

Microsoft SQL-Server 2005

Christian Zahler

1 Inhaltsverzeichnis

- 1 Inhaltsverzeichnis
- 2 SQL Server: Übersicht, Versionen, Editionen
 - 2.1 Versionen
 - 2.2 Editionen im Vergleich
- 3 Installation von SQL Server 2005
 - 3.1 Basisinstallation
 - 3.2 Nachträgliches Hinzufügen von Beispielen oder anderen SQL Server-Komponenten
 - 3.3 Installieren von Service Pack 2
 - 3.4 Architektur von SQL Server 2000/2005
 - 3.5 Administrations-Tools
 - 3.6 Systemdatenbanken
 - 3.7 Objektnamen, Schemas
 - 3.8 Dateimäßiger Aufbau einer SQL Server 2005-Datenbank
- 4 Sicherheit und Zugriff auf SQL Server 2005
 - 4.1 Unterschied Login - User
 - 4.2 Authentifizierungs-Methoden
 - 4.3 Anlegen von Login-Konten
 - 4.4 Zuordnung eines Logins zu einem DB-User
 - 4.5 Rollen
 - 4.6 Anwendungsrollen
- 5 Client-Zugriff auf MS SQL Server 2005
 - 5.1 Grundlagen
 - 5.2 MS Access 2007 als Client mit Hilfe einer ODBC-Systemschnittstelle
 - 5.3 MS Access-Datenbankprojekte (ohne ODBC-Schnittstelle)
- 6 Migration einer Access 2007-Datenbank auf SQL Server 2005
 - 6.1 Upgrade mit dem Access 2007-Upsizing-Assistenten
 - 6.2 Datenimport aus einer Access-Datenbank mit dem SQL Server Integration Services (SSIS)-Import/Export-Assistent
- 7 Backup und Restore
 - 7.1 Backup-Grundlagen
 - 7.2 Restore
- 8 Datenbank-Snapshots
- 9 Überwachung
 - 9.1 Ausführungspläne
 - 9.2 SQL Server Profiler
 - 9.3 Datenbankmodul-Optimierungsratgeber
 - 9.4 Erkennung von Deadlocks
 - 9.5 Aktivitätsmonitor
- PCNEWS-110 Anhang (bei Webversion)
- 10 Hochverfügbarkeitstechnologien im Überblick
- 11 Datenbankspiegelung
- 12 Replikation
 - 12.1 Konzept
 - 12.2 Transaktionsreplikation
 - 12.3 Mergereplikation
- 13 Transaction Log Shipping (Protokollversand)
- 14 Tipps und Tricks
 - 14.1 Verbindungsserver (*Linked Server*)
 - 14.2 Umbenennen eines Servers, auf dem SQL Server 2005 als eigenständige Instanz ausgeführt wird
 - 14.3 Registrieren des SPN (*Service Principal Name*)

2 SQL Server: Übersicht, Versionen, Editionen

2.1 Versionen

Das Produkt *SQL Server* wurde ursprünglich von der Firma Sybase entwickelt und bis Version 6.5 von Microsoft zugekauft. Nach Differenzen zwischen Microsoft und Sybase entwickelte Microsoft das Produkt ab Version 7.0 selbst weiter. Somit gibt es zwischen Version 6.5 und 7.0 sehr große, auch konzeptuelle Unterschiede, während die Entwicklungen zwischen *SQL Server 7.0* und *2000* minimal sind.

- SQL Server 6.5
- SQL Server 7.0
- SQL Server 2000 (8.0)
- SQL Server 2005 (9.0), aktuell SP2
- SQL Server 2008 (10.0)

SQL Server 2008 erschien im August 2008 und ist in sieben Editionen (analog den im Folgenden beschriebenen Editionen von SQL Server 2005) verfügbar. Neu ist eine Web Edition.

2.2 Editionen im Vergleich

Es gibt folgende Editionen von SQL Server 2005:

- **SQL Server 2005 Enterprise Edition** (32-Bit und 64-Bit): Enterprise Edition ist auf die Leistungsebenen abgestimmt, die zur Unterstützung der größten OLTP-Systeme (*Online Transaction Processing*), hochkomplexer Datenanalysen, Datawarehousing-Systemen und Websites von Unternehmen benötigt werden. Dank der umfassenden Business Intelligence- und Analyse-möglichkeiten von Enterprise Edition sowie der hohen Verfügbarkeit bestimmter Features, z. B. Failover-Clusterunterstützung, kann ein Großteil der unternehmenswichtigen Arbeitsauslastung bewältigt werden. Mit Enterprise Edition ist die umfangreichste SQL Server-Edition verfügbar, die optimal für große Unternehmen und hochkomplexe Anforderungen geeignet ist.

- **SQL Server 2005 Evaluation Edition** (32-Bit und 64-Bit): SQL Server 2005 ist für 32-Bit- und 64-Bit-Plattformen auch als Evaluation Edition verfügbar, die nach 180 Tagen abläuft. SQL Server Evaluation Edition unterstützt dieselbe Featuregruppe wie *SQL Server 2005 Enterprise Edition*. Sie können SQL Server Evaluation Edition für die Verwendung im Produktionsbereich aktualisieren.

- **SQL Server 2005 Standard Edition** (32-Bit und 64-Bit): *SQL Server 2005 Standard Edition* ist die Plattform zur Datenverwaltung und -analyse in kleineren und mittleren Unternehmen. Im Lieferumfang der Edition sind die wesentlichen Funktionen von E-Commerce, Datawarehousing und Branchenlösungen enthalten. Dank der in Standard Edition integrierten Business Intelligence-Möglichkeiten sowie der hohen Verfügbarkeit einzelner Features wird die zur Unterstützung von Unternehmensabläufen erforderliche Funktionalität bereitgestellt. *SQL Server 2005 Standard Edition* ist optimal für kleinere bis mittlere Unternehmen geeignet, die eine umfassende Plattform zur Datenverwaltung und -analyse benötigen.

- **SQL Server 2005 Workgroup Edition** (nur 32-Bit): *SQL Server 2005 Workgroup Edition* ist die

Datenverwaltungslösung für kleinere Unternehmen, die Datenbanken ohne Beschränkung von Größe oder Benutzeranzahl verwenden möchten. *SQL Server 2005 Workgroup Edition* kann entweder als Front-End-Webserver oder für Abläufe in Abteilungen und Zweigstellen eingesetzt werden. Diese Edition enthält die zentralen Datenbankfeatures der SQL Server-Produktlinie und kann problemlos auf *SQL Server 2005 Standard Edition* oder *SQL Server 2005 Enterprise Edition* aktualisiert werden. SQL Server 2005 Workgroup Edition ist eine zuverlässige und robuste Datenbank, die einfach zu verwalten und daher für Einsteiger optimal geeignet ist.

- **SQL Server 2005 Developer Edition** (32-Bit und 64-Bit): *SQL Server 2005 Developer Edition* ermöglicht Entwicklern das Erstellen beliebiger Anwendungen auf der Basis von *SQL Server*. Es schließt die gesamte Funktionalität von *SQL Server 2005 Enterprise Edition* ein, ist jedoch lizenziert für die Verwendung als Entwicklungs- und Testsystem, nicht als Produktionsserver. *SQL Server 2005 Developer Edition* ist erste Wahl für alle, die Anwendungen entwickeln und testen: unabhängige Softwarehersteller (*Independent Software Vendor, ISV*), Berater, Systemintegratoren, Lösungsanbieter und Entwickler in Unternehmen. Sie können *SQL Server 2005 Developer Edition* für die Verwendung im Produktionsbereich aktualisieren.

- **SQL Server 2005 Embedded Edition, auch: Windows Internal Database**: Ist als Feature im Windows Server 2008 enthalten und kann - ohne Konfigurationsoberfläche - als Datenbank für verschiedene Anwendungsservertools genutzt werden.

- **SQL Server 2005 Express Edition (nur 32-Bit)**: Die Datenbankplattform von SQL Server Express basiert auf SQL Server 2005. Sie stellt außerdem einen Ersatz für *Microsoft Desktop Engine (MSDE)* dar. Aufgrund der Integration in Microsoft Visual Studio 2005 ermöglicht SQL Server Express die einfache Entwicklung daten-gesteuerter Anwendungen, die vielfältige Möglichkeiten bieten, sichere Speichermethoden unterstützen und kurzfristig bereitgestellt werden können.

Das frei erhältliche *SQL Server Express* kann weiterverteilt (gemäß Lizenzbestimmungen) und sowohl als Clientdatenbank als auch als einfache Serverdatenbank eingesetzt werden. *SQL Server Express* ist erste Wahl für unabhängige Softwarehersteller (*Independent Software Vendors, ISVs*) und Serverbenutzer sowie für nicht berufsmäßige Entwickler, Entwickler von Webanwendungen, Websitehosts und Laien, die Clientanwendungen erstellen. Wenn Sie erweiterte Datenbankfeatures benötigen, können Sie *SQL Server Express* nahtlos auf anspruchsvollere Versionen von SQL Server aktualisieren.

Darüber hinaus bietet SQL Server Express zusätzliche Komponenten, die als Bestandteil von SQL Server 2005 Express Edition with Advanced Services (SQL Server Express) verfügbar sind. Neben den Features von SQL Server Express enthält SQL Server Express with Advanced Services die folgenden Features:

SQL-Server 2005 Versionen

Funktion	Express	Workgroup	Standard	Enterprise	Bemerkungen
Anzahl der CPUs	1	2	4	Keine Beschränkung	Schließt Unterstützung von Multicore-Prozessoren ein
RAM	1 GB	3 GB	OSMax	OSMax	Speicher ist auf den vom Betriebssystem unterstützten Höchstwert begrenzt
64-Bit-Unterstützung	WOW	WOW	✓	✓	WOW (<i>Windows on Windows</i>)
Datenbankgröße	4 GB	Keine Beschränkung	Keine Beschränkung	Keine Beschränkung	
Partitionierung				✓	Unterstützung umfangreicher Datenbanken
Parallelindeoperationen				✓	Parallelverarbeitung von Indexoperationen
Indizierte Ansichten				✓	Das Erstellen indizierter Ansichten wird in allen Editionen unterstützt. Der Vergleich indizierter Ansichten durch den Abfrageprozessor wird nur in der Enterprise Edition unterstützt.
Management Studio		✓	✓	✓	Vollständige Verwaltungsplattform für SQL Server, enthält <i>Business Intelligence (BI) Development Studio</i>
Auftragsplanungsdienst für den SQL Server-Agent		✓	✓	✓	

- SQL Server Management Studio Express (SSMSE), eine Teilmenge von SQL Server Management Studio.
- Unterstützung für Volltextkataloge.
- Unterstützung für das Anzeigen von Berichten über Reporting Services.

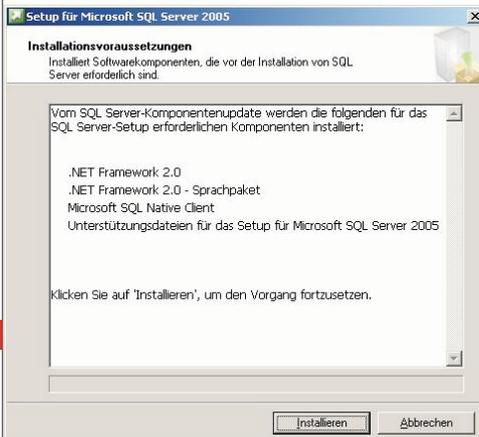
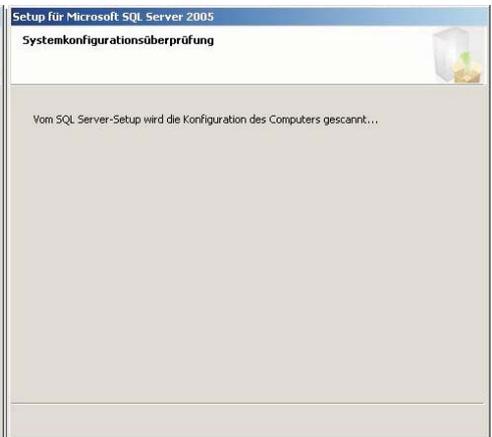
● **SQL Server 2005 Compact Edition** (nur 32-Bit): *SQL Server Compact Edition* ist die kompakte Datenbank, durch die Verwaltungsfunktionen für Unternehmensdaten auf Geräte erweitert werden. SQL Server Compact Edition ist zur Datenreplikation mit SQL Server 2005 und SQL Server 2000 in der Lage, sodass Benutzer einen mobilen Datenspeicher verwalten können, der mit der primären Datenbank synchronisiert ist. *SQL Server Compact Edition* ist die einzige Edition von SQL Server, die relationale Datenbank-Managementfunktionen für intelligente Geräte bereitstellt.

● **SQL Server 2005 Runtime Edition** (32- und 64-Bit): *SQL Server 2005 Runtime Edition* wird über das Microsoft-ISV-Royalty-Lizenzprogramm bereitgestellt. Gemäß dem Endbenutzer-Lizenzvertrag für *SQL Server 2005 Runtime Edition* kann ein unabhängiger Softwarehersteller (*Independent Software Vendor* oder *ISV*) SQL Server-Code in seine Lösung integrieren, vorausgesetzt, der Kunde verwendet den SQL Server-Code nicht zum Ausführen einer anderen Anwendung und verwendet den SQL Server-Code auch in einem anderen Kontext nicht.

3 Installation von SQL Server 2005

3.1 Basisinstallation

Während SQL Server 2005 das .NET-Framework 2.0 voraussetzt, ist für SQL Server 2008 bereits .NET 3.5 Voraussetzung. Das Installationsprogramm von SQL Server 2008 sieht optisch etwas anders aus, verlangt aber im Grunde dieselben Installationsschritte.



Registrierungsinformationen
 Mit den folgenden Informationen personalisieren Sie die Installation.
 Das Feld 'Name' muss ausgefüllt werden, um den Vorgang fortzusetzen. Das Feld 'Firma' ist optional.

Name:

Firma:

Geben Sie den 25 Zeichen langen Product Key ein. Sie finden diese Nummer auf dem geben Aufkleber des CD-Heftes oder der CD-Hülle.

WIKGDC DJ9D1 - CC77E - BFDJ1 - DFBYG

Hilfe < Zurück Weiter > Abbrechen

Zu installierende Komponenten
 Wählen Sie die Komponenten aus, die Sie installieren oder aktualisieren möchten.

SQL Server-Datenbankdienste
 SQL Server-Failovercluster erstellen

Analysis Services
 Analysis Server-Failovercluster erstellen

Reporting Services
 Notification Services
 Integration Services

Arbeitsstationskomponenten, Onlinedokumentation und Entwicklungstools

Klicken Sie auf 'Erweitert', um weitere Optionen anzuzeigen. Erweitert

Hilfe < Zurück Weiter > Abbrechen

Instanzname
 Sie können eine Standardinstanz installieren oder eine benannte Instanz angeben.

Geben Sie einen Namen für die Instanz an. Klicken Sie für eine Standardinstallation auf 'Standardinstanz' und klicken Sie dann auf 'Weiter'. Wenn Sie eine vorhandene Standardinstanz aktualisieren möchten, klicken Sie auf 'Standardinstanz'. Wenn Sie eine vorhandene benannte Instanz aktualisieren möchten, klicken Sie auf 'Benannte Instanz', und geben Sie dann den Instanznamen an.

Standardinstanz
 Benannte Instanz

Hilfe < Zurück Weiter > Abbrechen

Dienstkonto
 Dienstkonten definieren die Anmeldedaten.

Für jedes Dienstkonto anpassen
 Dienst:

Integriertes Systemkonto verwenden

Domänenbenutzerkonto verwenden
 Benutzername:
 Kennwort:
 Domäne:

Dienste am Ende der Installation starten
 SQL Server
 SQL Server-Agent
 SQL-Browser

Hilfe < Zurück Weiter > Abbrechen

Authentifizierungsmodus
 Der Authentifizierungsmodus gibt die Sicherheit an, die beim Herstellen der Verbindung mit SQL Server verwendet wird.

Wählen Sie den Authentifizierungsmodus für diese Installation aus.

Windows-Authentifizierungsmodus
 Gemischter Modus (Windows-Authentifizierung und SQL Server-Authentifizierung)

Geben Sie unten das Anmeldewort für 'sa' an:

Kennwort eingeben:

Kennwort bestätigen:

Hilfe < Zurück Weiter > Abbrechen

Sortierungseinstellungen
 Sortierungseinstellungen definieren das Sortierverhalten für den Server.

Sortierungseinstellungen für den Dienst: SQL Server

Sortierungskennzeichner und Sortierreihenfolge:
 Latin1_General
 Binär
 Binär - Codepunkt
 Unterscheidung nach Groß-/Kleinschreibung
 Unterscheidung nach Kana
 Unterscheidung nach Akzent
 Unterscheidung nach Breite

SQL-Sortierungen (wird für Kompatibilität mit früheren SQL Server-Versionen verwendet)
 Binäreihenfolge, die auf dem Codepunktvergleich basiert, wird verwendet mit...
 Strenge Kompatibilität mit Datenbanken der Version 1.x, ohne Unterscheidung...
 Wörterbuchreihenfolge, Unterscheidung nach Groß-/Kleinschreibung, wird verwendet...
 Wörterbuchreihenfolge, keine Unterscheidung nach Groß-/Kleinschreibung, wird verwendet...
 Wörterbuchreihenfolge, keine Unterscheidung nach Groß-/Kleinschreibung, Gr...

Hilfe < Zurück Weiter > Abbrechen

Einstellungen für Fehler- und Verwendungsberichte
 Helfen Sie Microsoft beim Verbessern einiger SQL Server 2005-Komponenten und -Dienste.

Automatisch Fehlerberichte für SQL Server 2005 an Microsoft oder den Firmen-Fehlerberichtsserver senden. Fehlerberichte enthalten Informationen zum Zustand von SQL Server 2005 zum Zeitpunkt eines Fehlers, zur Hardwarekonfiguration sowie weitere Daten. Fehlerberichte können unbeabsichtigt persönliche Informationen enthalten, die von Microsoft nicht verwendet werden.

Automatisch Daten zur Featureverwendung für SQL Server 2005 an Microsoft senden. Verwendungsdaten enthalten anonyme Informationen zu Ihrer Hardwarekonfiguration und der Verwendung von Software und Diensten.

Klicken Sie auf 'Hilfe', um weitere Informationen zur Fehlerberichtsfunction und zur Art der gesendeten Informationen zu erhalten.

Hilfe < Zurück Weiter > Abbrechen

Installationsbereit
 Die Installation kann nun gestartet werden.

Setup hat genügend Informationen, um mit dem Kopieren der Programmdateien beginnen zu können. Klicken Sie auf 'Installieren', um den Vorgang fortzusetzen. Klicken Sie auf 'Zurück', um die Installationsinstellungen zu ändern. Klicken Sie auf 'Abbrechen', um das Setup zu beenden.

Die folgenden Komponenten werden installiert:

- SQL Server-Datenbankdienste (Datenbankdienste, Replikation, Volltextsuche)
- Clientkomponenten (Konnektivitätskomponenten, Verwaltungstools, Business Intelligence Development Studio, SQL Server-Onlinedokumentation)

Hilfe < Zurück Installieren Abbrechen

Setupstatus
 Die ausgewählten Komponenten werden konfiguriert.

Produkt	Status
MSXML6	
SQL-Setupunterstützungsdateien	Das Setup wurde beendet.
SQL Native Client	Das Setup wurde beendet.
SQL VSS Writer	
OWB11	Komponenten werden konfiguriert...
SQL Server-Abwärtskompatibilitätsdat...	
SQL Server-Datenbankdienste	
Integrierte Entwicklungsumgebung (ID...	

Status
 Programmidentifikatoren werden registriert
 Programmierung: MSCOMCTL.IMGELISTCTRL2

Hilfe << Zurück Weiter >> Abbrechen

Microsoft SQL Server 2005-Setup wird abgeschlossen
 Die Konfiguration von Microsoft SQL Server 2005 wurde beendet.

Informationen zu Setupfehlern finden Sie in den Setupfehlerprotokollen. Klicken Sie auf 'Fertig stellen', um den Installations-Assistenten zu beenden.
[Zusammenfassungsprotokoll](#)

Zum Minimieren der Oberfläche von SQL Server 2005 sind einige Features und Dienste in neuen Installationen standardmäßig deaktiviert. Konfigurieren Sie die Oberfläche mit dem [Oberflächen-Konfigurationstool](#).

Analysis Services
 Wenn Analysis Services von SQL Server 2000 aktualisiert wurde, müssten alle Cubes, Dimensionen und Miningmodelle mit SQL Server Management Studio neu verarbeitet werden.

Reporting Services
 Die von Ihnen beim Setup angegebenen Installationsoptionen von Reporting Services bestimmen, ob eine weitere Konfiguration notwendig ist, bevor Sie auf den Berichtsserver zugreifen können. Der Berichtsserver kann direkt verwendet werden, wenn Sie die [Berichtsserver-Konfigurationstool](#) verwenden.

Hilfe Fertig stellen

3.2 Nachträgliches Hinzufügen von Beispielen oder anderen SQL Server-Komponenten

Systemsteuerung – Software

Software

Zuletzt installierte Programme	Legende anzeigen	Sortieren nach	Name	Größe
Microsoft .NET Framework 2.0			Größe: 138,00 KB	
Microsoft .NET Framework 2.0 Language Pack - DEU			Größe: 178,00 KB	
Microsoft Office 2003 Web Components			Größe: 15,74 MB	
Microsoft Office Professional 2003			Größe: 303,00 MB	
Microsoft SQL Server 2005			Größe: 222,00 MB	
Klicken Sie auf 'Verbinden' oder 'Trennen', um dieses Programm zu verbinden oder zu trennen.			Zuletzt installiert am: 04.08.2007	
Microsoft SQL Server 2005-Überwachungsdiagnostik			Größe: 22,00 MB	
Microsoft SQL Server 2005-Überwachungsdiagnostik (Deutsch)			Größe: 128,00 KB	
Microsoft SQL Server Native Client			Größe: 4,84 MB	
Microsoft SQL Server VSS Writer			Größe: 6,68 MB	
Microsoft Visual Studio 2005 Presser Edition - DEU			Größe: 188,00 MB	
MSXML 6.0 Parser			Größe: 1,30 MB	
SQL Server 2005			Größe: 2,61 MB	
Lernressourcen für das Microsoft SQL Server-Setup (Deutsch)			Größe: 24,11 MB	
Windows Server 2003 Service Pack 1 - Heruntergeladen			Größe: 33,12 MB	

Microsoft SQL Server 2005-Wartung

Komponentenauswahl
 Wählen Sie SQL Server-Komponenten aus, die geändert oder gewartet werden sollen.

Wählen Sie zum Ändern der Installation einer vorhandenen Komponente eine Instanznamen oder eine gemeinsame Komponente aus, und klicken Sie auf 'Weiter'.

[Klicken Sie hier, um eine neue Komponente zu installieren.](#)

SQL Server 2005-Instanzen:
 MSSQLSERVER: Analysis Services, Datenbankmodul

Gemeinsame SQL Server 2005-Komponenten:
 Arbeitsstationskomponenten
 Integration Services
 Notification Services

Bericht...

< Zurück Weiter > Abbrechen Hilfe

Setup für Microsoft SQL Server 2005

Willkommen

Das Setup unterstützt Sie beim Installieren, Ändern oder Entfernen von Microsoft SQL Server. Klicken Sie auf 'Weiter', um den Vorgang fortzusetzen.

< Zurück Weiter > Abbrechen

Setup für Microsoft SQL Server 2005

Systemkonfigurationsüberprüfung
 Bitte warten Sie, während das System auf potenzielle Installationsprobleme überprüft wird.

Erfolg 14 Gesamt 0 Fehler
 13 Erfolg 1 Warnung

Details:

Aktion	Status	Meldung
IIS-Featureanforderung	Warnung	Meldungen...
Ausstehende Neustartanforderung	Erfolg	
Anforderung an den Leistungsindikator	Erfolg	
Anforderung an die Berechtigung für d...	Erfolg	
Anforderung an Internet Explorer	Erfolg	
Anforderung an die COM+-Katalogkom...	Erfolg	
Registrierungsanforderung für die ASP...	Erfolg	
Mindestanforderungen an die MDACV...	Erfolg	

Filter Beenden Beicht

Hilfe Weiter

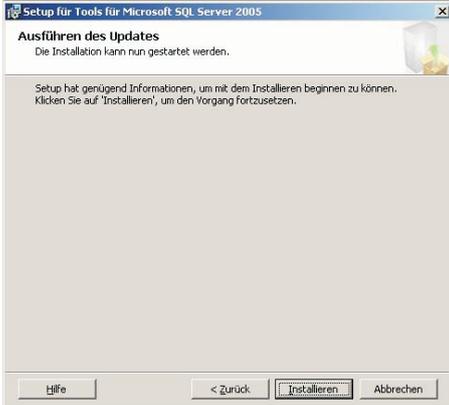
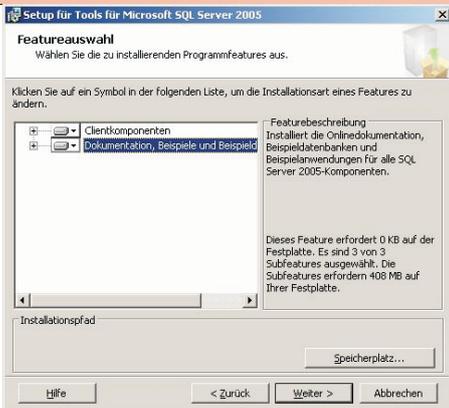
Setup für Tools für Microsoft SQL Server 2005

Instanz ändern oder entfernen
 Sie können diese Instanz von SQL Server ändern oder entfernen.

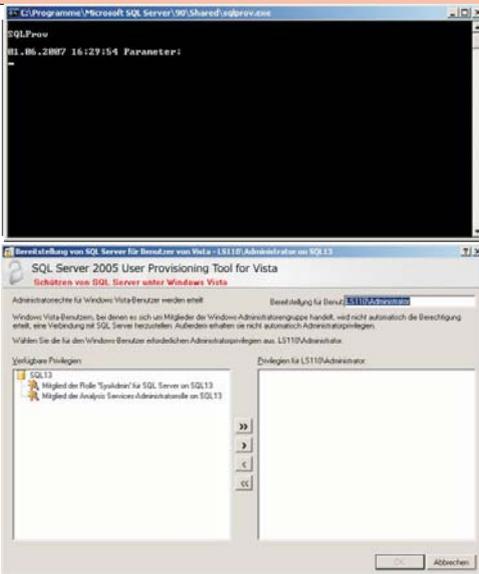
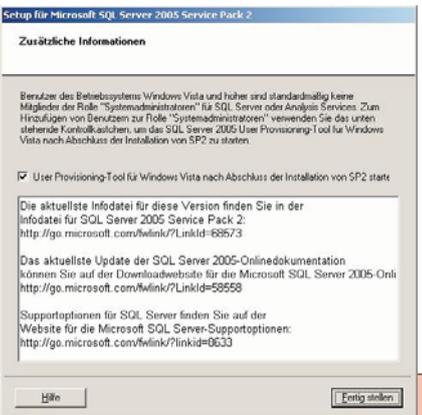
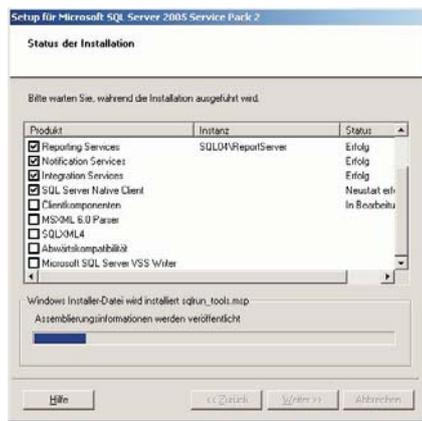
Installierte Komponenten ändern
 Features dieser Instanz wählen.

Microsoft SQL Server entfernen
 Alle Features der vorhandenen Installation vom Computer entfernen.

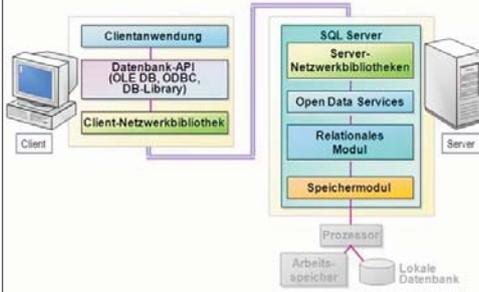
Hilfe < Zurück Weiter > Abbrechen



3.3 Installieren von Service Pack 2

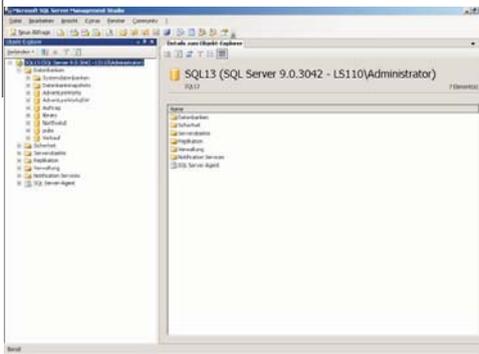


3.4 Architektur von SQL Server 2000/2005

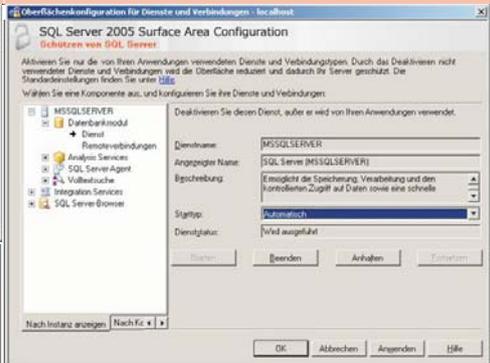


3.5 Administrations-Tools

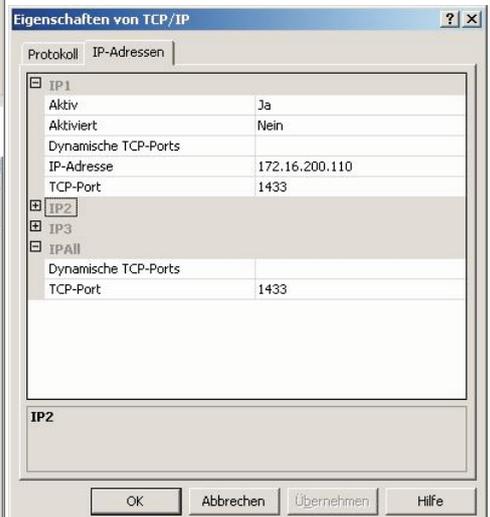
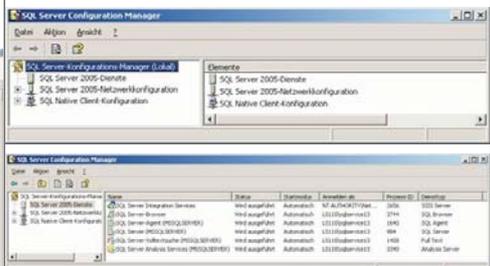
SQL Server 2005 Management Studio: wichtigstes Verwaltungs- und Entwicklungstool, ersetzt die in früheren SQL Server-Versionen üblichen Tools "Enterprise Manager" und "Query Analyzer". Das Oberflächenkonfigurationstool gibt es in SQL Server 2008 nicht mehr als eigenständiges Tool.



SQL Server-Oberflächenkonfigurations-Tool



SQL Server-Konfigurations-Manager



b) TSQL: CREATE LOGIN, ALTER LOGIN, DROP LOGIN

In der Tabelle syslogins (1. Zeile = Benutzer AKopflös) findet man die SQL-Logins:

sevid	sid	status	xdetail
1	NULL	0x0709632BAA9047816CCAB55C3DB0FP	2 2002-11-05
2	NULL	0x0102000000000005000000020020000	32 2002-11-04
3	NULL	0x01050000000000000515000000B8AD111AD2C7A828BA...	30 2002-11-04
4	NULL	0x01	18 2000-09-06
5	0	NULL	192 2002-11-04

Domain Users werden über die SID identifiziert; die SQL Users erkennt man an der wesentlich kürzeren SID. In der Spalte xstatus befindet sich eine Zahl, deren letztes Bit den Serverzugang kennzeichnet (1 – ungerade Zahl = Verbot; 0 – gerade Zahl = Erlaubnis).

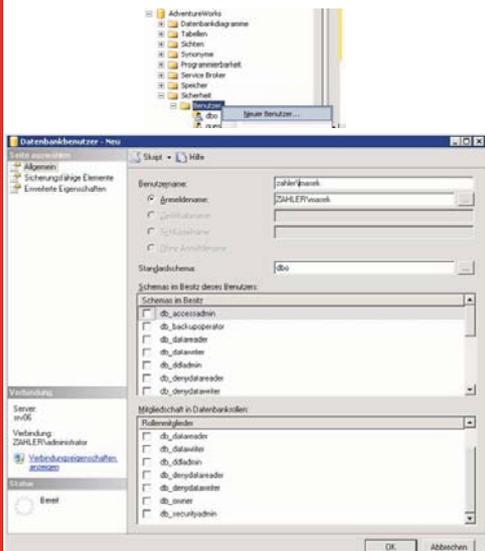
Es kann auch eine Windows-Sicherheitsgruppe einem SQL-Login zugeordnet werden (obiges Beispiel: Administratoren-Gruppe).

	name	password	dbid
1	1.040 AKopflös	0x0100F133DF2B8B98D4...	1
2	2.033 BUILTIN\Administrators	NULL	1
3	2.050 NTFRÄDERS\SQLService	NULL	1
4	2.687 sa	NULL	1
5	2.330 NULL	NULL	0

In der Spalte password ist das SQL-Passwort gespeichert, der Eintrag NULL bedeutet Windows-Authentifizierung.

5.4 Zuordnung eines Logins zu einem DB-User

Hier bestünde noch die Möglichkeit, einen eigenen Usernamen für Zugriffe innerhalb der DB zu erstellen. Aus heutiger Sicht ist das aber nicht mehr nötig, man verwendet denselben Namen wie beim Login.



Selbe Aufgabe mit TSQL: CREATE USER

5.5 Rollen

Die Rolle public stammt eigentlich aus der SQL Server 6.5-Umgebung. Damals durfte jeder Benutzer nur einer "SQL-Gruppe" (heute: Rolle) angehören, wobei die SQL-Gruppe "public" eine übergeordnete Bedeutung hatte, in der alle anderen SQL-Gruppen enthalten waren.

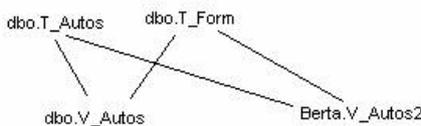
Heute ist jeder Benutzer Mitglied der Rolle public ("Zwangmitgliedschaft").

Permissions werden nur einmal geprüft!

Objekteigentümer (dbo) können ebenfalls Zugriffsrechte vergeben.

Alle Objekte sollten dem Benutzer dbo gehören. Legt ein Mitglied der Serverrolle sysadmin oder der Datenbankrolle db_owner Objekte in einer Datenbank an, so gehören diese Objekte standardmäßig dem Benutzer dbo.

Beispiel

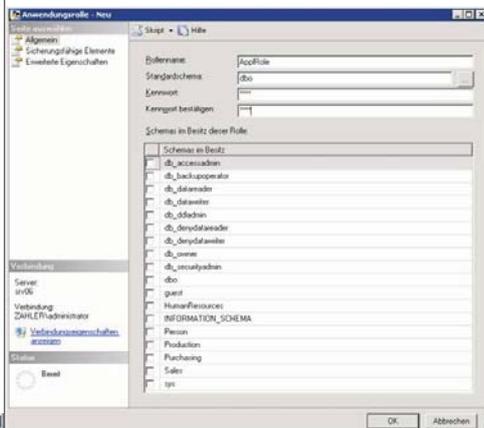


Permissions brauchen nur für die view dbo.V_Autos erteilt werden! Im Falle von Berta.V_Autos2 ("broken ownership chain") müssten zusätzlich Berechtigungen für die beiden Tabellen erteilt werden, die ja einen anderen Besitzer haben – dies ist problematisch!

5.6 Anwendungsrollen

haben keine Mitglieder, müssen mit Passwort gesichert werden.

Zweck: Wenn ein Benutzer nur über ein Clientprogramm auf den SQL Server zugreifen soll



use northwind

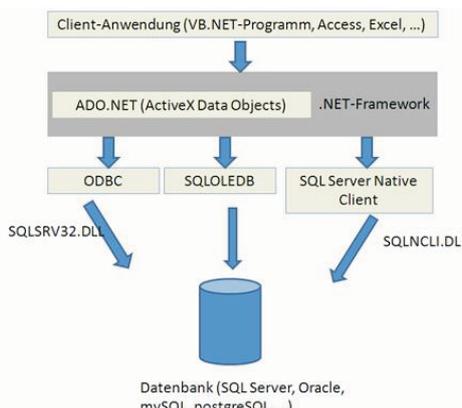
```

go
create procedure employee_proc as
select FirstName, LastName, Title
from employees
grant select on employees_view to public
grant select on employee_proc to public
revoke select on employees from public
Normalerweise wird dem Benutzer nicht erlaubt, auf die Datenbank zuzugreifen, erst durch "Annehmen" der Anwendungsrolle (dies ist im Client-Programm programmiert) erhält er die nötigen Berechtigungen.
exec sp_setapprole 'order_entry', 'password'
select * from products
    
```

6 Client-Zugriff auf MS SQL Server 2005

6.1 Grundlagen

Um eine (Server-)Datenbank programmier-technisch anzusprechen, ist es nötig, eine Schnittstelle zu definieren. Grundsätzlich gilt: Es ist nicht möglich, die Datenbank direkt anzusprechen.

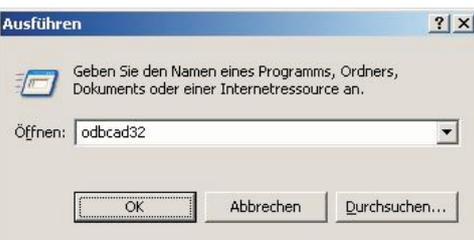


6.2 MS Access 2007 als Client mit Hilfe einer ODBC-Systemschnittstelle

Ein relativ einfaches Verfahren zur Erstellung eines SQL Server-Clients bietet MS Access (ab Version 2003). Der eigentliche Datenbankzugriff wird von einer ODBC-Schnittstelle durchgeführt.

ODBC (Open DataBase Connectivity) stellt über spezielle Treiber (ODBC-Treiber) eine Programmierschnittstelle bereit, die standardmäßig (von Access oder durch VB-Programmierung) angesprochen werden kann.

Schritt 1: Einrichten einer ODBC-Schnittstelle: mit dem ODBC-Datenquellen-Administrator Start – Ausführen – odbcad32



Der ODBC-Datenquellen-Administrator erlaubt die Erstellen von drei Schnittstellentypen, die auch als DSN (data source name, Datenquellennamen) bezeichnet werden:

- **Benutzer-DSN:** Diese Schnittstelle kann nur von dem Benutzer verwendet werden, der sie erstellt hat.
- **System-DSN:** Diese Schnittstelle steht allen Benutzern und dem lokalen Systemkonto zur Verfügung.
- **Datei-DSN:** Die Schnittstellenparameter werden in einer *.dsn-Datei gespeichert und können so auf andere PCs transportiert werden.

(a) Verwenden des ODBC-Treibers für SQL Server (SQLSRV32.DLL; verwendbar für Versionen ab SQL Server 7.0)



Auf „Hinzufügen“ klicken, dann den ODBC-Treiber für SQL Server (SQLSRV32.DLL) auswählen:



Auf „Fertigstellen“ klicken.

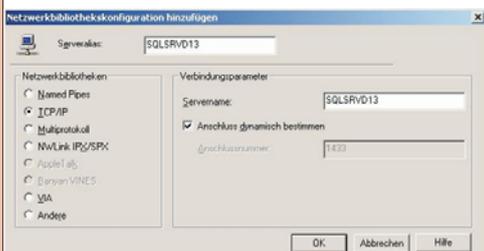


Der Name ist als DSN-Name zu verstehen, der zukünftig für das Ansprechen der Datenbank verwendet wird.

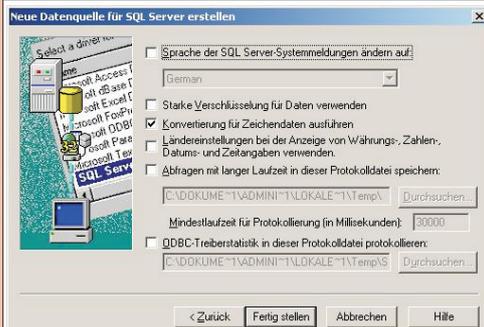


Hier wählen Sie bitte aus, ob Windows- oder SQL Server-Authentifizierung verwendet werden soll.

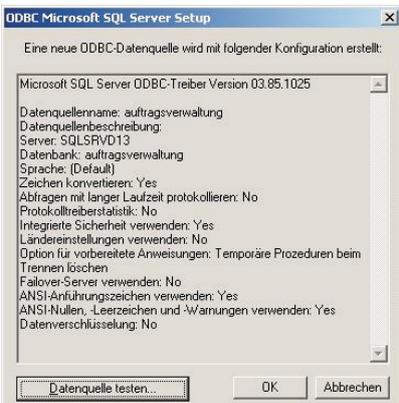
Unter „Clientkonfiguration“ überprüfen Sie, ob TCP/IP als verwendete Netzwerkbibliothek eingestellt ist:



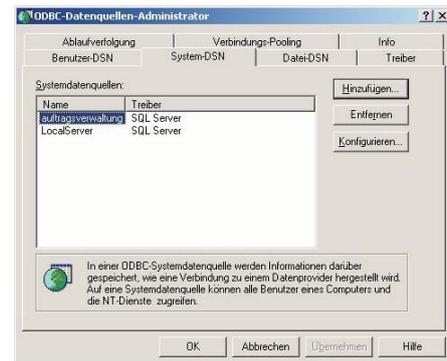
Wählen Sie anschließend die zu verwendende Datenbank:



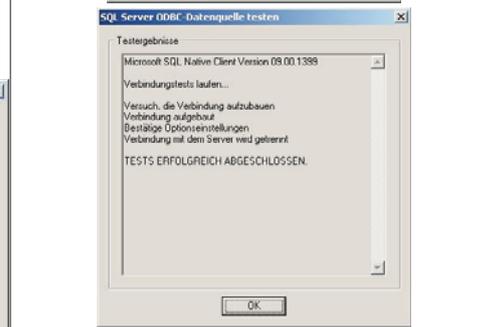
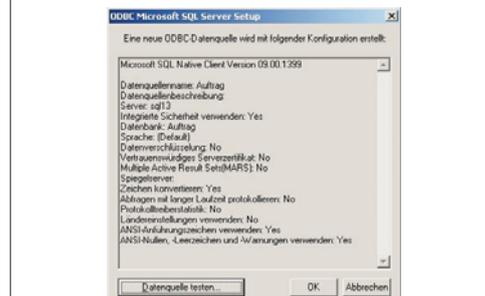
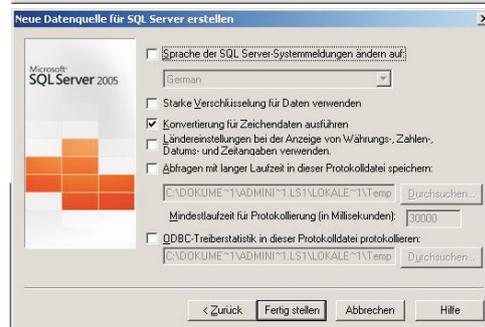
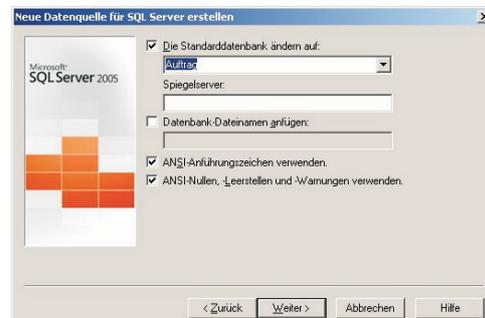
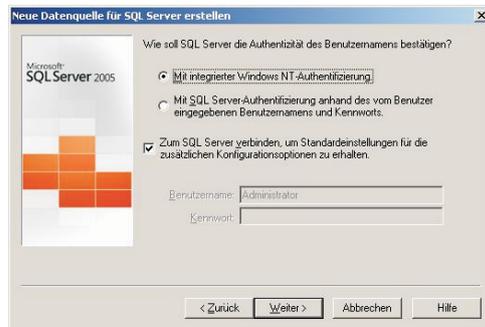
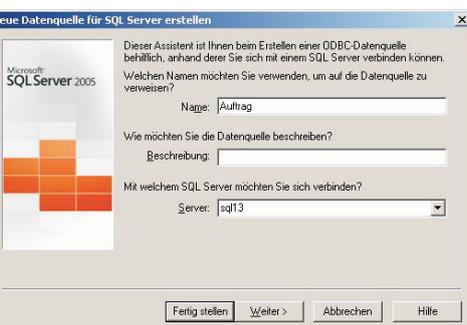
Mit „Datenquelle testen...“ können Sie den Zugriff auf die Server-Datenquelle überprüfen:



Wenn Sie die ODBC-Schnittstelle erfolgreich erstellt haben, sollte das ungefähr so aussehen:



(b) Verwenden des ODBC-Treibers für SQL Native Client (SQLNCLI.DLL; verwendbar ab SQL Server 2005)



Schritt 2: Erstellen verknüpfter Tabellen in Access

Legen Sie zunächst eine neue Access-Datenbank an.



Leere Datenbank

Erstellen Sie eine Microsoft Office Access-Datenbank, die keine vorhandenen Daten oder Objekte enthält.

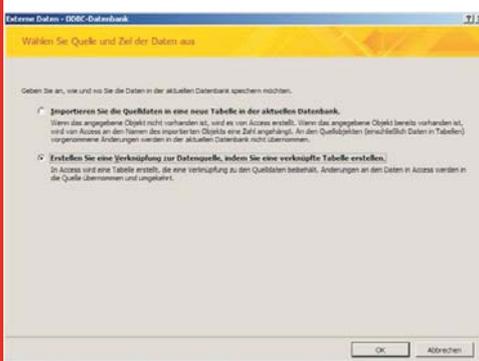
Dateiname:

C:\Dokumente und Einstellungen\Administrator.LS.110\Eigene Dateien\

Nun wählen Sie den wählen Sie im Ribbon „Externe Daten“ das Symbol für „Weitere Datenbankformate importieren“ aus und wählen „ODBC-Datenbank“:



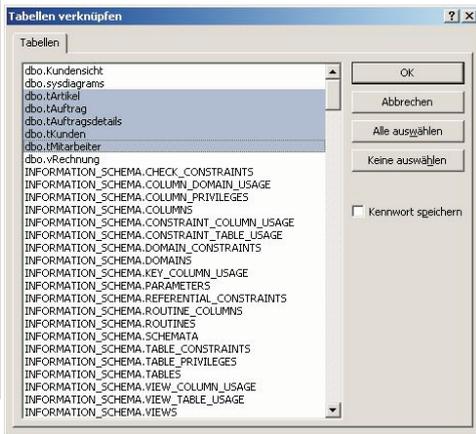
Wählen Sie im erscheinenden Dialog den Punkt „Erstellen Sie eine Verknüpfung zur Datenquelle, indem Sie eine verknüpfte Tabelle erstellen“:



Im Menüpunkt „Datenquelle auswählen“ aktivieren Sie die Karteikarte „Computerdatenquelle“ und wählen die vorher konfigurierte ODBC-Schnittstelle aus:



Wählen Sie dann die zu verknüpfenden Tabellen aus:



Ergebnis:

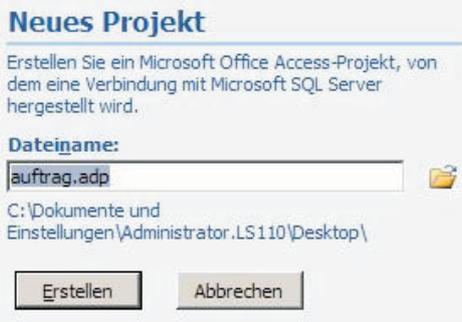


Auf Basis dieser Verknüpfungen können nun Abfragen, Formulare und Berichte erstellt werden.

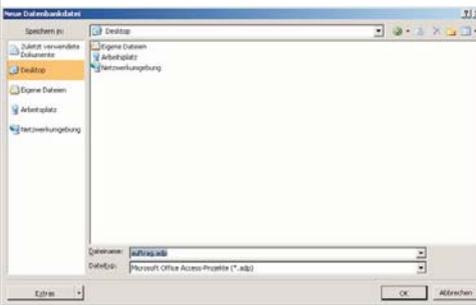
6.3 MS Access-Datenbankprojekte (ohne ODBC-Schnittstelle)

Eine zweite Möglichkeit besteht in der Verwendung einer Access-internen Zugriffsmöglichkeit, die aber erst seit Access 2003 fehlerfrei und stabil arbeitet.

Datenbankprojekte werden als *.ADP (Access Data Project) gespeichert.



Speichern Sie das Projekt:

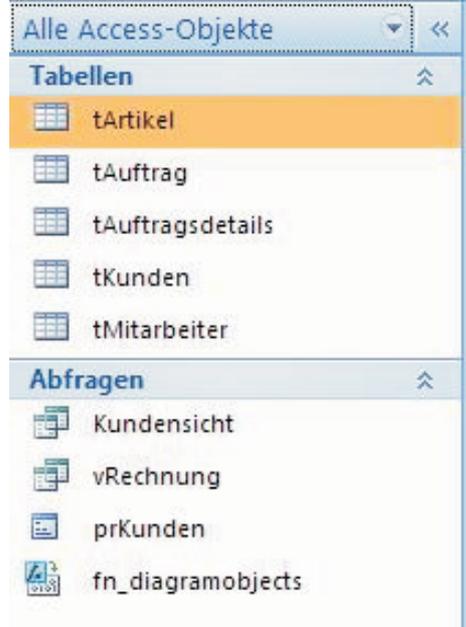


Wählen Sie in diesem Dialog den SQL-Server, die Art der Authentifizierung und die Datenbank aus.

Die Verbindung kann auch getestet werden:



Man sieht, dass hier nicht nur Tabellenzugriffe übernommen wurden, sondern auch Sichten und gespeicherte Prozeduren (unter „Abfragen“).



7 Migration einer Access 2007-Datenbank auf SQL Server 2005

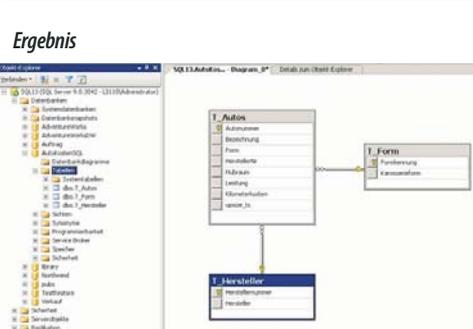
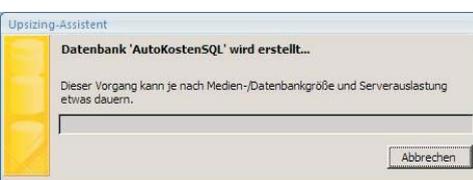
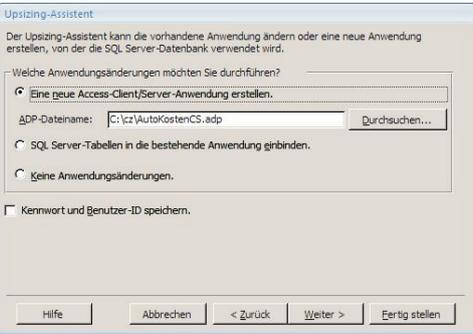
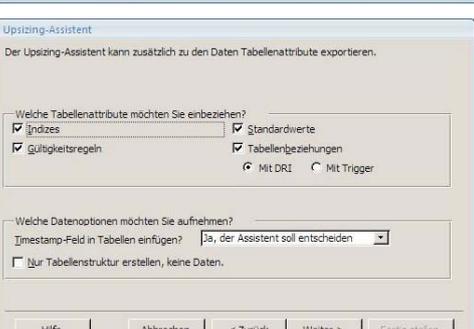
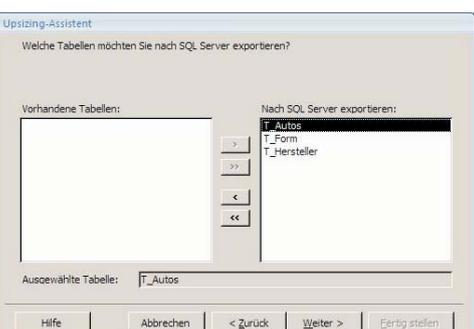
7.1 Upgrade mit dem Access 2007-Upsizing-Assistenten

Öffnen Sie die Access-Datenbank und wählen Sie aus dem Menüband "Datenbanktools" das Symbol "SQL Server":



SQL Server
Einige oder alle Bestandteile einer Datenbank zu einer neuen oder vorhandenen Microsoft SQL Server-Datenbank migrieren.

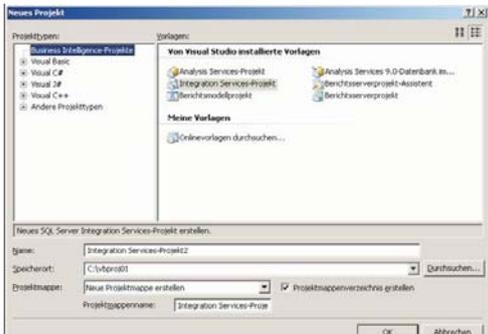
Es startet der "Upsizing-Assistent", mit dem Sie sowohl eine neue SQL Server-Datenbank erstellen können, als auch eine vorhandene SQL Server-Datenbank mit Daten befüllen können.



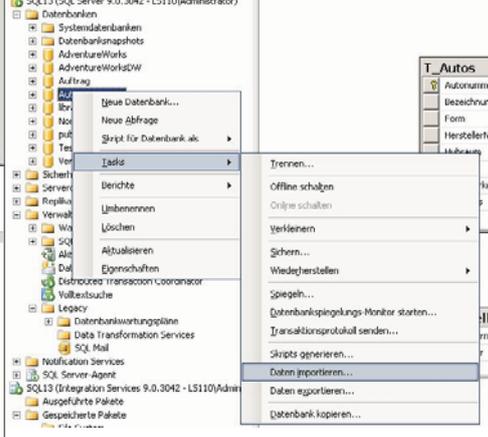
Hinweis: Abfragen werden nicht übernommen; weder werden Sie in Views oder Procedures am SQL Server konvertiert, noch im ADP-Projekt gespeichert.

7.2 Datenimport aus einer Access-Datenbank mit dem SQL Server Integration Services (SSIS)-Import/Export-Assistent

Variante 1: Starten Sie das SQL Server Business Intelligence Development-Studio und erstellen Sie ein neues Integration Services-Projekt
Im Projektmappen-Explorer klicken Sie mit der rechten Maustaste auf "SSIS-Pakete" und wählen aus dem Kontextmenü [SSIS-Import/Export-Assistent].



Variante 2: Führen Sie in einem Eingabeaufforderungsfenster `DTSWizard.exe` aus. Diese Datei ist im Verzeichnis `c:\Programme\Microsoft SQL Server\90\DTSS\Binn` gespeichert.
Variante 3: Im SQL Server Management Studio Kontextmenü einer Datenbank auswählen, [Tasks] – [Daten importieren]



SQL Server-Import/Export-Assistent

Datenquelle auswählen
Wählen Sie die Quelle aus, von der Daten kopiert werden sollen.

Datenquelle:

Wählen Sie zum Herstellen einer Verbindung einer Datenbank aus, und geben Sie einen Benutzernamen und ein Kennwort an. Möglicherweise müssen Sie erweiterte Optionen angeben.

Dateiname:

Benutzername:

Kennwort:

SQL Server-Import/Export-Assistent

Tabelle kopieren oder Datenbank abfragen
Geben Sie an, ob mindestens eine Tabelle oder Sicht kopiert werden soll oder ob die Ergebnisse einer Abfrage aus der Datenquelle kopiert werden sollen.

Daten aus mindestens einer Tabelle oder Sicht kopieren
Mit dieser Option können Sie alle Daten aus den vorhandenen Tabellen und Sichten in der Quelldatenbank kopieren.

Abfrage zum Angeben der zu übertragenden Daten schreiben
Mit dieser Option können Sie eine SQL-Abfrage zum Ändern oder Einschränken der Quelldaten für den Kopiervorgang schreiben.

SQL Server-Import/Export-Assistent

Vorgang wird ausgeführt...
Klicken Sie auf die Schaltfläche 'Beenden', um den Vorgang abzubrechen.

5 verbleiben 7 Gesamt
2 Erfolg
0 Fehler
0 Warnung

Details:

Aktion	Status	Meldung
Datenflusktask wird initialisiert	Erfolg	
Verbindungen werden initialisiert	Erfolg	
SQL-Befehl wird festgelegt	0 % abgeschlossen	
Quellverbindung wird festgelegt		
Zielverbindung wird festgelegt		
Überprüfung wird ausgeführt		
Speichern		

- .Net Framework Data Provider for Odbc
- .Net Framework Data Provider for Oracle
- .Net Framework Data Provider for SqlServer
- Flatfilequelle
- Microsoft Access
- Microsoft Excel
- Microsoft Office 12.0 Access Database Engine OLE DB Provid...
- Microsoft OLE DB Provider for Analysis Services 9.0
- Microsoft OLE DB Provider For Data Mining Services
- Microsoft OLE DB Provider for OLAP Services 8.0
- Microsoft OLE DB Provider for Oracle
- Microsoft OLE DB Provider for SQL Server
- SQL Native Client
- SQLXMOLEDB
- SQLXMOLEDB.4.0

SQL Server-Import/Export-Assistent

Quellentabellen und -sichten auswählen
Wählen Sie mindestens eine Tabelle oder Sicht zum Kopieren aus.

Tabellen und Sichten:

Quelle	Ziel
<input checked="" type="checkbox"/> T_Autos	[AutokostenSQL][dbo].[T_Autos]
<input checked="" type="checkbox"/> T_Form	[AutokostenSQL][dbo].[T_Form]
<input checked="" type="checkbox"/> T_Hersteller	[AutokostenSQL][dbo].[T_Hersteller]
<input type="checkbox"/> 'Autos mit Detaildaten'	
<input type="checkbox"/> 'Formhäufigkeiten'	
<input type="checkbox"/> 'Herstellereübersicht'	
<input type="checkbox"/> 'Technische Daten'	

Für mehrere Tabellen optimieren
 In einer Transaktion ausführen

Ein neues SSIS-Paket wird erzeugt.
Ablaufsteuerung:

```

    graph TD
      A[SQL-Task 'Vorbereitung'] --> B[Datenflusktask]
    
```

SQL Server-Import/Export-Assistent

Ziel auswählen
Geben Sie an, wohin die Daten kopiert werden sollen.

Ziel:

Servername:

Authentifizierung:
 Windows-Authentifizierung verwenden
 SQL Server-Authentifizierung verwenden

Benutzername:

Kennwort:

Datenbank:

SQL Server-Import/Export-Assistent

Spaltenzuordnungen

Quelle: T_Hersteller
Ziel: [AutokostenSQL][dbo].[T_Hersteller]

Zielabgabe erstellen SQL bearbeiten...
 Zielen in Zielabgabe löschen Zielabgabe löschen und erneut erstellen
 Zielen an die Zielabgabe anfügen IDENTITY_INSERT aktivieren

Zuordnungen:

Quelle	Ziel	Typ	NULL	Größe	Genau.	Dest.
Herstellernummer	Herstellernummer	int	<input checked="" type="checkbox"/>			
Hersteller	Hersteller	nvarchar	<input checked="" type="checkbox"/>	50		

Quellpaket:

Datenflusktask:

Datenbank erstellen

Geben Sie den Namen und Eigenschaften für die SQL Server-Datenbank an.

Name:

Datendateiname:

Protokolldateiname:

Größe der Datendatei:
Anfangsgröße: MB
 Keine Vergrößerung zulässig
 Vergrößerung nach Prozent:
 Vergrößerung nach Größe: MB

Größe der Protokolldatei:
Anfangsgröße: MB
 Keine Vergrößerung zulässig
 Vergrößerung nach Prozent:
 Vergrößerung nach Größe: MB

SQL Server-Import/Export-Assistent

Assistenten abschließen
Überprüfen Sie die im Assistenten vorgenommene Auswahl, und klicken Sie dann auf 'Fertig stellen'.

Klicken Sie auf 'Fertig stellen', um die folgenden Aktionen auszuführen:

- Zeilen kopieren von "T_Autos" zu [AutokostenSQL][dbo].[T_Autos]
Die neue Zielabgabe wird erstellt.
- Zeilen kopieren von "T_Form" zu [AutokostenSQL][dbo].[T_Form]
Die neue Zielabgabe wird erstellt.
- Zeilen kopieren von "T_Hersteller" zu [AutokostenSQL][dbo].[T_Hersteller]
Die neue Zielabgabe wird erstellt.
- Das Paket wird in der Paketdatei 'C:\vbproj01\Integration Services-Projekt1\Integration Services-Projekt1\Package1.dtsx' gespeichert.
- Das Paket wird nicht sofort ausgeführt.

Management Studio:

Objekt-Explorer

- Verbinden
- Datenbankmodul...
- Analysis Services...
- Integration Services...
- Reporting Services...
- SQL Server Compact Edition...

Paket importieren

Paket Speicherort:

Server:

Authentifizierung:
Authentifizierungstyp:

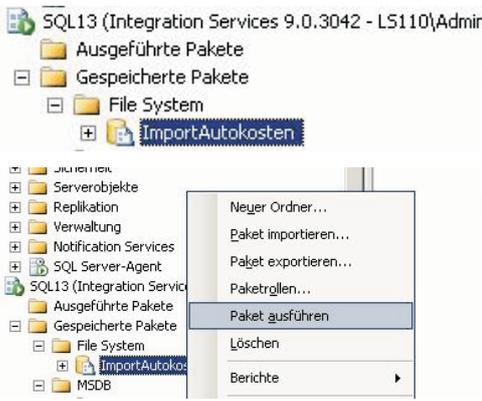
Benutzername:

Kennwort:

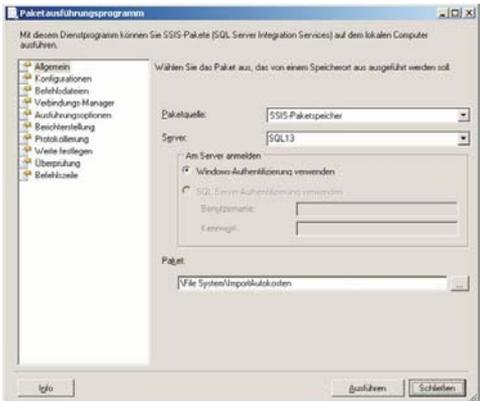
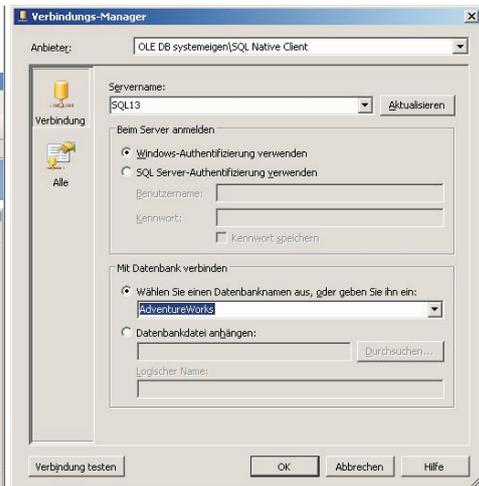
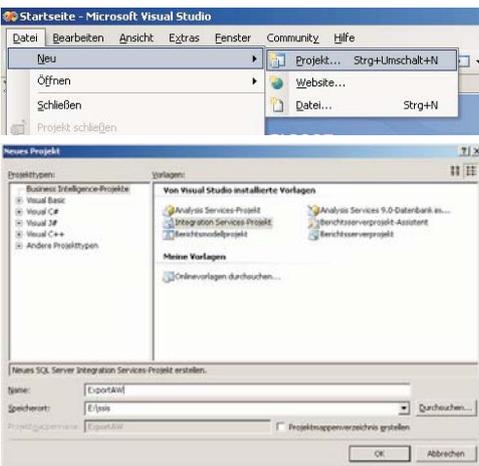
Paketpfad:

Paket importieren als:
Paketname:

Schutzebene:

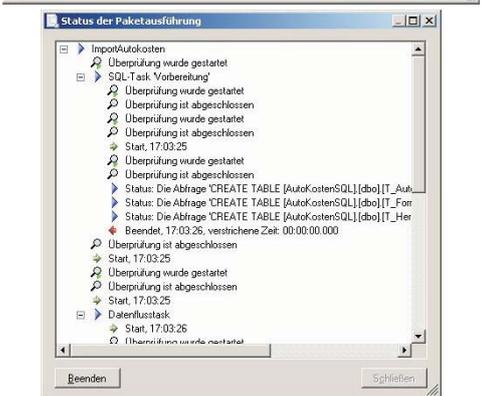
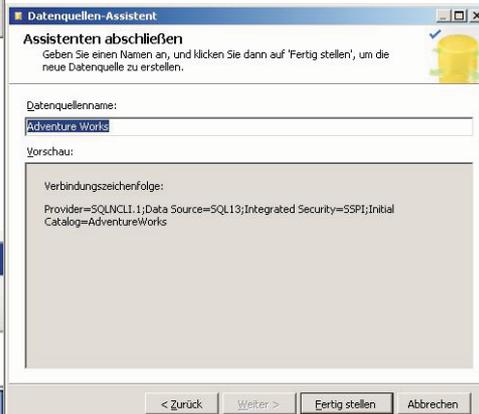


Erstellen Sie ein neues Integration Services-Projekt:



Achten Sie darauf, einen passenden Projektnamen und Speicherort anzugeben.

Schritt 2: Konfigurieren Sie eine Datenquelle. Dazu klicken Sie im Projektmappen-Explorer mit der rechten Maustaste auf „Datenquellen“ und wählen [Neue Datenquelle...].



ADO-Connection-String

Provider=SQLNCLI.1;Data Source=SQL13;Integrated Security=SSPI;Initial Catalog=AdventureWorks

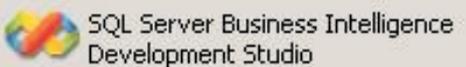


Schritt 3: Erstellen Sie aus der Datenquelle einen Verbindungs-Manager

Beachten Sie: Es sind keine Fremdschlüssel-schränkungen vorhanden!

8 SSIS

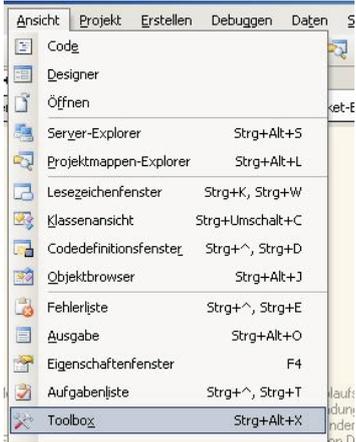
Schritt 1: Starten Sie SQL Server Business Intelligence Development Studio.



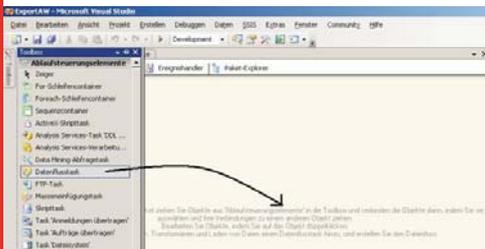
Es startet Visual Studio, dessen Kernbestandteile auch bei einer SQL Server 2005-Installation mitinstalliert werden.



Schritt 4: Erstellung eines Datenflusstasks
Blenden Sie zunächst die Toolbox ein:



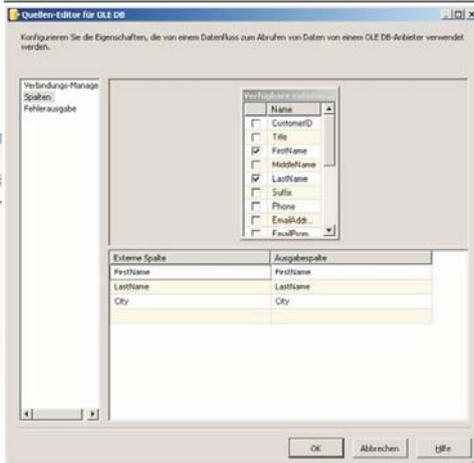
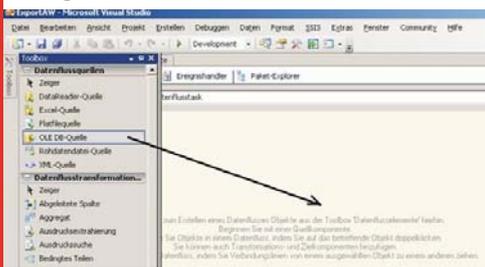
Ziehen Sie aus der Toolbox das Element „Datenflusstask“ in den Designer-Bereich.



Wechseln Sie zur Karteikarte „Datenfluss“:



Ziehen Sie nun das Element „OLE DB-Quelle“ aus dem Toolbox-Bereich „Datenflussquellen“ in den Designer-Bereich.



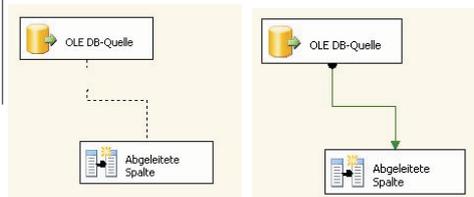
Klicken Sie auf **OK**. Nun ist das rote X neben dem Datenquellensymbol verschwunden, da die Quelle ordnungsgemäß konfiguriert ist.



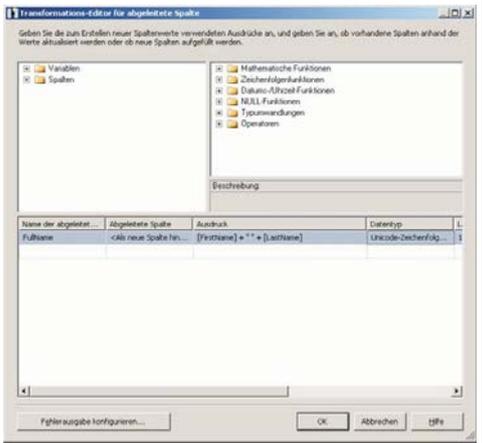
Ziehen Sie nun aus dem Bereich „Datenflustransformationen“ der Toolbox das Element „Abgeleitete Spalte“ in den Designer-Bereich.



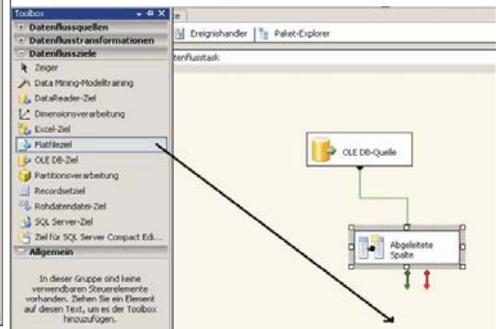
Wählen Sie die OLE DB-Quelle aus und ziehen Sie nun den grünen Pfeil zum Element „Abgeleitete Spalte“:



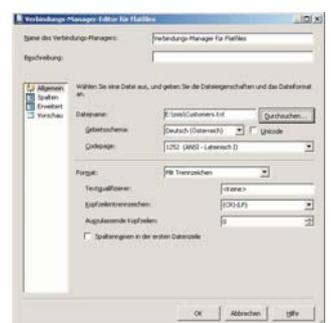
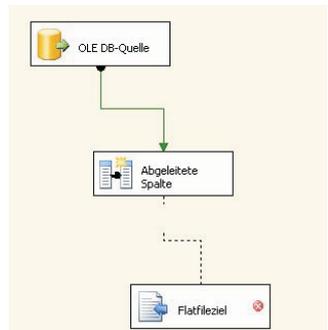
Bearbeiten Sie nun die Eigenschaften des Elements „Abgeleitete Spalte“:

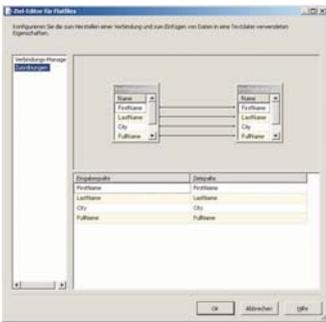


Als nächstes ziehen Sie aus dem Bereich Datenflussziele der Toolbox das Element „Flatfileziel“ in den Designer-Bereich.

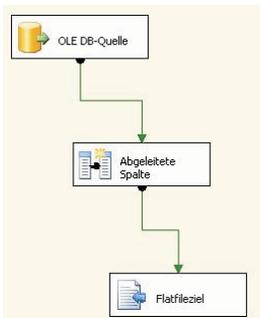


Ziehen Sie nun den grünen Pfeil des Elements „Abgeleitete Spalte“ zum Element „Flatfileziel“:





Damit ist der Datenflusstask fertig konfiguriert:

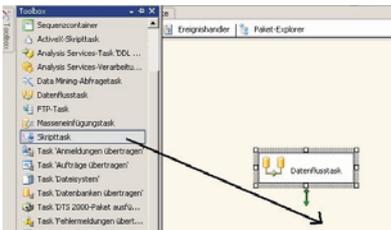


Schritt 5: Hinzufügen eines Skripttasks im Anschluss an den Datenflusstask

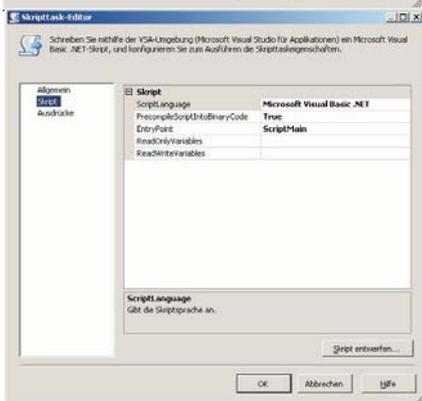
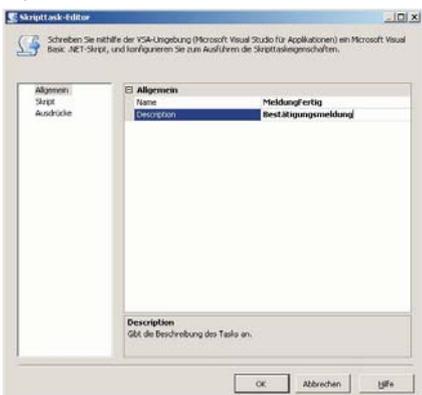
Wechseln Sie zur Karteikarte „Ablaufsteuerung“:



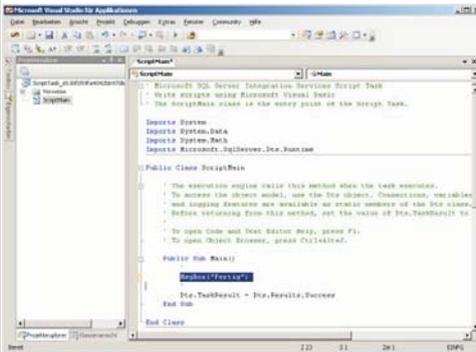
Ziehen Sie aus der Toolbox das Element „Skripttask“ in den Designer-Bereich:



Bearbeiten Sie nun die Eigenschaften des Skripttasks:

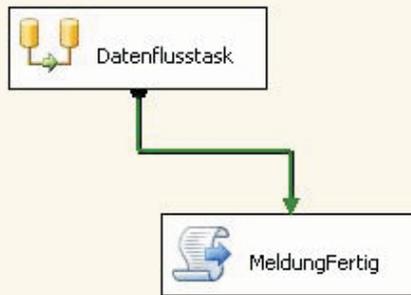


Klicken Sie auf „Skript entwerfen...“:



[Datei] – [Schließen und zurück]

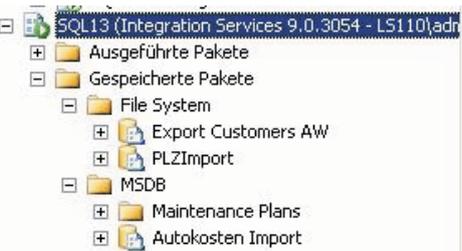
Schließlich ziehen Sie im Designerbereich den grünen Pfeil vom Datenflusstask zum Task MeldungFertig.



Damit haben Sie ein Beispiel-SSIS-Paket konfiguriert.

Zu Testzwecken können Sie das Paket über den Menüpunkt [Debuggen] – [Debuggen starten] ausführen.

Das Paket kann nun auch im Dateisystem oder in der MSDB-Datenbank gespeichert werden. Vom SQL Server Management Studio ist es dann möglich, diese Pakete auszuführen bzw. mit einem SQL Server Agent-Zeitplan zu verknüpfen.



9 Backup und Restore

9.1 Backup-Grundlagen

1. Lokales Band

mit SQL Server-eigener Software empfehlenswert für kleinere Umgebungen

2. File Backup

Sicherung mit SQL-Server eigener Software auf lokale Datei, diese Datei wird von zentraler Backup Server Lösung gesichert

Vorteil:

- keine zusätzlichen Probleme durch Drittanbietersoftware

Nachteil:

- hoher Speicherplatzbedarf

3. Drittanbieter-Software-Agent

greift auf die SQL Server-API zu und sichert auf eine zentrale Bandstation.

Beispiele: BrightStor ARCserve Backup v9, Seagate Backup Exec, ...

4. Offline Backup

SQL Server-Dienst beenden, Daten- und Transaktionsdatei sichern

Nachteile:

- kein Arbeiten beim Sichern möglich
- Transaktions-Log kann nicht genutzt werden

Beispiel:

C: ... Windows Server 2003 + SQL Server 2005

D: ... SQL Server Datenfiles

E: ... SQL Server Transaktionsprotokolle

Full Backup wurde am Mi 01:00 Uhr durchgeführt, Sicherung Transaktionslog Mi 09:00

Um 11:00 fällt Datenplatte D: aus.

Was macht man:

- aktuelles Transaktionsprotokoll sichern!!!!
- NeuePlatteeinsetzen
- DBFullBackupRestore(StandMi01:00)
- Transaktionslog Mi09:00Uhrrestore
- aktuelles Transaktionslogrestore

Anlegen von Backup-Geräten

```
USE master
EXEC sp_addumpdevice 'disk', 'Nwstripe1', 'C:\Backup\Nwstripe1.bak'
EXEC sp_addumpdevice 'disk', 'Nwstripe2', 'C:\Backup\Nwstripe2.bak'
```

Durchführen eines Backups (Anhängen)

```
BACKUP DATABASE Northwind TO Nw1
WITH NOINIT,
DESCRIPTION = 'The second full backup of Northwind'
```

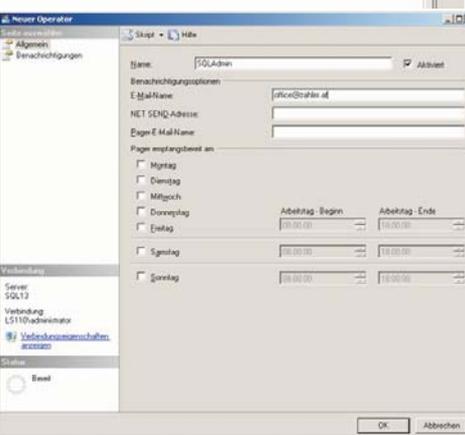
Differentielles Backup

```
BACKUP DATABASE Northwind TO DISK = 'C:\Backup\Nwdiff.bak'
WITH NOINIT, DIFFERENTIAL
```

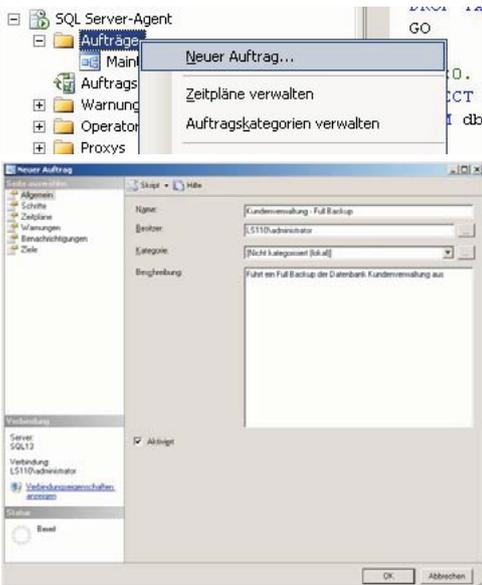
Automatisieren von Backup-Jobs

Backup-Jobs sollten unbedingt mit SQL Server Agent automatisiert werden.

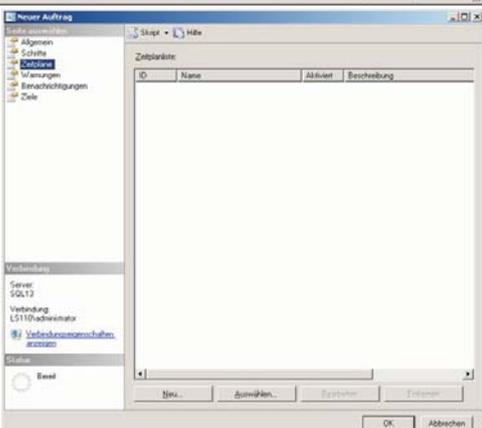
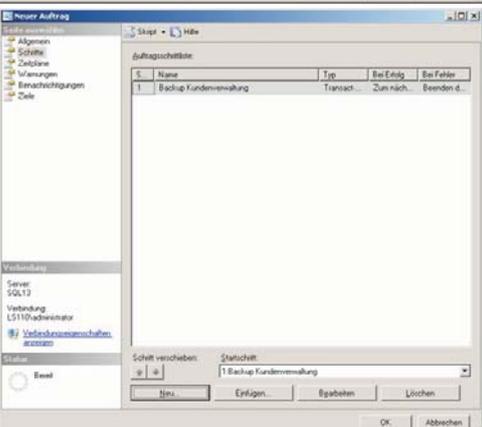
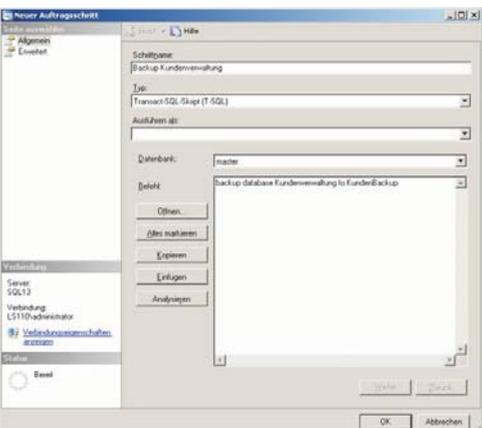
Erstellen Sie zunächst einen neuen Operator:



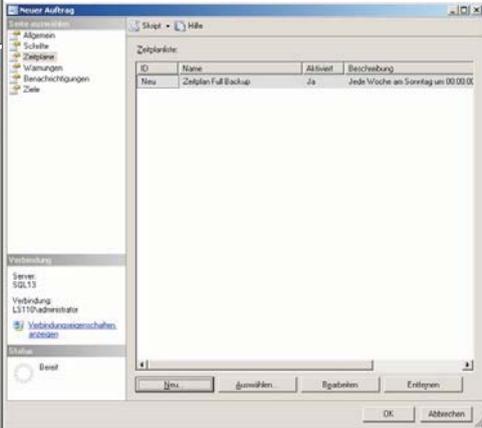
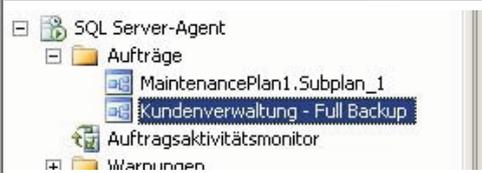
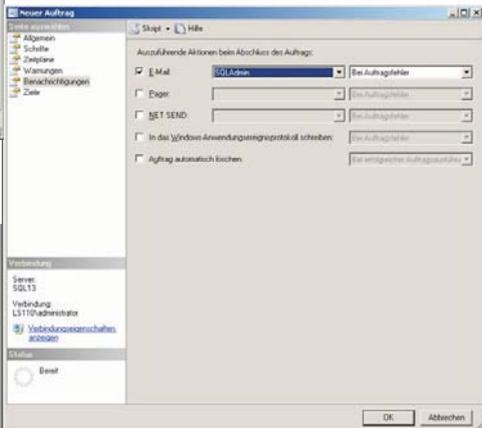
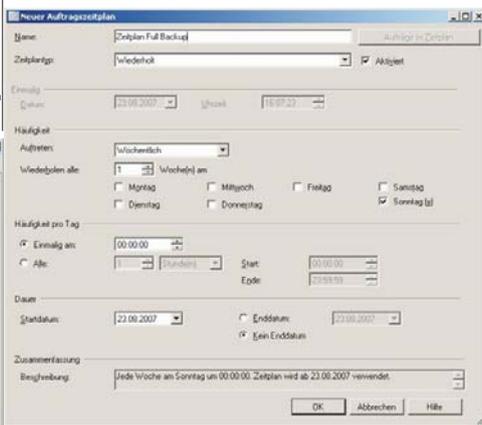
Erstellen Sie nun einen neuen Auftrag:



Klicken Sie in der linken Spalte auf „Schritte“, dann auf die Schaltfläche „Neu“:



Klicken Sie auf die Schaltfläche „Neu...“, um einen neuen Zeitplan zu erstellen:



9.2 Restore

Normalsyntax

```
RESTORE DATABASE Northwind FROM NWindBackup
RESTORE DATABASE Northwind FROM NWindBackup
WITH FILE=2, NORECOVERY
```

Beispiel

Schritt 1 – eine Testdatenbank "NWCOPY" wird rüch gesichert

```
RESTORE DATABASE NWCOPY FROM DISK =
'C:\Backup\NWC1.bak'
WITH REPLACE, RECOVERY
EXEC sp_dboption 'nwcop', 'single user',
'FALSE'
USE NWCOPY
GO
```

Schritt 2 – ein neues Produkt wird in die Tabelle Products eingefügt

```
INSERT products(productID, ProductName,
SupplierID, CategoryID, QuantityPerUnit,
UnitPrice, UnitsInStock, UnitsOnOrder,
ReorderLevel, Discontinued)
Values(150,'Maple Flavor Pancake Mix',15,0,'12
per case',1.27,5,5,1,0)
SELECT * FROM products WHERE ProductName =
'Maple Flavor Pancake Mix'
```

Schritt 3 – ein Backup-Device wird erzeugt und die Datenbank gesichert

```
USE MASTER
GO
sp_addumpdevice 'disk',
'NWC2','c:\backup\NWC2.bak'
---Backup the database
BACKUP DATABASE NWCOPY to NWC2
WITH FORMAT, NAME = 'NWCOPY-Full',
DESCRIPTION = 'A single file full backup of
NWCOPY'
```

Spezielsyntax

restore database with recovery würde keine Rücksicherung durchführen, aber eine versehentlich nicht online geschalteter DB (Zum Beispiel nach dem Einspielen des letzten Transaktionsprotokolls) online bringen.

Point-in-time Recovery

Stellt Datenbank bis zu einem definierten Zeitpunkt wieder her. Ausgangspunkt natürlich immer FullBackup.

```
use master
restore database db_video from videobackupfull
with file=1,norecovery
restore log db_video from videologbackup
with file=3,recovery,stopat='November 7, 2002
09:09 AM'
```

Übungsbeispiel BACKUP/RESTORE

Anlegen einer Datenbank

```
use master
create database Kundenverwaltung
on primary
(name='Kunden',
filename='E:\Kunden\Kunden.mdf',
size=5 MB,
filegrowth=10%
)
log on
(name='KundenLog',
filename='E:\Kunden\KundenLog.ldf',
size=1 MB,
maxsize=5 MB,
filegrowth=1 MB
)
go
use Kundenverwaltung
create table tKunden
( KdNr int identity(1,1) primary key,
Vorname nvarchar(50) NULL,
Nachname nvarchar(50) NOT NULL,
Zeit datetime NOT NULL
)
go
insert tKunden (Vorname, Nachname, Zeit)
values ('Christian', 'Zahler', getdate())
insert tKunden (Vorname, Nachname, Zeit)
values ('Matthias', 'Jandl', getdate())
```

Test
select * from tKunden

Backup Device anlegen

```
exec sp_addumpdevice 'disk', 'KundenBackup',
'E:\backup\FullBackup.dat'
```

Full Backup, z.B. jeden Sonntag, 02:00 Uhr früh

```
backup database Kundenverwaltung to
KundenBackup
```

Wir simulieren die Weiterbearbeitung der Datenbank

```
insert tKunden (Vorname, Nachname, Zeit)
values ('Markus', 'Meller', getdate())
```

Transaktionsprotokoll-Backup, z.B. täglich 09:00, 12:00, 15:00

```
backup log Kundenverwaltung to KundenBackup
```

Nach diesem Vorgang wird weitergearbeitet

```
insert tKunden (Vorname, Nachname, Zeit)
values ('Anton', 'Postl', getdate())
```

Wir simulieren Absturz, mdf ist beschädigt

```
select databaseproperty('Kundenverwaltung', 'IsShutDown')
```

Ergebnis 1 bedeutet, dass Datenbank nicht zur Verfügung steht

Wiederherstellungsvorgang

Schritt 1: restliches Transaktionsprotokoll sichern

```
backup log Kundenverwaltung to KundenBackup with NO_TRUNCATE
```

Schritt 2: beschädigte Datenbank im Management Studio löschen (inkl. Dateien)

Schritt 3: Sicherungsmedium analysieren

```
restore headeronly from KundenBackup
```

Schritt 4: Einspielen Full Backup

```
restore database Kundenverwaltung from KundenBackup with file = 1,
norecovery
```

Schritt 5: Einspielen 1. Transaktionslog-Backup

```
restore log Kundenverwaltung from KundenBackup with file = 2, norecovery
```

Schritt 6: Einspielen letztes Transaktionslog-Backup

```
restore log Kundenverwaltung from KundenBackup with file = 3, recovery
```

Zusammenfassung

1. Datenbank komplett löschen
2. Aktuellstes Datenbank Vollbackup wiederherstellen mit der Option: "Datenbank nicht weiter ausführen. Zusätzliche Transaktionsprotokolle können wiederhergestellt werden"
3. Erstes Transaktionsprotokoll wiederherstellen mit Option: "Datenbank nicht weiter ausführen. Zusätzliche Transaktionsprotokolle können wiederhergestellt werden"
4. Letztes Transaktionsprotokoll wiederherstellen mit Option: "Datenbank weiter ausführen. Zusätzliche Transaktionsprotokolle können nicht wiederhergestellt werden"

```
RESTORE HEADERONLY FROM TestRestore_Backup
DECLARE @File int
DECLARE @FileBegin int
DECLARE @FileEnd int
SET @FileBegin = 2
SET @FileEnd = 3
RESTORE DATABASE TestRestore FROM TestRestore_Backup WITH FILE = 1,
NORECOVERY
SET @File = @FileBegin
WHILE @File <= @FileEnd - 1
BEGIN
    RESTORE LOG TestRestore FROM TestRestore_Backup WITH FILE = @File,
    NORECOVERY
    SET @File = @File + 1
END
RESTORE LOG TestRestore FROM TestRestore_Backup WITH FILE = @FileEnd
Ergebnis "Restore HEADERONLY"
```

```
use master
backup database AdventureWorks to BackupDev
RESTORE HEADERONLY
FROM DISK = N'C:\Backups\BackupDev.bak'
WITH NOUNLOAD;
GO
```

BackupName	BackupDescription	BackupType	ExpirationDate	Compressed	Position	DeviceType
NULL	NULL	1	NULL	0	1	2

UserName	ServerName	DatabaseName	DatabaseVersion	DatabaseCreationDate	BackupSize	FirstLSN	LastLSN
1	MIAMI\Administrator	MIAMI	AdventureWorks	611	2006-10-24 10:53:44.000	172044800	38000000064000070

CheckpointLSN	DatabaseBackupLSN	BackupStartDate	BackupFinishDate	SortOrder	CodePage	UnicodeLocaleId	UnicodeComparisonStyle
38000000064000070	0	2006-10-24 11:18:06.000	2006-10-24 11:18:21.000	52	0	1033	196609

CompatibilityLevel	SoftwareVendorId	SoftwareVersionMajor	SoftwareVersionMinor	SoftwareVersionBuild	MachineName	Flags
90	4508	9	0	1399	MIAMI	512

BindingID	RecoveryForkID	Collation
59568629-86DF-4000-BACD-52194A773D3B	B935AAC8-BB1A-4C10-AD0B-014DFEF2FC72	SQL_Latin1_General_CP1_CI_AS

FamilyGUID	HasBulkLoggedData	IsSnapshot	IsReadOnly	IsSingleUser	HasBackupChecksums	IsDamaged	BeginLogChain
B935AAC8-BB1A-4C10-AD0B-014DFEF2FC72	0	0	0	0	0	0	0

HasIncompleteMetaData	IsForceOffline	IsCopyOnly	FirstRecoveryForkID	ForkPointLSN	RecoveryModel	DifferentialBaseLSN
0	0	0	B935AAC8-BB1A-4C10-AD0B-014DFEF2FC72	NULL	SIMPLE	NULL

DifferentialBaseGUID	BackupTypeDescription	BackupSetGUID
NULL	Database	9F2CE04F-25DF-4F2D-9486-5F1A51EFC34E

Column name	Data type	Description for SQL Server backup sets	Description for other backup sets
BackupName	nvarchar(128)	Backup set name.	Data set name
BackupDescription	nvarchar(255)	Backup set description.	Data set description
BackupType	smallint	Backup type: 1 = Database 2 = Transaction log 4 = File 5 = Full differential 6 = File differential 7 = Partial 8 = Partial differential	Backup type: 1 = Normal 5 = Differential 16 = Incremental 17 = Daily
ExpirationDate	datetime	Expiration date for the backup set.	NULL
Compressed	tinyint	0 = No. SQL Server does not support software compression.	Whether the backup set is compressed using software-based compression: 1 = Yes 0 = No
Position	smallint	Position of the backup set in the volume (for use with the FILE = option).	Position of the backup set in the volume
DeviceType	tinyint	Number corresponding to the device used for the backup operation. Disk: 2 = Logical 102 = Physical Tape: 5 = Logical 105 = Physical Virtual Device: 7 = Logical 107 = Physical Logical device names and device numbers are in sys.backup_devices.	NULL
UserName	nvarchar(128)	Username that performed the backup operation.	Username that performed the backup operation
ServerName	nvarchar(128)	Name of the server that wrote the backup set.	NULL
DatabaseName	nvarchar(128)	Name of the database that was backed up.	NULL
DatabaseVersion	int	Version of the database from which the backup was created.	NULL
DatabaseCreationDate	datetime	Date and time the database was created.	NULL
BackupSize	numeric(20,0)	Size of the backup, in bytes.	NULL
FirstLSN	numeric(25,0)	Log sequence number of the first log record in the backup set.	NULL
LastLSN	numeric(25,0)	Log sequence number of the next log record after the backup set.	NULL
CheckpointLSN	numeric(25,0)	Log sequence number of the most recent checkpoint at the time the backup was created.	NULL
DatabaseBackupLSN	numeric(25,0)	Log sequence number of the most recent full database backup.	NULL
DatabaseBackupStart	datetime	Date and time that the backup operation began.	Write Date
DatabaseBackupFinish	datetime	Date and time that the backup operation finished.	Write Date
SortOrder	smallint	Server sort order. This column is NULL valid for database backups only. Provided for backward compatibility.	NULL
CodePage	smallint	Server code page or character set used by the server.	NULL

UnicodeLocaleId	int	Server Unicode locale ID configuration option used for Unicode character data sorting. Provided for backward compatibility.	IsDamaged	bit	1 = Database was damaged when backed up, but the backup operation was requested to continue despite errors.
UnicodeComparisonStyle	int	Server Unicode comparison style configuration option, which provides additional control over the sorting of Unicode data. Provided for backward compatibility.	BeginLogChain	bit	1 = This is the first in a continuous chain of log backups. A log chain begins with the first log backup taken after the database is created or when it is switched from the Simple to the Full or Bulk-Logged Recovery Model.
CompatibilityLevel	tinyint	Compatibility level setting of the database from which the backup was created.	HasIncompleteMetaData	bit	1 = A tail-log backup with incomplete meta-data. For information about tail-log backups with incomplete backup metadata, see Tail-Log Backups.
SoftwareVendorId	int	Software vendor identification number. For SQL Server, this number is 4608 (or hexadecimal number 0x1200).	IsForceOffline	bit	1 = Backup taken with NORECOVERY; the database was taken offline by backup.
SoftwareVersionMajor	int	Major version number of the server that created the backup set.	IsCopyOnly	bit	1 = A copy-only backup. A copy-only backup does not impact the overall backup and restore procedures for the database. For more information, see Copy-Only Backups (Simple Recovery Model) or Copy-Only Backups (Full Recovery Model).
SoftwareVersionMinor	int	Minor version number of the server that created the backup set.			
SoftwareVersionBuild	int	Build number of the server that created the backup set.			
MachineName	nvarchar(128)	Name of the computer that performed the backup operation.	FirstRecoveryForkID	uniqueidentifier	ID for the starting recovery fork. This column corresponds to first_recovery_fork_guid in the backupset table. For data backups, FirstRecoveryForkID equals RecoveryForkID.
Flags	int	Individual flags bit meanings if set to 1: 1 = Log backup contains bulk-logged operations. 2 = Snapshot backup. 4 = Database was read-only when backed up. 8 = Database was in single-user mode when backed up. 16 = Backup contains backup checksums. 32 = Database was damaged when backed up, but the backup operation was requested to continue despite errors. 64 = Tail log backup. 128 = Tail log backup with incomplete metadata. 256 = Tail log backup with NORECOVERY. Important: We recommend that instead of Flags you use the individual Boolean columns (listed below starting with HasBulkLoggedData and ending with IsCopyOnly).	ForkPointLSN	numeric(25,0) NULL	If FirstRecoveryForkID is not equal to RecoveryForkID, this is the log sequence number of the fork point. Otherwise, this value is NULL.
			RecoveryModel	nvarchar(60)	Recovery model for the Database, one of: FULL BULK-LOGGED SIMPLE
			DifferentialBaseLSN	numeric(25,0) NULL	For a single-based differential backup, the value equals the FirstLSN of the base backup; changes with LSNs greater than or equal to DifferentialBaseLSN are included in the differential. For a multi-based differential, the value is NULL, and the base LSN must be determined at the file level). For non-differential backup types, the value is always NULL. For more information, see The Base of a Differential Backup.
BindingID	uniqueidentifier	Binding ID for the database. This corresponds to sys.databases.database_guid. When a database is restored, a new value is assigned. Also see FamilyGUID (below).	DifferentialBaseGUID	uniqueidentifier	For a single-based differential backup, the value is the unique identifier of the base backup. For multi-based differentials, the value is NULL, and the differential base must be determined per file. For non-differential backup types, the value is NULL.
RecoveryForkID	uniqueidentifier	ID for the ending recovery fork. This column corresponds to last_recovery_fork_guid in the backupset table. For data backups, RecoveryForkID equals FirstRecoveryForkID.	BackupTypeDescription	nvarchar(60)	Backup type as string, one of: DATABASE TRANSACTION LOG FILE OR FILEGROUP DATABASE DIFFERENTIAL FILE DIFFERENTIAL PARTIAL PARTIAL DIFFERENTIAL INCREMENTAL DAILY
Collation	nvarchar(128)	Collation used by the database.	BackupSetGUID	uniqueidentifier NULL	Unique identification number of the backup set, by which it is identified on the media.
FamilyGUID	uniqueidentifier	ID of the original database when created. This value stays the same when the database is restored.			
HasBulkLoggedData	bit	1 = Log backup containing bulk-logged operations.			
IsSnapshot	bit	1 = Snapshot backup.			
IsReadOnly	bit	1 = Database was read-only when backed up.			
IsSingleUser	bit	1 = Database was single-user when backed up.			
HasBackupChecksums	bit	1 = Backup contains backup checksums.			

Die Kapitel 10-14 befinden sich im Anhang zu dieser Ausgabe. Der Anhang ist eine PDF-Datei, die bei der Webversion dieses Artikels downgeloadet werden kann. <http://pcnews.at/?id=pcn110>
Direkter Link: http://pcnews.at/_pdf/n1101001.pdf
Details zur Installations des SQL-Servers 2008 finden sich ebenfalls in einem Anhang als PDF-Datei: http://pcnews.at/_pdf/n1101011.pdf