WSW Software QUMBU

Ganz nach Plan

von Sandro Lucifora

Datenbanken sind das technische Herz von Unternehmen. Wenn diese nicht mehr funktionieren, steht meist der komplette Arbeitsprozess still. Daher ist es wichtig, sie stetig zu pflegen und zu sichern. Wie sich das mit der Software QUMBU bewerkstelligen lässt, haben wir uns angesehen und festgestellt, dass sich die Entwickler viele Gedanken um die einfache Bedienung und ein planvolles Vorgehen gemacht haben.

Der deutsche Anbieter WSW Software hat mit QUMBU eine neue Software für das Erstellen von Backups und die Wartung des SQL-Servers entwickelt. An der Aussage, dass sich diese Vorgänge mühelos und sicher organisieren lassen, musste sich die Software in unserem Test messen lassen.

Die Installation nahmen wir auf einem Windows Server 2019 vor, wobei das Tool ab Windows 7 oder Windows Server 2012 mit einem .NET-Framework läuft. Als SQL-Server setzen die Entwickler von QUMBU die Version 2008 R2 oder neuer voraus. Wir hatten einen SQL-Server 2019 installiert. Die deutsche, gut strukturierte Anleitung hilft auch den Administratoren durch die Installation und die Einarbeitung, die nicht täglich mit SQL-Servern arbeiten. Der Setupassistent erlaubt die Auswahl zur Installation eines einzelnen QUMBU-Clients, des QUMBU-Servers oder beider Komponenten zusammen. Daraus ergibt sich schon, dass sich die Serverkomponente und der Client getrennt aufspielen lassen, zum Beispiel auf dem Server und einem administrativen Desktoprechner. Wir wählten die vollständige Installation aus. Der QUMBU-Server läuft im Hintergrund als Windows-Dienst, dessen Ausführung einen angemeldeten Benutzer erfordert. Dieses Benutzerkonto gaben wir im Laufe der Installation an. Es ist dann das Konto, das der Dienst zur Ausführung verwendet.

Danach erwartet der Assistent die Eingabe eines QUMBU-Accounts. Das ist der Benutzer, der Zugriff auf alle SQL-Server-Instanzen hat. Das kann auch ein User aus der lokalen Windows-Authentifizierung sein, wodurch ein Domänencontroller keine Voraussetzung ist. Der Einfachheit halber konnten wir uns alle möglichen Benutzer anzeigen lassen und den gewünschten aus einer Liste auswählen. Nach der Eingabe des Kennworts schlossen wir die Installation ab. Es ist jedoch zu beachten, dass die Installation nur mit Administratoren-Rechten erfolgen kann, da sich sonst die notwendigen Dienste nicht starten lassen. Während der Installation auf Windows Server 2019 hatten wir zunächst Probleme, da der QUMBU-Dienst nicht starten wollte. Der Support des Herstellers analysierte daraufhin das Problem und stellte fest, dass – entgegen der anderslautenden Systemvoraussetzung – noch das .NET-Framework 3.5 notwendig war. Wir installierten dieses nach und schlossen das Setup erfolgreich ab.

Keine Verbindung zur Azure-Datenbank

Nach dem Start öffnete sich ein aufgeräumtes und übersichtliches Fenster, das QUMBU-Cockpit. Neben der Symbolleiste gibt es einen Bereich für die Anzeige der Server, für die Aufgaben sowie die Aufgabendetails und Benachrichtigungen. Wir begannen damit, zunächst einen Microsoft-SQL-Server hinzuzufügen. Das erfolgte durch einen Rechtsklick auf den Bereich "Server" und die Auswahl des Eintrags "Mit Server" verbinden. Im folgenden Dialog gaben wir den Servernamen und unsere Logindaten ein.

Nach der Verbindung sahen wir im Bereich "Server" unser hinzugefügtes Gerät. An dieser Stelle lassen sich auch weitere Server eintragen, die jedoch über das Netzwerk erreichbar sein müssen. Verbindungen über VPN sind genauso denkbar wie öffentlich erreichbare Server. Und hierin liegt auch schon ein Vorteil gegen-

über Microsofts SQL Server Management Studio: Mit QUMBU konnten wir mehrere Server einbinden und verwalten. Jedoch verweigerte QUMBU die Verbindung zu einer SQL-Datenbank, die wir in Azure betrieben, mit dem Hinweis "Dieser Server läuft nicht auf einem Windows-Betriebssystem". Demnach ist es Voraussetzung, dass wir den SQL-Server auch auf einem dedizierten Windows-Server betreiben, damit QUMBU damit umgehen kann. Das kann wiederum ein physischer wie auch virtueller Server sein.

Damit einhergeht auch das Lizenzmodell. Der Hersteller setzt lediglich pro physischem beziehungsweise virtuellem Server eine Lizenz voraus. Diese gilt dann für die Verwaltung beliebig vieler Datenbankinstanzen und Benutzer. Über das Kontextmenü bekamen wir die Eigenschaften der Serverinstanz und Datenbank angezeigt. Damit erfuhren wir beispielsweise die Serverversion und ob der zur Verbindung verwendete Benutzer alle Rechte besitzt. Für die jeweilige Datenbank zeigte uns QUMBU den aktuellen Status sowie die Daten der Erzeugung und der letzten Sicherung an.

Serverübergreifende Tasks

War der Server eingerichtet, widmeten wir uns dem zentralen Bereich der Aufgaben. Je nachdem, welcher Server im entsprechenden Bereich ausgewählt war, nahmen wir für diesen die Konfiguration vor. So konnten wir auch in der obersten Ebene der Baumstruktur "Alle Server" markieren, um damit die nachfolgend definierte Aufgabe auf alle Server anzuwenden. Der Bereich "Aufgabendetails" zeigt die Historie der ausgeführten Aufgaben tabellarisch an. Hier konnten wir über einen Filter bestimmte Server oder Aufgaben eingrenzen. Ohne diese Einschränkung zeigte uns QUMBU sämtliche Aufgaben aller SQL-Server an. Im unteren Fenster sahen wir aktuelle Benachrichtigungen zum Zustand des Systems. Diese betrafen einzelne Aufgaben oder einzelne SQL-Server-Instanzen. Gut gefallen hat uns, dass wir hier auch die Dringlichkeit erkennen konnten – das heißt, ob die Nachricht eine reine Mitteilung oder eine Warnung war, auf die wir reagieren sollten.

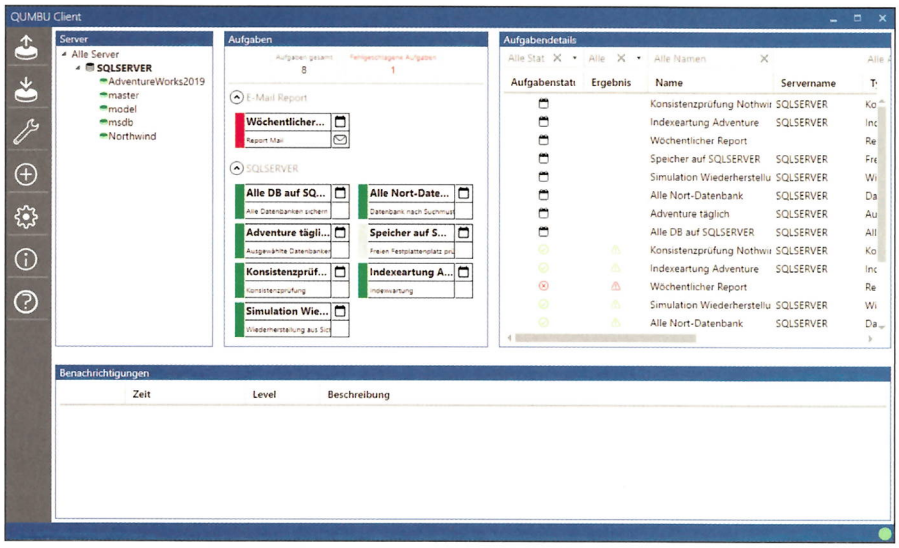

Bild 1: Der QUMBU-Client hat eine moderne und übersichtliche Oberfläche.

Alles über Aufgaben geregelt

Egal, was wir mit QUMBU machen wollten, die Entwickler haben alle Funktionen recht einfach in Aufgaben verpackt. Sowohl die Sicherung als auch die Wiederherstellung einer Datenbank sind genau so Aufgaben wie die verschiedenen Wartungsfunktionen.

Wir legten als Erstes eine Aufgabe zur Datenbanksicherung an, indem wir auf der Funktionsleiste den Eintrag "Sicherung" wählten. Danach gab es die Optionen, alle oder nur ausgewählte Datenbanken sowie eine Datenbank entsprechend eines Suchmusters zu sichern. Ein Klick auf den jeweiligen Eintrag brachte uns in die korrespondierende Aufgabenkonfiguration. Die Bedienung war immer gleich: Wir öffneten das Kontextmenü mit der rechten Maustaste und wählten "Hinzufügen" aus. Der Konfigurationsdialog fragte der Aufgabe entsprechende Parameter ab. So reichte es, für das Sichern aller Datenbanken der Aufgabe einen Namen zu geben und den zu sichernden SQL-Server auszuwählen.

Auf der nächsten Seite des Dialoges bestimmten wir den Pfad, Dateinamen und die Beschreibung. Dabei stellte uns QUMBU auch Variablen für den Servernamen, die Datenbank und das Sicherungsdatum zur Verfügung. Im Reiter "Experte" gab es noch eine Vielfalt von Optionen bezüglich der Sicherung selbst. So legten wir die Komprimierung fest und bestimmten, ob es eine differenzielle Sicherung oder nur eine reine Kopie sein sollte. Die optionale Verschlüsselung gab uns noch eine weitere Stufe zum Schutz unserer Daten. Weitere Optionen waren noch die Anzahl der Puffer, die maximale Transferrate und die Einstellung der Blockgröße.

WSW Software QUMBU

Produkt
Software für das Backup und die Wartung von Microsoft SQL Server.

Hersteller
WSW Software GmbH
www.qumbu.de

Preis
Laufzeit 12 Monate:
59 Euro pro Monat und Server.
Laufzeit 36 Monate:
39 Euro pro Monat und Server.

Systemanforderungen
- Unterstützte Betriebssysteme: Ab Windows 7 mit .NET-Framework 4.6.2; Windows Server 2012, 2016 oder 2019
- Prozessor: 2 GHz oder schneller
- Arbeitsspeicher: Mindestens 4 GByte RAM
- Festplattenplatz: Mindestens 1 GByte freier Festplattenplatz
- Grafikauflösung: Mindestens 1024 x 768
- SQL-Server-Versionen: Microsoft SQL-Server ab Version 2008 R2 unter Windows
- Internetzugang zur Lizenzierung des Programms

Technische Daten
www.it-administrator.de/downloads/datenblaetter

Bild 2: Bei den Angaben für den Pfad und Dateinamen einer Sicherung lassen sich auch Variablen verwenden.

Benachrichtigung nur per E-Mail

Um die E-Mail-Einstellungen zur Benachrichtigung des Sicherungsstatus vorzunehmen, mussten wir zunächst in den E-Mail-Einstellungen einen SMTP-Server festlegen, mit dem QUMBU seine E-Mails versendet. Danach gaben wir noch Parameter zur Benachrichtigung der Sicherung an. Hier bestimmten wir auch, bei welchen Ereignissen wir eine Mitteilung erhalten wollten. Was uns an der Stelle fehlte, waren alternative Benachrichtigungswege wie über einen Webhook, über den wir zum Beispiel Slack hätten mit einbinden können.

Im letzten Fenster definierten wir den Zeitplan der Sicherung. Neben der Auswahl, ob täglich, wöchentlich oder monatlich, hatten wir noch Platz für die korrespondierenden Angaben wie Wochentag, Uhrzeit sowie Start- und optional Enddatum. Die Konfiguration für die Sicherung einzelner Datenbanken war komplett identisch bis auf die Möglichkeit, am Anfang statt nur den SQL-Server auch eine oder mehrere Datenbanken auszuwählen.

Eine sehr hilfreiche Funktion ist auch die Sicherung nach Mustern. Darunter verstehen die Entwickler die Durchführung der Aktion für alle Datenbanken, die einem bestimmten Filter entsprechen. Das kann hilfreich sein, wenn zum Beispiel alle Datenbanken für den produktiven Betrieb mit "Prod" beginnen und die für eine Weiterentwicklung mit "Dev". Die Dev-Datenbank wollen wir nicht täglich sichern, aber die der produktiven Umgebung schon. In diesem Szenario würden wir nach Datenbanknamen filtern, beginnend mit "Prod". Uns erscheint das ein sehr gutes Werkzeug, vor allem wenn Datenbanken dynamisch hinzukommen oder gelöscht werden und sich eine Vielzahl von Datenbanken auf einem Server befindet.

Im Bedarfsfall soll eine gesicherte Datenbank auch wieder auf demselben oder einem anderen Microsoft-SQL-Server wiederhergestellt werden. Dazu gibt es in QUMBU die Aufgabe "Wiederherstellung". In den Optionen haben die Entwickler die Wiederherstellung aus der Sicherungshistorie, die nur auf demselben Server möglich ist, als auch die Wiederherstellung aus einer Datei implementiert. Letzteres ermöglicht das Recovery der Datenbank auf einem anderen SQL-Server. Bei beiden Optionen ist der Konfigurationsdialog für die Aufgabe von der Bedienung her genauso einfach aufgebaut wie bei der Sicherung.

Durch einen Rechtsklick und "Hinzufügen" öffnet sich der jeweilige Dialog, in dem wir bei der Wiederherstellung aus der Historie den Server sowie die Datenbank auswählten. Im nächsten Fenster zeigt uns das Programm die Historie an, aus der wir den gewünschten wiederherzustellenden Zeitpunkt auswählten. Danach ermöglichten uns die Experten- und E-Mail-Einstellungen weitere Anpassungen. Über den Zeitplan legten wir fest, wann die Wiederherstellung erfolgen sollte. Auch tägliche, wöchentliche und monatliche Intervalle waren möglich.

Bei der Wiederherstellung aus einer Datei unterschied sich die Festlegung der Aufgabe lediglich im ersten Dialog, über den wir die Sicherungsdatei auswählten und den SQL-Server und die Zieldatenbank festlegten, die für die Rücksicherung gewünscht waren. Der restliche Konfigurationsprozess glich exakt dem der Wieder-

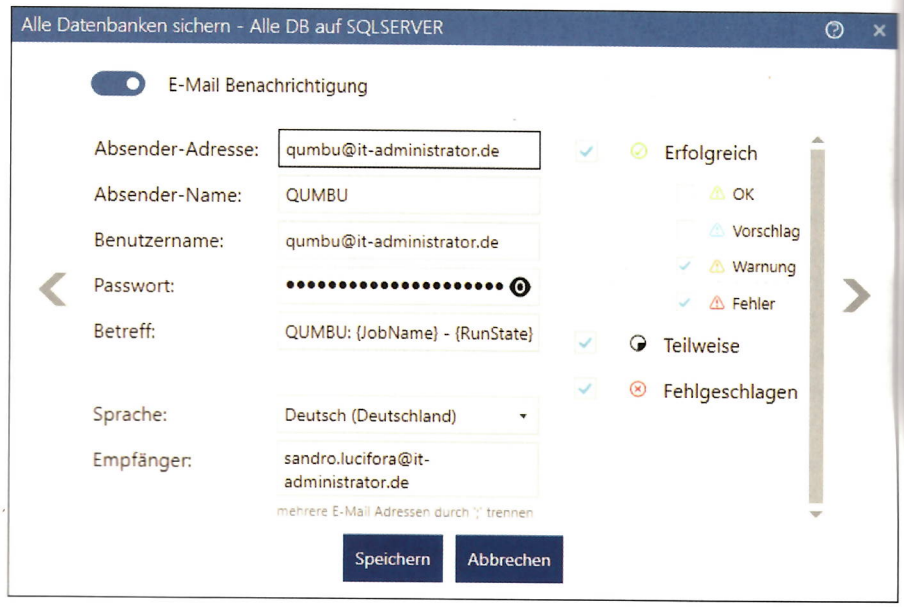

Bild 3: Die E-Mail-Benachrichtigung informiert über Erfolg oder Misserfolg einer Aufgabe.

herstellung aus der Historie. Um eine Datenbank von einem Server zum anderen zu klonen, verwendet der Hersteller die zwei nacheinander ablaufenden Aufgaben "Sicherung" und "Wiederherstellung". Ein Anwendungsbeispiel hierfür ist, regelmäßig eine Datenbank auf einen anderen Server zu duplizieren.

Einfache Wiederherstellung

Zum Beispiel für die Überprüfung der Konsistenz einer erfolgten Sicherung hat WSW die Option eingebaut, die Wiederherstellung nur zu simulieren. Dabei liest und verarbeitet QUMBU die Daten exakt so, als würde die reale Wiederherstellung erfolgen – nur mit dem Unterschied, dass die Software am Ende keine Daten schreibt. Damit konnten wir prüfen, ob die Sicherung von einem anderen Datenbankserver kompatibel zum Zielserver ist, wenn beide SQL-Server unterschiedliche Versionen hatten. Die entsprechenden Aufgaben für die Wiederherstellung der Simulation sind identisch zu den zuvor beschriebenen Prozessen der tatsächlichen Wiederherstellung.

Neben den Hauptaufgaben der Sicherung und Wiederherstellung der Datenbanken haben die Entwickler den Anwendern konsequenterweise auch einige Wartungsaufgaben mit an die Hand gegeben. Unter dem gleichnamigen Menüpunkt standen uns die Konsistenzprüfung, die Indexwartung und das Ermitteln unbenutzter Indizes als auch die Prüfung auf noch freie Festplattenkapazität zur Auswahl.

Auch hier hat uns sehr gut gefallen, dass das Anlegen der Aufgaben identisch zur vorher beschriebenen Arbeitsweise ablief. Über das Kontextmenü fügten wir eine neue Aufgabe hinzu und nahmen die aufgabenspezifischen Einstellungen vor. Für die Konsistenzprüfung wählten wir den Server und die Datenbank aus. Über die Experteneinstellungen bestimmten wir das DBCC-Kommando, das wir für die Prüfung ausgeführt haben wollten. Neben der bekannten E-Mail-Benachrichtigung legten wir noch den Zeitplan zur Prüfung fest. In diesem Fall entschieden wir uns für eine wöchentliche Prüfung an einem Sonntagmorgen.

Für die regelmäßige Indexwartung legten wir wieder eine Aufgabe an, für die wir den Server und die Datenbank auswählten. In den Experteneinstellungen konnten wir dieses Mal die Schwellenwerte für die Reorganisation und den Neuaufbau in Prozent festlegen. Auch für die Mindestanzahl der Indexseiten haben die Entwickler eine Option eingebaut. Laut Hersteller ergibt die Indexwartung aber erst nach sieben Tagen Laufzeit einen Sinn, da sich vorher keine belastbaren Aussagen zu veralteten oder unbenutzten Indizes treffen lassen. Doch im Bedarfsfall kann der Administrator diese Zeit in der Konfiguration anpassen.

Um sicherzugehen, dass der Datenbankserver auch über genügend Speicherplatz verfügte, richteten wir die entsprechende Prüfung ein. Im Konfigurationsdialog wählten wir den SQL-Server aus und konnten aus der angezeigten Liste der verfügbaren Laufwerke diejenigen auswählen, die wir überwachen lassen wollten. Die Einstellungen der Grenzwerte in GByte für

IT Administrator Intensiv-Seminar

PowerShell für Admins

Quelle: goodluz – 123RF.com

3. bis 5. Mai 2021

Online-Intensiv-Seminar

Anmeldung unter:
www.it-administrator.de/trainings

Unser Angebot:

- Maximal 8 Teilnehmer je Seminar
- Das ONLINE-Intensiv-Seminar startet am ersten Tag um 9:30 Uhr und endet am dritten Tag gegen 15:30 Uhr.

WICHTIG! Für eine erfolgreiche Teilnahme gelten folgende Voraussetzungen: Der Teilnehmer muss über zwei Bildschirme (alternativ einen modernen Breitbildschirm), Webcam und Headset verfügen.

Seminarpreise:

Sonderpreis für Abonnenten des IT-Administrator:
975,80 Euro
(inkl. 19% MwSt.)

für Nicht-Abonnenten:
1.130,50 Euro
(inkl. 19% MwSt.)

Die Teilnahmegebühr umfasst das Intensiv-Seminar inklusive Dokumentation.

Ihr Dozent:

Franz-Georg Clodt

Agenda:

- **Einführung und Basiswissen**
  Architektur der PowerShell und Versionsunterschiede / Cmdlets, Cmdlet-Parameter und Hilfefunktionen / Objekt-Pipeline und Ausgabefunktionen / Navigationsmodell (PowerShell-Provider)
- **Scripting**
  PowerShell Language (PSL): Variablen und Kontrollstrukturen / Objektorientiertes Programmieren mit Klassen (ab PowerShell 5.0) / Sicherheitsfunktionen (Execution Policy) / Vordefinierte Variablen, Fehlerbehandlung und Fehlersuche / Scripting mit PowerShell Integrated Scripting Environment (ISE)
- **Aufbauwissen**
  Fernaufruf/Fernadministration mit WS-Management (Remoting) / Serververwaltung Active Directory und Automatisierung von Azure

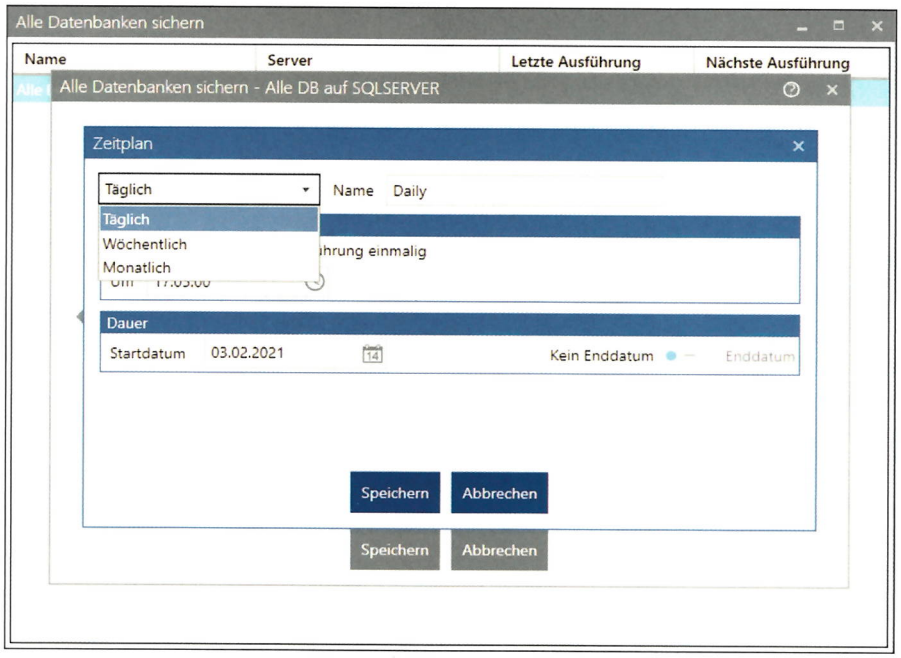

Bild 4: Die Konfiguration des Zeitplans ist für jede Aufgabe identisch und deckt alle sinnvollen Möglichkeiten ab.

eine Warnung als auch für die Benachrichtigung als Fehler nahmen wir individuell pro Laufwerk vor.

Die anschließende Konfiguration für die E-Mail-Benachrichtigung wie auch das Anlegen des auszuführenden Zeitplanes waren wohlbekannt. Für diese Aufgabe wählten wir jedoch nicht die einmalige Ausführung pro Tag, Wochentag oder innerhalb eines Monats aus, sondern legten hier eine tägliche Prüfung in einem vorgegebenen Zeitrahmen und einem bestimmten Intervall fest. Im Test war das die wochentägliche Prüfung, ohne Wochenende, in der Produktionszeit von 8 bis 22 Uhr mit einem Intervall von 15 Minuten. Damit wollten wir sicherstellen, dass es gerade zu den Stoßzeiten, in denen unsere Datenbank benutzt wird, keinen Ausfall mangels Festplatten-Speicherplatz geben kann.

Kein Benutzermanagement

Erwähnenswert ist noch, dass es in QUMBU kein Benutzermanagement im Bezug auf die Bedienung der Software gibt. Die Zielgruppe der Anwender ist insofern begrenzt, als dass es lediglich den Administrator gibt, der alles darf, oder Benutzer, die nur Read-Only-Rechte haben. Diese Einstellungen zeigten sich auch in der Liste der Benutzer, unterhalb der Funktion "Einstellungen". Hier gab QUMBU alle Benutzer aus, die wir für den Zugriff auf die Datenbanken hinterlegt hatten. Eine zusätzliche Checkbox "nur Lesen" begrenzte deren Möglichkeiten.

Um die Funktionen noch abzurunden, hat der Hersteller, ebenfalls als Aufgabe, das Versenden von Berichten integriert. Uns war es so möglich, zum Beispiel einen wöchentlichen Report per E-Mail zu erhalten, der die Ergebnisse der definierten Aufgaben übersichtlich darstellte. Neben der Auswahl des Servers und der möglichen Aufgaben stellten wir dafür noch den Reporting-Zeitraum ein.

Fazit

Die Kernaufgaben der Sicherung und Wiederherstellung von SQL-Datenbanken hat die Software vorbildlich gemeistert. Auch die zusätzlichen Wartungsaufgaben erlauben es selbst weniger versierten Datenbank-Administratoren, prophylaktische Prüfungen durchzuführen. Das Herunterbrechen aller Funktionen in eine Aufgabe sowie das durchgehende, gut verständliche Benutzerinterface vereinfachen die Integration in eine bestehende Infrastruktur.

Im Gegensatz zum Microsoft SQL Server Management Studio lassen sich mit QUMBU auch mehrere Microsoft SQL Server verwalten und die Aufgaben auch serverübergreifend identisch konfigurieren. Funktional haben die Entwickler bei der Konfiguration der Aufgaben darauf geachtet, dass sich die Dialoge nur da unterscheiden, wo sie funktionsspezifische Eingaben erwarten. Diese sind nicht überladen und die Konfigurationsmöglichkeiten sind sehr durchdacht.

Zum zeitgesteuerten Klonen von Datenbanken setzen die Entwickler auf eine Sicherung mit anschließender Wiederherstellung. Die zusätzlichen Wartungsfunktionen zur Konsistenzprüfung, Indexwartung, Ermitteln unbenutzter Indizes und auch die Festplattenplattenprüfung unterscheiden aus unserer Sicht QUMBU von den meisten anderen Tools dieser Kategorie. Die Wartungen sind zwar nur "Kleinigkeiten", doch helfen Sie, eine Datenbank kontinuierlich zu optimieren.

Nicht zu verschweigen ist, dass es gerade bei der Installation Anlaufschwierigkeiten gab, die jedoch durch den Support schnell und kompetent gelöst wurden. Für uns ist QUMBU rundherum ein gelungenes Tool mit einer durchdachten und modernen Benutzeroberfläche, einfacher Bedienung und vor allem sinnvollen und zuverlässigen Funktionen. *(dr)* IT

So urteilt IT-Administrator

Sicherung von Datenbanken	7
Wiederherstellung von Datenbanken	8
Wartungsfunktionen	8
Konfigurationsmöglichkeiten	7
Benachrichtigungen	6

Die Details unserer Testmethodik finden Sie unter www.it-administrator.de/testmethodik

Dieses Produkt eignet sich

optimal für die Sicherung und Wartung von mehreren Microsoft SQL Servern. Das Werkzeug ist auch für weniger erfahrene Administratoren konzipiert.

bedingt als als reine Wartungssoftware für SQL Server.

nicht als Sicherungslösung für Datenbanken, die nicht auf einem Windows-Betriebssystem laufen, wie Azure oder AWS.