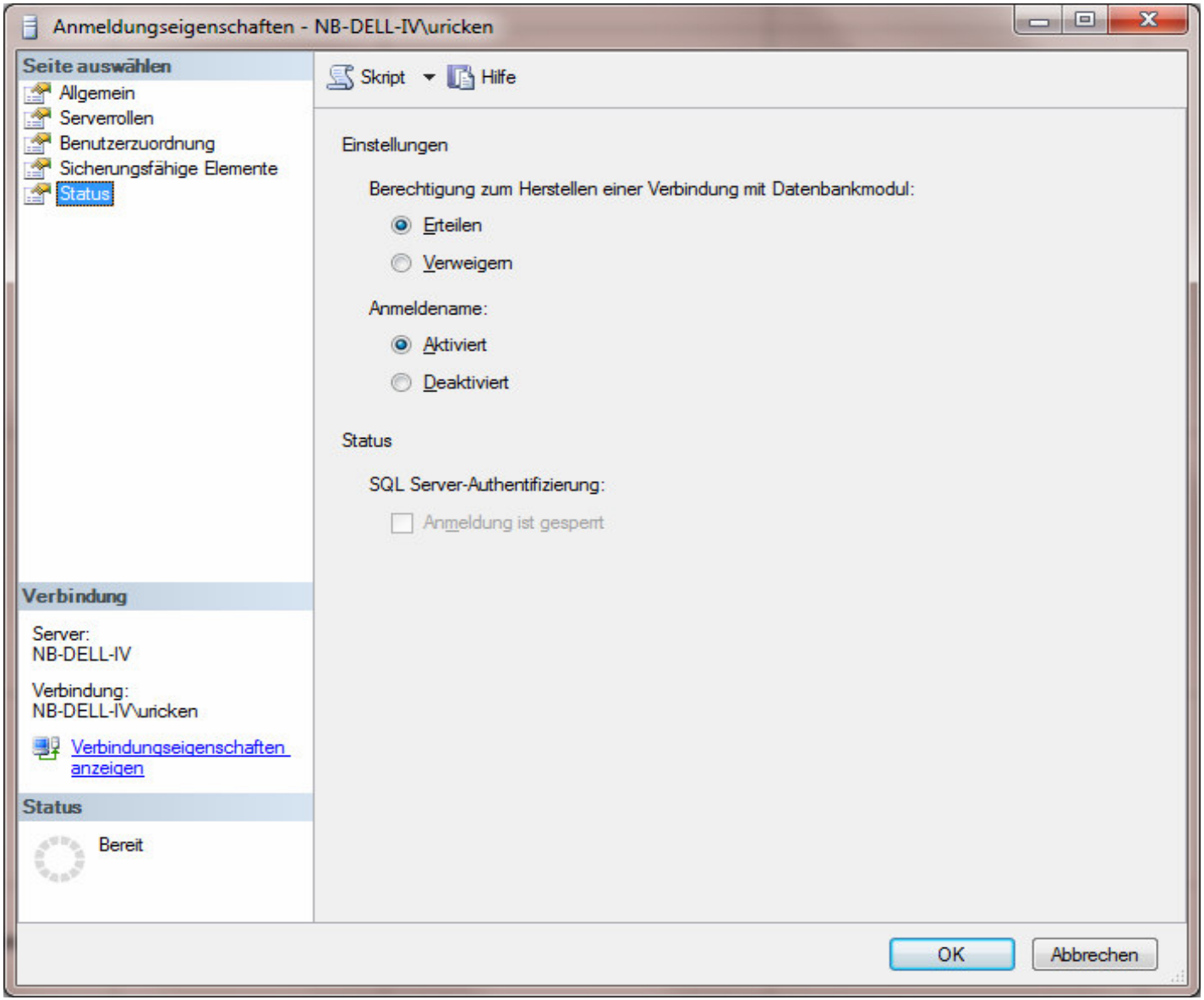
Eintrittskarte (Login)





Eintrittskarte (Login)





Eintrittskarte (Login)

Erstellen eines neuen Logins (SQL)

```
CREATE LOGIN <name> WITH PASSWORD = <'Password'>
MUST_CHANGE
CHECK_POLICY = ON / OFF                    -- Domänenrichtlinie
CHECK_EXPIRATION = ON / OFF                -- Domänenrichtlinie
```

Erstellen eines neuen Logins (Domänenkonto)

```
CREATE LOGIN <domain\konto> FROM WINDOWS
```

Beispiel:

```
CREATE LOGIN u1 WITH PASSWORD = 'sek' MUST_CHANGE
CREATE LOGIN [db-berater\uricken] FROM WINDOWS
```

Eintrittskarte (Login)

Zuordnung zu Serverrollen

EXEC sp_addsrvrolemember @loginame = '<Login>', @rolename = '<ServerRole>'

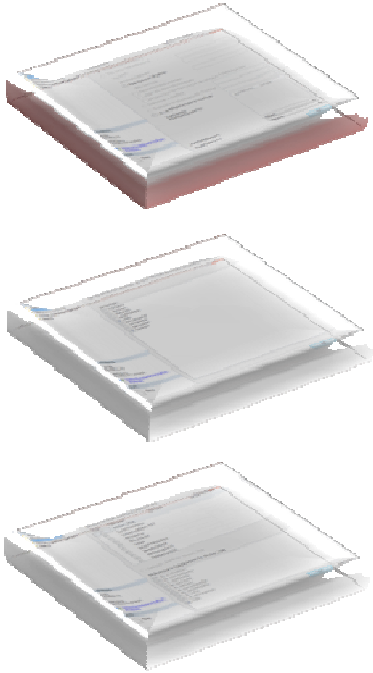
Beispiel:

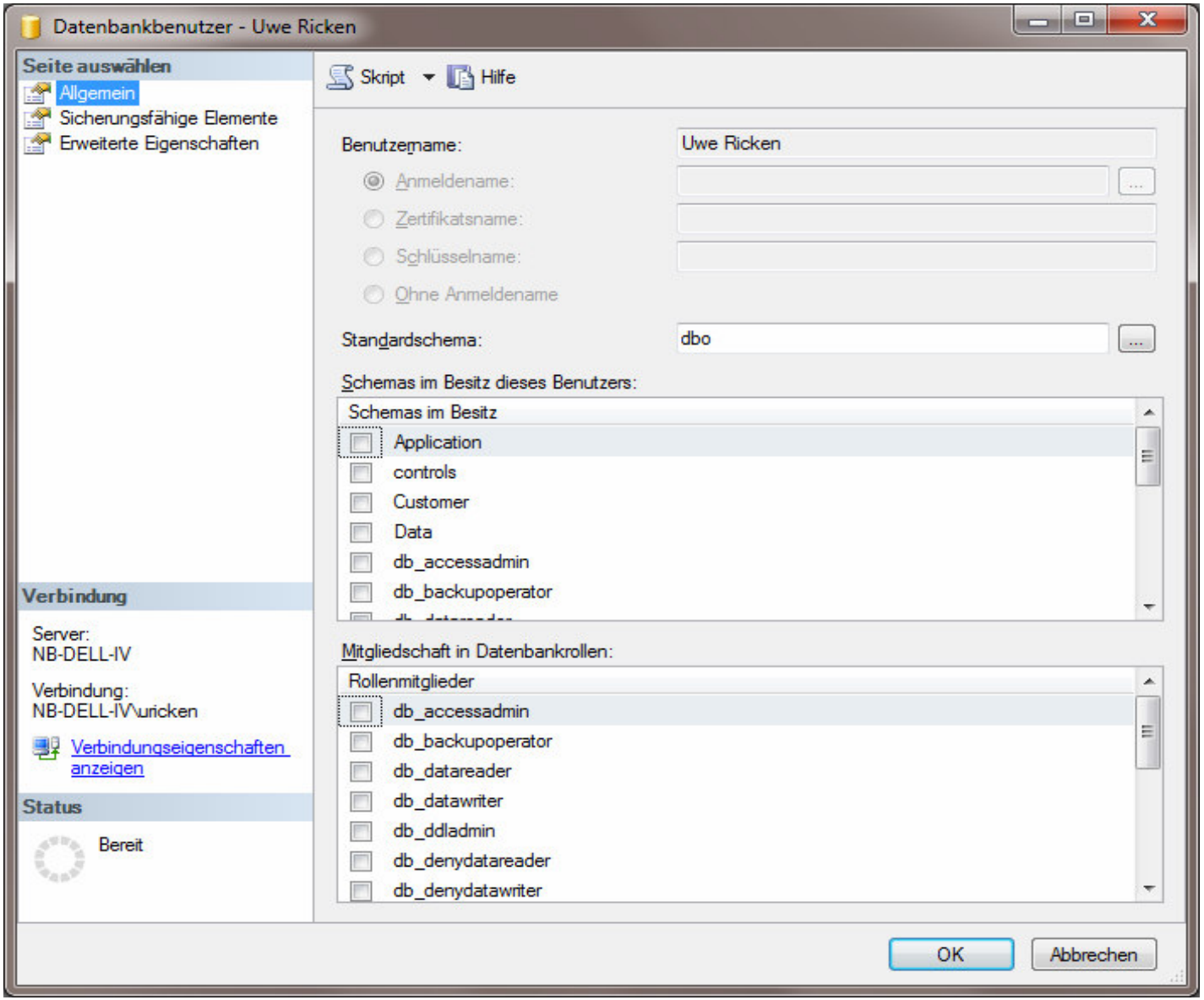
```
EXEC sp_addsrvrolemember @loginame = 'u1', @rolename = 'sysadmin'
```

Berechtigungen

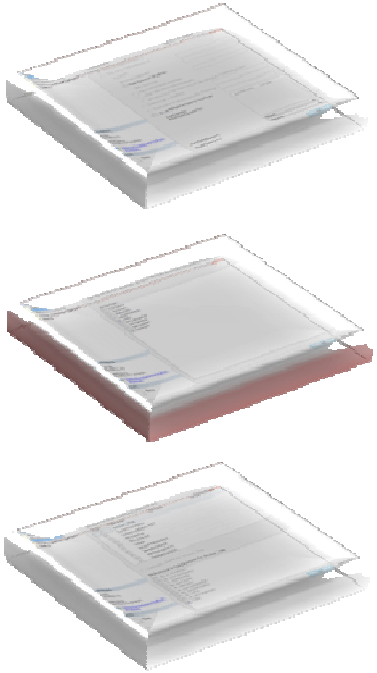
- Erfordert die ALTER ANY LOGIN-Berechtigung oder die ALTER LOGIN-Berechtigung für den Server.

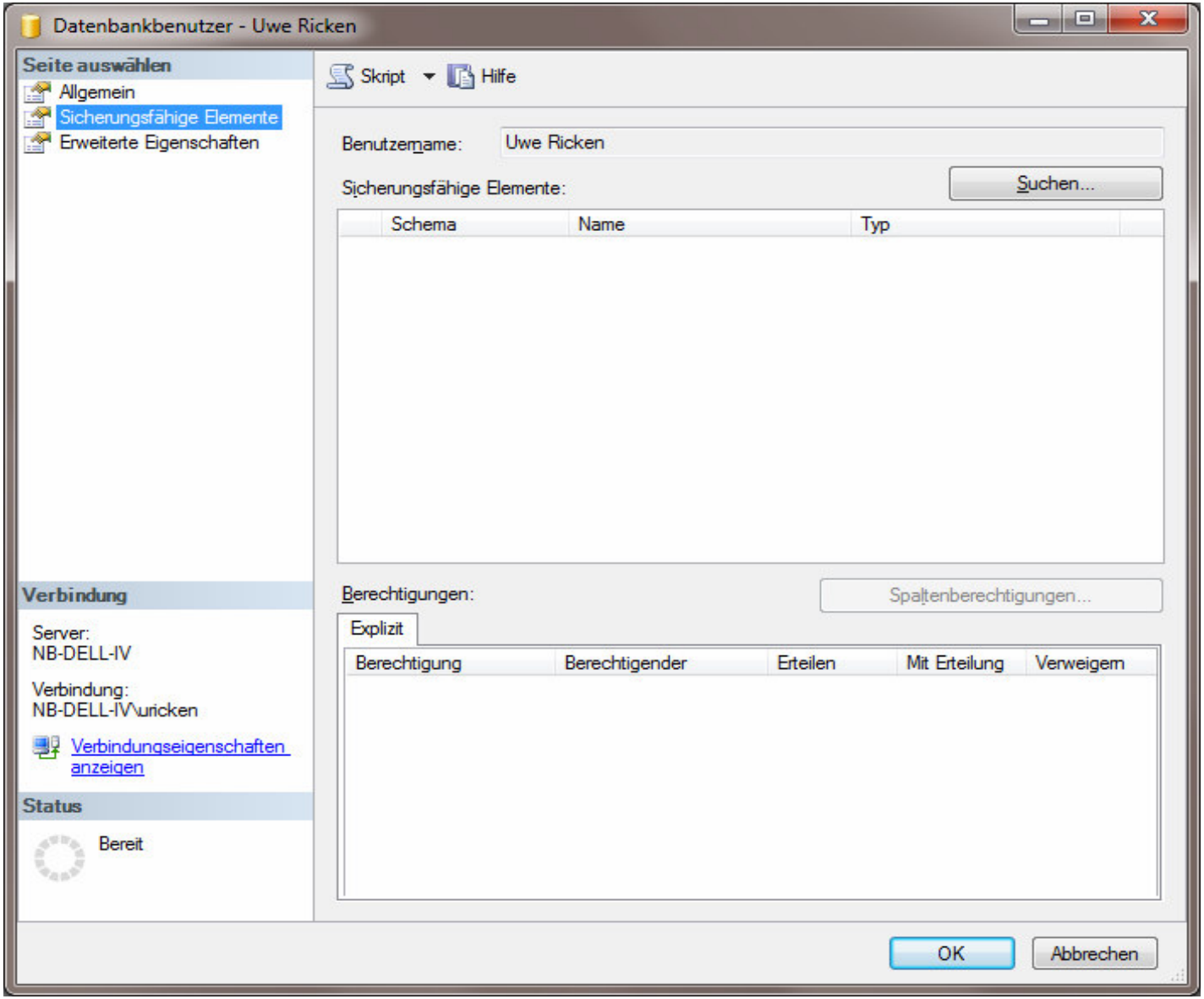
Eintrittskarte (User)



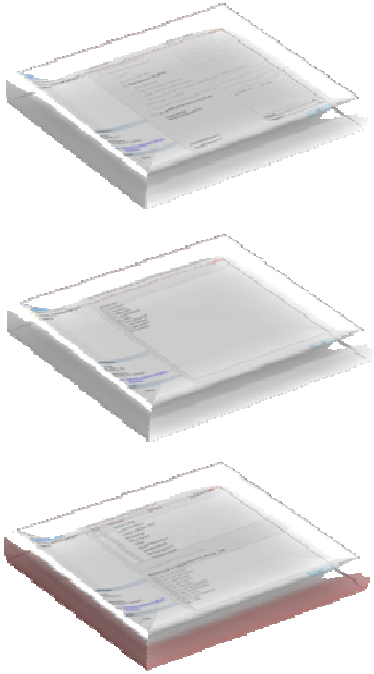


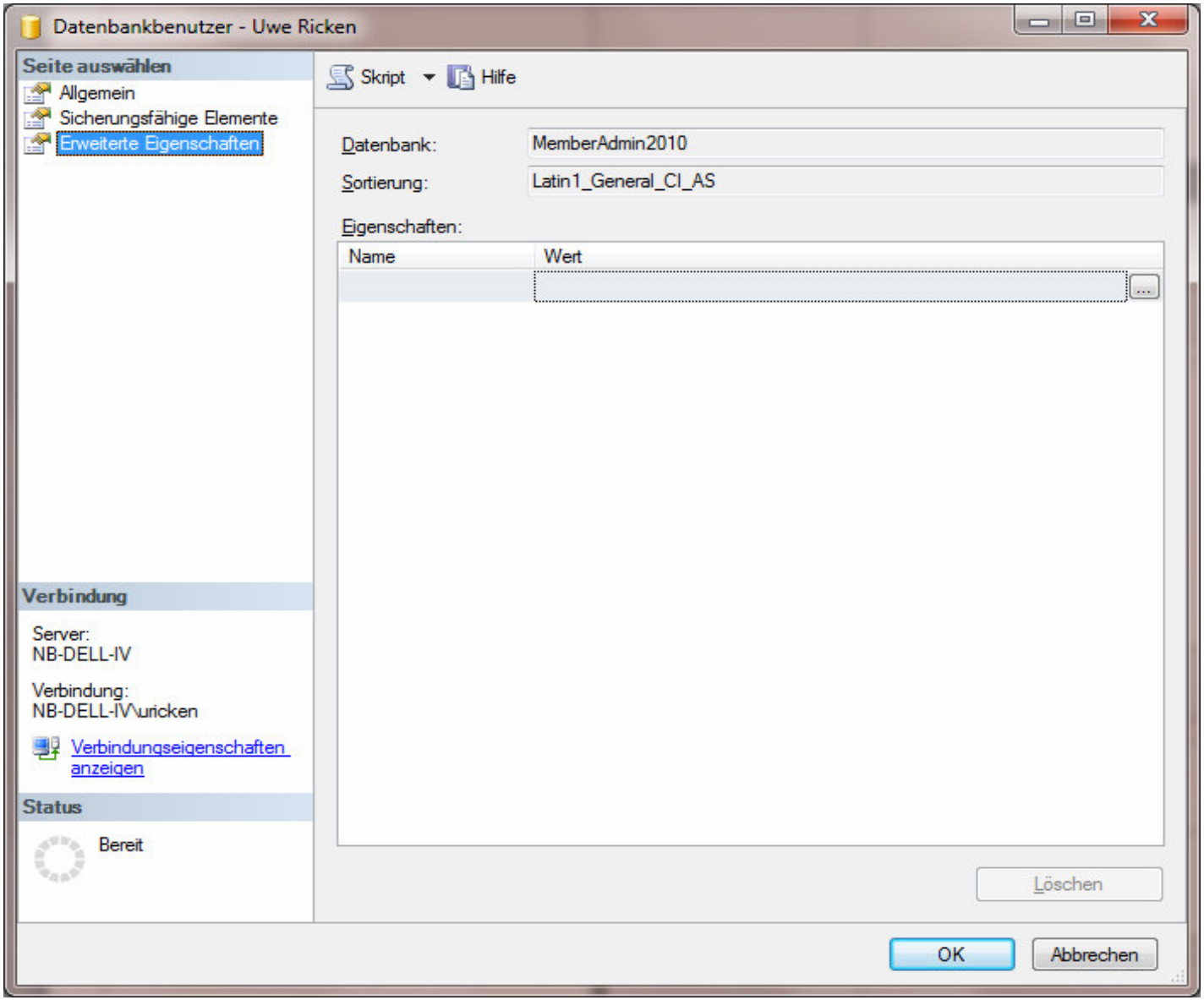
Eintrittskarte (User)





Eintrittskarte (User)





Eintrittskarte (User)

Hinzufügen eines Benutzers zu einer Datenbank

```
CREATE USER <username> FOR LOGIN <loginname> [WITH DEFAULT_SCHEMA  = <schema>
```

Beispiel

```
CREATE USER u1 FOR LOGIN u1 WITH DEFAULT_SCHEMA = u1
```

Hinweis:

- Wird FOR LOGIN ausgelassen, wird der Datenbankbenutzer dem SQL Server-Anmeldenamen mit demselben Namen zugeordnet.
- Wird DEFAULT_SCHEMA nicht definiert, verwendet der Datenbankbenutzer dbo als Standardschema.
- DEFAULT_SCHEMA kann auf ein Schema festgelegt werden, das derzeit nicht in der Datenbank vorhanden ist.

Berechtigungen

- Erfordert die ALTER ANY USER-Berechtigung in der Datenbank.

Besonderheiten bei SQL Benutzern

WICHTIG!

Die Verbindung zwischen einem Login und einem Datenbankbenutzer besteht durch den eindeutigen Schlüssel [Sid] in der Login- und der Benutzertabelle

```
SELECT sid, name, isntname, isntgroup, isntuser FROM sys.syslogins
```

	sid	name	isntname	isntgroup	isntuser
1	0x01	sa	0	0	0
2	0x010600000000000901000000041AB05E0B31EB78666287ED...	##MS_SQLResourceSigningCertificate##	0	0	0
3	0x010600000000000901000000AC5DA6E535933FFD392B6A5...	##MS_SQLReplicationSigningCertificate##	0	0	0
4	0x010600000000000901000000B2CD74E94B9C1B6756EA8B...	##MS_SQLAuthenticatorCertificate##	0	0	0
5	0x010600000000000901000000 5DB061F0B3BF9B487E202EF...	##MS_PolicySigningCertificate##	0	0	0
6	0x010600000000000901000000064FC3D29EAB41849908D7F3...	##MS_SmoExtendedSigningCertificate##	0	0	0
7	0x4EAF544D095570419CF280E6C9D103A6	##MS_PolicyEventProcessingLogin##	0	0	0
8	0x014EA8886B841C4CA1F7ED32489BBF62	##MS_PolicyTsqlExecutionLogin##	0	0	0
9	0x010600000000000901000000046118475F61285F02F9210FA...	##MS_AgentSigningCertificate##	0	0	0
10	0x01010000000000512000000	NT-AUTORITÄT\SYSTEM	1	0	1
11	0x010600000000000550000000E20F4FE7B15874E48E190264...	NT SERVICE\MSSQLSERVER	1	1	0
12	0x010500000000000515000000 2178DF9312926BEA63E273D...	NB-DELL-IV\uricken	1	0	1
13	0x010600000000000550000000DCA88F14B79FD47A992A3D8...	NT SERVICE\SQLSERVERAGENT	1	1	0
14	0x010500000000000515000000 2178DF9312926BEA63E273D...	NB-DELL-IV_sqlserver	1	0	1
15	0xCF9012EF0915E64B8FE80F283F6EE15C	u1	0	0	0

Besonderheiten bei SQL Benutzern

WICHTIG!

Die Verbindung zwischen einem Login und einem Datenbankbenutzer besteht durch den eindeutigen Schlüssel [Sid] in der Login- und der Benutzertabelle

```
SELECT sid, name, type_desc FROM sys.database_principals
SELECT sid, name, issqluser FROM sys.sysusers
```

	sid	name	type_desc
5	NULL	sys	SQL_USER
6	0xCF9012EF0915E64B8FE80F283F6EE15C	u1	SQL_USER
7	0x0105000000000009040000000000000000000000...	db_owner	DATABASE_ROLE
8	0x0105000000000009040000000000000000000000...	db_accessadmin	DATABASE_ROLE
9	0x0105000000000009040000000000000000000000...	db_securityadmin	DATABASE_ROLE
10	0x0105000000000009040000000000000000000000...	db_ddladmin	DATABASE_ROLE
11	0x0105000000000009040000000000000000000000...	db_backupoperator	DATABASE_ROLE
12	0x0105000000000009040000000000000000000000...	db_datareader	DATABASE_ROLE
13	0x0105000000000009040000000000000000000000...	db_datawriter	

	sid	name	issqluser
5	NULL	sys	1
6	0xCF9012EF0915E64B8FE80F283F6EE15C	u1	1
7	0x0105000000000009040000000000000000000000...	db_owner	0
8	0x0105000000000009040000000000000000000000...	db_accessadmin	0
9	0x0105000000000009040000000000000000000000...	db_securityadmin	0
10	0x0105000000000009040000000000000000000000...	db_ddladmin	0
11	0x0105000000000009040000000000000000000000...	db_backupoperator	0
12	0x0105000000000009040000000000000000000000...	db_datareader	0
13	0x0105000000000009040000000000000000000000...	db_datawriter	0

Besonderheiten bei SQL Benutzern

Problem:

Die Sid eines SQL Benutzers ist ein „uniqueidentifier“ und wird auf jedem Server unterschiedlich sein!

Scripten eines Benutzers mit dedizierter [Sid] und auf jedem Server entsprechend ausführen?

```
IF  EXISTS (SELECT * FROM sys.server_principals WHERE name = N'u1')
        DROP LOGIN [u1]
        GO

CREATE LOGIN [u1] WITH PASSWORD='sek',
                DEFAULT_DATABASE=[master],
                DEFAULT_LANGUAGE=[Deutsch],
                CHECK_EXPIRATION=OFF,
                CHECK_POLICY=ON,
                SId = 0x4461734973744D65696E4163636F756E
GO
```

Besonderheiten bei SQL Benutzern

Problem:

Die Sid eines SQL Benutzers ist ein „uniqueidentifier“ und wird auf jedem Server unterschiedlich sein!

Korrigieren der Sid von Login und Benutzer

```
USE tempdb
GO

EXEC     sp_change_users_login @Action = 'Report'
EXEC     sp_change_users_login @Action = 'Auto_Fix', @UserNamePattern = 'u1'
EXEC     sp_change_users_login @Action = 'Report,
```

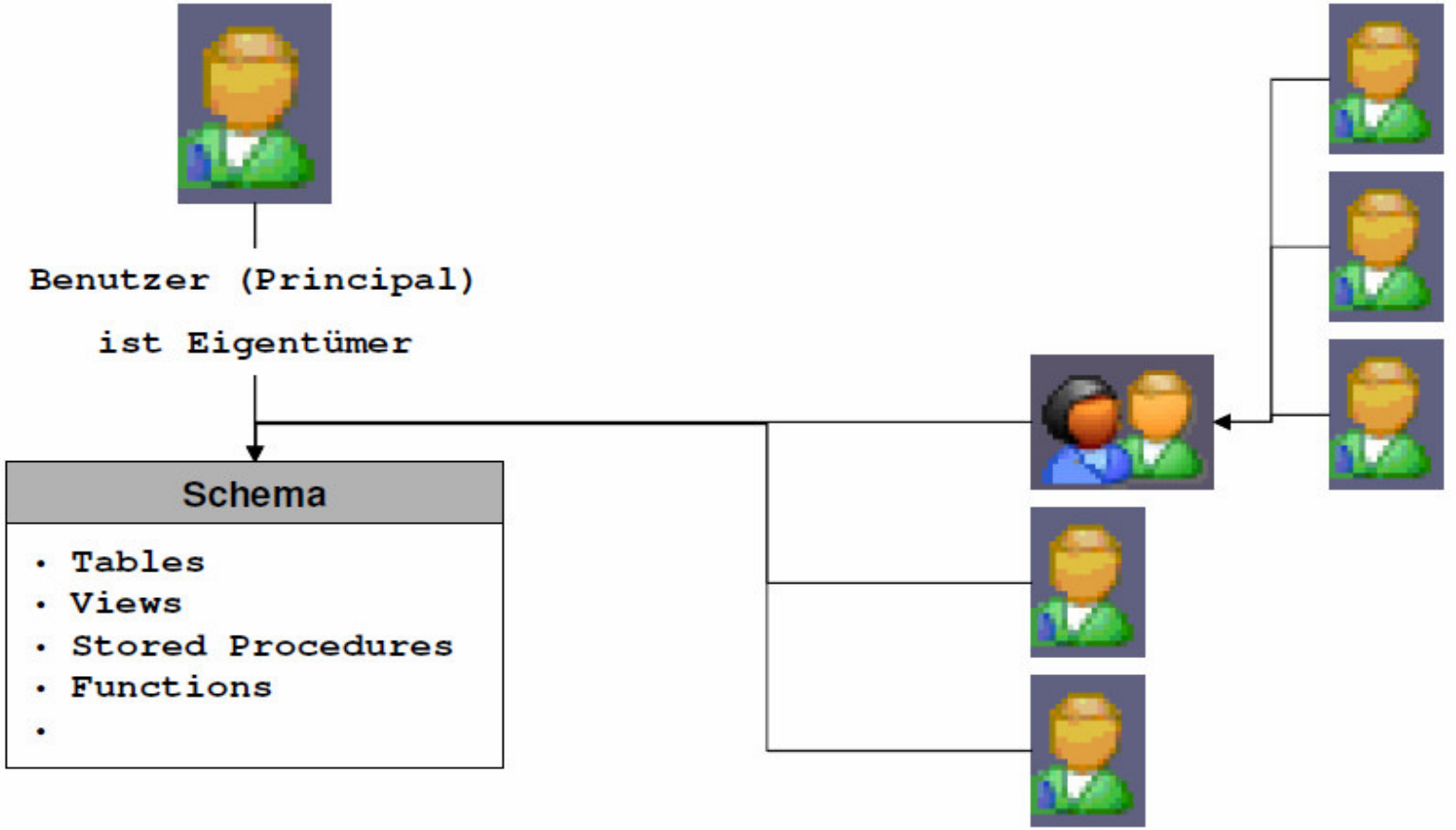
•Nur db_owner und/oder sysadmin können diese Prozedur ausführen

Schemata

Das Verhalten der Schemas wurde in SQL Server 2005 geändert. Schemas entsprechen nicht mehr Datenbankbenutzern. Jedes Schema ist jetzt ein eindeutiger Namespace, der unabhängig von dem Benutzer vorhanden ist, der ihn erstellt hat. Mit anderen Worten: ein Schema ist einfach ein Container von Objekten. Ein Schema kann im Besitz eines beliebigen Benutzers sein, und dieser Besitz ist übertragbar.

- Der Besitz von Schemas und sicherungsfähigen Elementen mit Schemabereich kann übertragen werden. Objekte können zwischen Schemas verschoben werden.
- In einem einzelnen Schema können Objekte enthalten sein, die im Besitz mehrerer Datenbankbenutzer sind.
- Mehrere Datenbankbenutzer können ein einzelnes Standardschema gemeinsam nutzen.
- Berechtigungen für Schemas und in Schemas enthaltene sicherungsfähige Elemente können mit größerer Genauigkeit verwaltet werden, als dies in früheren Versionen der Fall war.
- Jedes Schema kann im Besitz eines beliebigen Datenbankprinzipals sein. Dies schließt auch Rollen und Anwendungsrollen ein.
- Ein Datenbankbenutzer kann gelöscht werden, ohne dass Objekte im entsprechenden Schema gelöscht werden müssen.
- Mit Code, der für frühere Versionen von SQL Server geschrieben wurde, werden möglicherweise falsche Ergebnisse zurückgegeben, wenn im Code angenommen wird, dass Schemas und Datenbankbenutzer äquivalent sind.
- Wenn Sie beim Erstellen eines Datenbankobjekts einen gültigen Domänenprinzipal als Objektbesitzer (Benutzer oder Gruppe) angeben, wird der Domänenprinzipal der Datenbank als Schema hinzugefügt. **Das neue Schema befindet sich im Besitz des Domänenprinzipals.**

Schemata



Berechtigungen werden nur noch auf Schema
nicht mehr auf DB-Objekte vergeben!

Schemata

```
USE tempdb
GO
CREATE LOGIN u1 WITH PASSWORD = 'sek'
CREATE USER u1 WITH DEFAULT_SCHEMA = u1
GRANT CREATE TABLE to u1
CREATE SCHEMA sch1
CREATE SCHEMA u1 AUTHORIZATION u1

EXECUTE AS user = 'u1'
CREATE TABLE t1(c1 int)
Revert
GO

SELECT user_name(uid), * FROM sys.sysobjects WHERE name = 't1`
???

SELECT * FROM sys.sysusers
SELECT * FROM sys.schemas
```

Datenbankobjekte

Berechtigungen auf Datenbankobjekte hängen von dem Objekttypen ab:

Objekttype	Select	Insert	Update	Delete	Execute
Tabellen / Abfragen (U, V)	X	X	X	X	
Skalarfunktionen (FN)					X
Tabellenfunktionen (IF, TF)	X				
Gespeicherte Prozeduren (P)					X

Berechtigungsvergabe auf Ebene eines Schemas

- Einfache Verwaltung der Berechtigungen auf Ebene von Schemata
- Berechtigungen werden an Benutzer / Rollen vergeben
- Keine expliziten Berechtigungen auf Objekte mehr notwendig

Beispiel:

GRANT EXECUTE, SELECT ON SCHEMA :: <SchemaName>

Zugriff auf Dateisystem

- Generell ist der Zugang zum Dateisystems deaktiviert
- Zugang erfolgt ausschließlich mit Systemprozedur xp_cmdShell

Aktivierung erfolgt über Configuration Manager oder sp_configure (Globale Konfiguration des SQL-Servers)

- Ausführungsberechtigung (ohne Parameterkonfiguration für Gruppe <public>
- Ausführungsberechtigung mit Parameterkonfiguration für Mitglieder der Servergruppe <serveradmin> oder <alter settings> - Berechtigung
- Jede Änderung mittels sp_configure muss mit RECONFIGURE abgeschlossen werden.
- Änderungen werden i. d. R. sofort wirksam

Beispiel:

```
EXEC      sp_configure
EXEC      sp_configure 'show advanced option', 1
RECONFIGURE
```

Aktivierung von xp_cmdShell

```
EXEC   sp_configure 'show advances option', 1
RECONFIGURE
```

Ergebnisse | Meldungen

	name	minimum	maximum	config_value	run_value
66	transform noise words	0	1	0	0
67	two digit year cutoff	1753	9999	2049	2049
68	user connections	0	32767	0	0
69	user options	0	32767	0	0
70	xp_cmdshell	0	1	0	0

```
EXEC   sp_configure 'xp_cmdshell', 1
RECONFIGURE
```

	name	minimum	maximum	config_value	run_value
66	transform noise words	0	1	0	0
67	two digit year cutoff	1753	9999	2049	2049
68	user connections	0	32767	0	0
69	user options	0	32767	0	0
70	xp_cmdshell	0	1	1	1

Sicherheitsaspekte bei der Verwendung von xp_cmdshell

- Per Standard wird xp_cmdShell im Kontext des Dienstkontos des SQL Servers ausgeführt
- Standardbenutzer haben per Standard keine Ausführungsberechtigung für xp_cmdshell
- xp_cmdshell wird SYNCHRON ausgeführt. Die Kontrolle wird erst nach der vollständigen Ausführung von xp_cmdShell an den Batch zurückgegeben

Berechtigungsvergabe für xp_cmdshell

1. Login als Benutzer zur Datenbank <master> hinzufügen

```
USE master
GO

CREATE USER u1 FROM LOGIN u1
```

2. Berechtigungen auf die Prozedur xp_cmdShell erteilen

```
GRANT EXECUTE ON xp_cmdShell TO u1
```

Verwendung eines Proxyaccounts

- Verwendung von Proxyaccount limitiert die Berechtigungen
- Proxyaccount kann auch serverübergreifende Prozesse starten
- Proxyaccount kann Berechtigungslevel von Standardbenutzern erteilt werde

1. Erstellen eines Domänen-Accounts (Windows- /PC-Domäne)
2. Ausführen der Prozedur sp_xp_cmdshell_proxy_account

Beispiel

```
DECLARE @ReturnValue      int
EXEC    @ReturnValue = sp_xp_cmdshell_proxy_account 'Domain\user', 'Passwort,
SELECT  @ReturnValue
```

Deaktivierung eines Proxyaccounts

```
EXEC    @ReturnValue =  sp_xp_cmdshell_proxy_account NULL
```

Dynamisches SQL

- Hinreichend bekannt, dass dynamisches SQL ein hohes Sicherheitsrisiko beeinhaltet
  Siehe (www.sommerskog.com)

- Dynamisches SQL wird IMMER im Kontext des angemeldeten Benutzer ausgeführt

- Zwei Möglichkeiten der Verwendung von dynamischem SQL
  - EXEC('SQL-Statement')
  - EXEC sp_executeSQL

- EXEC speichert keine Abfragepläne
- EXEC gibt keine Werte zurück
- Verwendung von Variablen bei sp_executeSQL möglich
- Rückgabewerte sind bei sp_executeSQL möglich !!!

Dynamisches SQL (EXEC)

Beispiele:

```
-- Standardverfahren
DECLARE  @stmt              nvarchar(255)
DECLARE  @username          nvarchar(50)
DECLARE  @Attribute         nvarchar(50)
SET      @stmt    =         'SELECT * FROM sys.database_principals'
EXEC     (@stmt)

-- Verwendung von Variablen
SET @username     = 'guest'
SET @stmt         = 'SELECT * FROM sys.database_principals WHERE name = ''' +
   @username + ''''
EXEC (@stmt)

-- Variablen können für jeden Teil des Statements verwendet werden
SET @Attribute    = '[name]'
SET @stmt         = 'SELECT ' + @Attribute + ' FROM sys.database_principals'
EXEC (@stmt)
```

sp_executeSQL

```
-- Standardverfahren
DECLARE  @stmt              nvarchar(255)
DECLARE  @parms             nvarchar (50)
DECLARE  @username          nvarchar (50)

SET @stmt = 'SELECT * FROM sys.database_principals'
EXEC sp_executeSQL @stmt

SET @stmt = 'SELECT * FROM sys.database_principals WHERE name =
   @principal_name'
SET @parms = '@principal_name nvarchar(64)'
SET @username =  'guest'
EXEC sp_executeSQL @stmt, @parms, @principal_name = @username

SET @stmt = 'SELECT @ReturnValue = COUNT(SId) FROM sys.database_principals'
SET@parms = '@ReturnValue int OUTPUT'
DECLARE @ret int
EXEC sp_executeSQL @stmt, @parms, @ReturnValue = @ret OUTPUT
SELECT   @ret
```

Identitätswechsel

Ein Identitätswechsel kann durch eine Stored Procedure oder T-SQL ausgelöst werden

- Ein Datenbankbenutzer oder Serveranmeldename, der in der EXECUTE AS-Klausel eines Moduls angegeben ist, kann erst gelöscht werden, wenn das Modul zur Ausführung unter einem anderen Kontext geändert wird.
- Der in der EXECUTE AS-Klausel angegebene Benutzer oder Anmeldename muss als Prinzipal in **sys.database_principals** bzw. **sys.server_principals** vorhanden sein. Darüber hinaus benötigt der Benutzer, der das Modul erstellt oder ändert, **IMPERSONATE-Berechtigungen für den Prinzipal**.
- Falls der Benutzer impliziten Zugriff auf die Datenbank oder Instanz von SQL Server über eine Windows-Gruppenmitgliedschaft hat, wird der in der EXECUTE AS-Klausel angegebene Benutzer beim Erstellen des Moduls implizit erstellt. Dazu muss eine der folgenden Anforderungen erfüllt sein:
  - Der angegebene Benutzer oder Anmeldename ist ein Mitglied der festen Serverrolle **sysadmin**.
  - Der Benutzer, der das Modul erstellt, hat die Berechtigung zum Erstellen von Prinzipalen.

Identitätswechsel

Syntax:

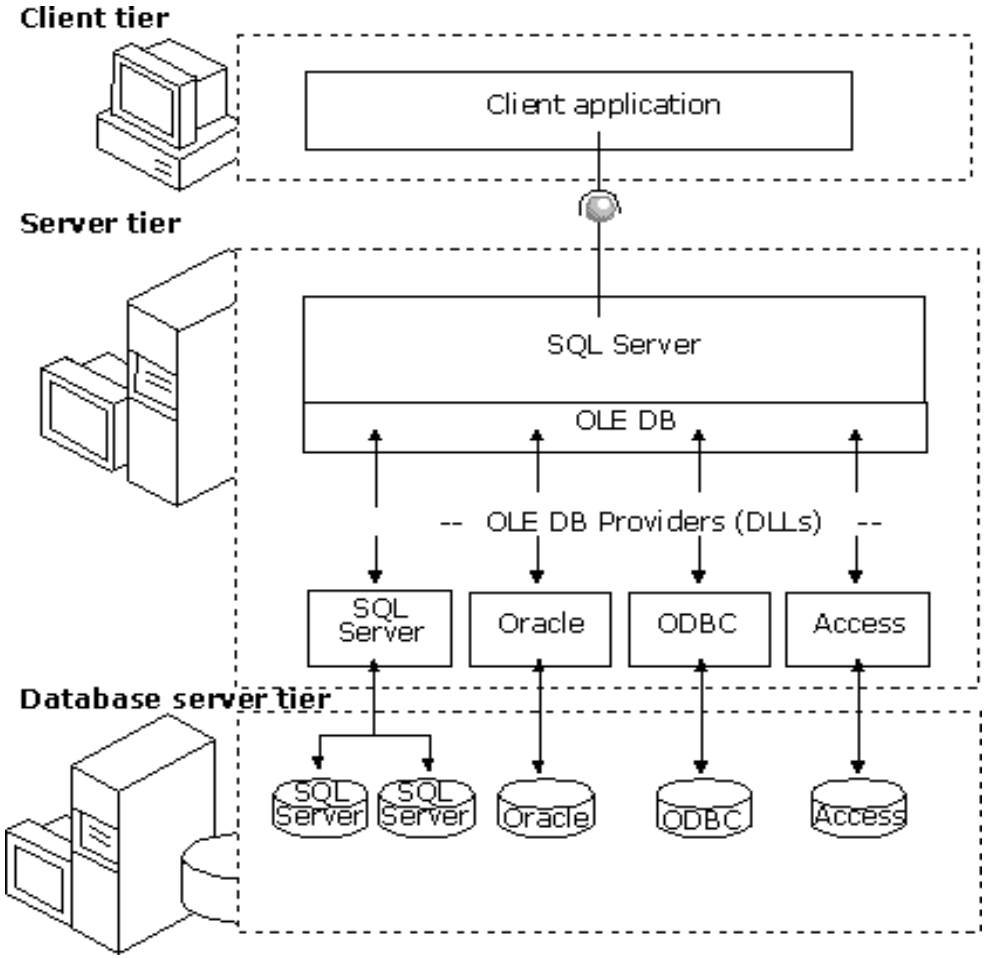
EXEC AS user = '<Datenbankbenutzer>'

Beispiel:

```
-- u1 als Benutzer hinzufügen
CREATE USER u1 FROM LOGIN u1
GO

DROP TABLE dbo.myTable
GO

CREATE TABLE dbo.myTable
(id int, mytext nvarchar(50))
GO

-- Security
REVOKE ALL ON dbo.myTable TO public
GO
```

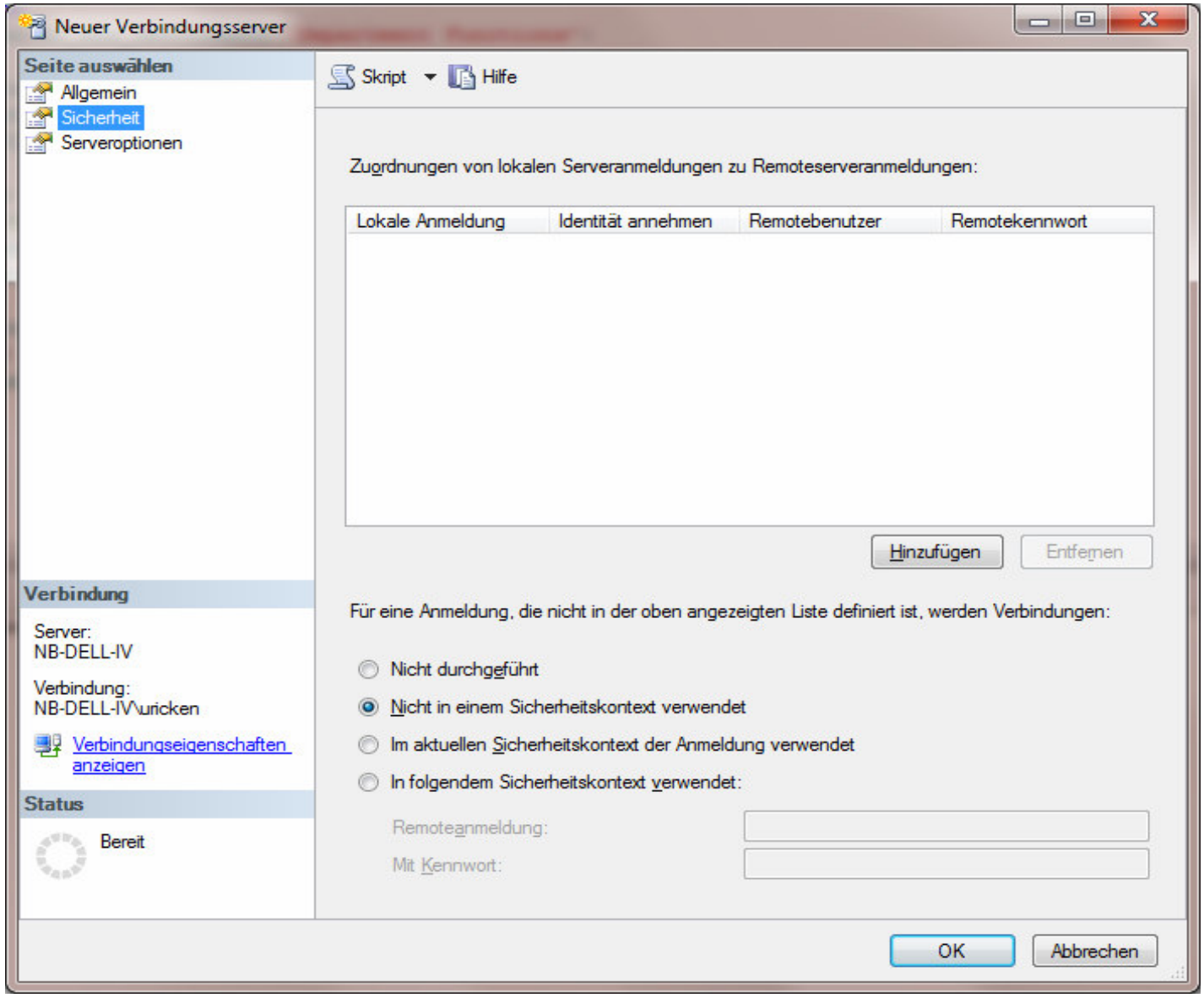
```
-- Klappt's
SELECT * FROM dbo.myTable

EXECUTE AS user = 'u1'
SELECT * FROM dbo.myTable
REVERT
```

Linked Server



- Ein „Linked Server“ erlaubt es Microsoft SQL Server, Befehle gegen eine OLEDB Datenquelle auf verschiedenen Servern auszuführen
  - SQL Server
  - Jet
  - ODBC
  - ORACLE
  - Indexing Services
- Erstellung eines Linked Servers entwender über SSMS oder Stored Procedures
  - sp_addLinkedServer
  - sp_dropserver

Linked Server

- Der Sicherheitskontext für den Zugriff auf einen [Linked Server] hängt vom Typen des Linked Servers ab.
- Nur SQL Server kann „im aktuellen Sicherheitskontext“ verwenden.
- Aliase können für lokale Anmeldungen verwendet werden, um ein abgestuftes Sicherheitsmodell zu implementieren.

**** DEMO ****