

Prüfungsnummer: 70-462-deutsch

Prüfungsname: Administering
Microsoft SQL Server 2012 Databases

Version: Demo

<http://zertifizierung-portal.de/>

1. Sie sind als Datenbankadministrator für das Unternehmen it-pruefungen.de tätig. Sie administrieren eine Microsoft SQL Server 2012 Datenbank mit dem Namen DB1.

Die TRUSTWORTHY-Datenbankeigenschaft von DB1 ist mit ON festgelegt.

Sie erstellen eine gespeicherte Prozedur, die Informationen der Datenbankebene aus dynamischen Verwaltungssichten abrufen.

Sie erteilen Tom das Recht, die gespeicherte Prozedur auszuführen.

Sie müssen sicherstellen, dass die gespeicherte Prozedur korrekt funktioniert, wenn sie von Tom ausgeführt wird. Sie wollen Tom nur die minimal erforderlichen Berechtigungen erteilen.

Wie gehen Sie vor?

(Jede korrekte Antwort stellt eine vollständige Lösung dar. Wählen Sie zwei Antworten.)

A. Erstellen Sie eine SQL Server Anmeldung mit der Berechtigung VIEW SERVER STATE und eine Anwendungsrolle mit einem sicheren Kennwort.

B. Ändern Sie die gespeicherte Prozedur und schließen Sie die Anweisung EXECUTE AS OWNER ein. Erteilen Sie dem Besitzer der gespeicherten Prozedur die Berechtigung VIEW SERVER STATE.

C. Erstellen Sie eine SQL Server Anmeldung mit der Berechtigung VIEW SERVER STATE. Ändern Sie die gespeicherte Prozedur und schließen Sie die Anweisung EXECUTE AS (newlogin) ein.

D. Weisen Sie Tom die Datenbankrolle db_owner für DB1 zu.

E. Weisen Sie Tom die Serverrolle sysadmin zu.

Korrekte Antwort: B, C

Erläuterungen:

Die TRUSTWORTHY-Datenbankeigenschaft gibt an, ob die SQL Server-Instanz die Datenbank und ihre Inhalte als vertrauenswürdig einstuft. Standardmäßig ist diese Einstellung OFF. Sie kann jedoch mithilfe der ALTER DATABASE-Anweisung auf ON festgelegt werden.

Beispiel:

```
ALTER DATABASE AdventureWorks2012 SET TRUSTWORTHY ON;
```

Dynamische Verwaltungssichten und -funktionen geben Serverstatusinformationen zurück, mit denen der Zustand einer Serverinstanz überwacht, Probleme diagnostiziert und die Leistung optimiert werden kann.

Auf dynamische Verwaltungssichten kann in Transact-SQL-Anweisungen mithilfe zweiteiliger, dreiteiliger oder vierteiliger Namen verwiesen werden. Auf dynamische Verwaltungsfunktionen kann andererseits in Transact-SQL-Anweisungen mithilfe zweiteiliger oder dreiteiliger Namen verwiesen werden. Auf dynamische Verwaltungssichten und -funktionen kann in Transact-SQL-Anweisungen nicht mithilfe einteiliger Namen verwiesen werden.

Alle dynamischen Verwaltungssichten und -funktionen sind im sys-Schema vorhanden und verwenden die Benennungskonvention dm_*. Wenn Sie eine dynamische Verwaltungssicht oder -funktion verwenden, müssen Sie vor dem Namen der Sicht oder

Funktion das sys-Schema als Präfix einfügen.

Zum Abfragen einer dynamischen Verwaltungssicht oder -funktion sind die SELECT-Berechtigung für das Objekt und die VIEW SERVER STATE- oder VIEW DATABASE STATE-Berechtigung erforderlich. Auf diese Weise können Sie den Zugriff eines Benutzers oder eines Anmeldenamens selektiv auf dynamische Verwaltungssichten und -funktionen einschränken. Dazu erstellen Sie zunächst den Benutzer in master und verweigern dem Benutzer dann die SELECT-Berechtigung für die dynamischen Verwaltungssichten oder -funktionen, auf die er keinen Zugriff haben soll. Der Benutzer kann dann diese dynamischen Verwaltungssichten oder -funktionen unabhängig vom Datenbankkontext des Benutzers nicht auswählen.

Eine Übersicht mit Datenbanken verbundener dynamischer Verwaltungssichten erhalten Sie unter dem folgenden Link:

Mit der Datenbank verbundene dynamische Verwaltungssichten (Transact-SQL)

2. Sie sind als Datenbankadministrator für das Unternehmen it-pruefungen.de tätig. Sie administrieren eine Microsoft SQL Server 2012 Instanz. Die Instanz hostet eine Datenbank für die Buchhaltung. Die Dateien der Datenbank sind in einem Storage Area Network (SAN) gespeichert.

Die Datenbank weist die folgenden Merkmale auf:

Die Datendatei mit einer Größe von zwei Terrabyte (TB) befindet sich auf einer dedizierten LUN, die auf dem Server als Laufwerk D: eingebunden ist.

Das Transaktionsprotokoll mit einer Größe von 10 Gigabyte (GB) befindet sich auf einer dedizierten LUN, die auf dem Server als Laufwerk E: eingebunden ist.

Auf Laufwerk D: ist ein Terrabyte freier Speicher verfügbar.

Auf Laufwerk E: sind fünf Gigabyte freier Speicher verfügbar.

Während der Geschäftszeiten werden fortlaufend Änderungen am Datenbestand der Datenbank durchgeführt. Die Geschäftszeiten sind von Montags bis Freitags jeweils von 09:00 Uhr bis 17:00 Uhr. Täglich werden ca. fünf Prozent der Daten geändert.

Die Buchhaltung lädt täglich zwischen 11:15 Uhr und 15:15 Uhr große CSV-Dateien in die Datenbank. Für die Aktion wird entweder das Hilfsprogramm BCP oder die Transact-SQL Anweisung BULK INSERT verwendet. Bei jedem Import werden der Datenbank ca. 3 GB Daten hinzugefügt. Der Datenimport muss in möglichst kurzer Zeit abgeschlossen werden.

Jeden Sonntag um 10:00 Uhr wird eine vollständige Sicherung der Datenbank durchgeführt. Zusätzliche Sicherungsvorgänge werden während der Geschäftszeiten im zwei-Stunden-Rhythmus (11:00, 13:00, 15:00 und 17:00) durchgeführt.

Sie müssen sicherstellen, dass die Sicherung auch beim Auftreten einer ungültigen Prüfsumme fortgesetzt wird.

Welche Option verwenden Sie?

(Diese Aufgabe gehört zu einer Reihe von Fragestellungen, für die dieselben Antworten zur Auswahl stehen. Im Hilfetext finden Sie erweiterte Antwortmöglichkeiten.)

A.STANDBY

- B.Differenziell
- C.Vollständig
- D.CHECKSUM
- E.Massenprotokolliert
- F.CONTINUE_AFTER_ERROR

Korrekte Antwort: F

Erläuterungen:

Mit der Option CONTINUE_AFTER_ERROR können Sie sowohl bei der Sicherung als auch bei der Wiederherstellung einer Datenbank angeben, ob der Vorgang beim Auftreten eines Fehlers angehalten oder fortgesetzt wird.

So bestimmen Sie, ob ein Sicherungsvorgang fortgesetzt oder angehalten wird, nachdem ein Fehler festgestellt wurde:

Stellen Sie eine Verbindung mit dem Datenbankmodul her.

Klicken Sie in der Standardleiste auf Neue Abfrage.

Geben Sie in der BACKUP-Anweisung die CONTINUE_AFTER_ERROR-Option an, um fortzufahren, oder geben Sie die STOP_ON_ERROR-Option an, um den Vorgang anzuhalten. Das Standardverhalten besteht, den Vorgang anzuhalten, wenn ein Fehler gefunden wird. In diesem Beispiel wird der Sicherungsvorgang angewiesen, den Vorgang fortzusetzen, obwohl ein Fehler gefunden wurde.

Beispiel:

```
BACKUP DATABASE AdventureWorks2012  
TO DISK = 'Z:\SQLServerBackups\AdvWorksData.bak'  
WITH CHECKSUM, CONTINUE_AFTER_ERROR;  
GO
```

Erweiterte Antwortmöglichkeiten:

Hätten Sie es auch gewußt, wenn mehr als die gezeigten 6 Antwortmöglichkeiten zur Auswahl stünden?

- A: Massenprotokolliert
- B: CONTINUE_AFTER_ERROR
- C: Einfach
- D: DBO_ONLY
- E: COPY_ONLY
- F: SKIP
- G: RESTART
- H: Transaktionsprotokoll
- I: NO_CHECKSUM
- J: NORECOVERY
- K: Differenziell
- L: STANDBY
- M: CHECKSUM
- N: Vollständig

3. Sie sind als Datenbankadministrator für das Unternehmen it-pruefungen.de tätig. Sie administrieren eine Microsoft SQL Server 2012 Instanz. Die Instanz hostet eine Datenbank für die Buchhaltung. Die Dateien der Datenbank sind in einem Storage Area Network (SAN) gespeichert.

Die Datenbank weist die folgenden Merkmale auf:

Die Datendatei mit einer Größe von zwei Terrabyte (TB) befindet sich auf einer dedizierten LUN, die auf dem Server als Laufwerk D: eingebunden ist.

Das Transaktionsprotokoll mit einer Größe von 10 Gigabyte (GB) befindet sich auf einer dedizierten LUN, die auf dem Server als Laufwerk E: eingebunden ist.

Auf Laufwerk D: ist ein Terrabyte freier Speicher verfügbar.

Auf Laufwerk E: sind fünf Gigabyte freier Speicher verfügbar.

Während der Geschäftszeiten werden fortlaufend Änderungen am Datenbestand der Datenbank durchgeführt. Die Geschäftszeiten sind von Montags bis Freitags jeweils von 09:00 Uhr bis 17:00 Uhr. Täglich werden ca. fünf Prozent der Daten geändert.

Die Buchhaltung lädt täglich zwischen 11:15 Uhr und 15:15 Uhr große CSV-Dateien in die Datenbank. Für die Aktion wird entweder das Hilfsprogramm BCP oder die Transact-SQL Anweisung BULK INSERT verwendet. Bei jedem Import werden der Datenbank ca. 3 GB Daten hinzugefügt. Der Datenimport muss in möglichst kurzer Zeit abgeschlossen werden.

Jeden Sonntag um 10:00 Uhr wird eine vollständige Sicherung der Datenbank durchgeführt. Zusätzliche Sicherungsvorgänge werden während der Geschäftszeiten im zwei-Stunden-Rhythmus (11:00, 13:00, 15:00 und 17:00) durchgeführt.

Am Mittwoch um 10:00 Uhr bittet Sie der Leiter der Abteilung für Softwareentwicklung, die Datenbank in der Entwicklungsumgebung auf einen möglichst aktuellen Stand aufzufrischen.

Sie müssen eine vollständige Sicherung der produktiv verwendeten Datenbank durchführen, um sie auf dem Datenbankserver der Entwicklungsabteilung wiederherzustellen.

Welche Option verwenden Sie?

(Diese Aufgabe gehört zu einer Reihe von Fragestellungen, für die dieselben Antworten zur Auswahl stehen. Im Hilfetext finden Sie erweiterte Antwortmöglichkeiten.)

A.STANDBY

B.Differenziell

C.Vollständig

D.COPY_ONLY

E.Transaktionsprotokoll

F.Einfach

Korrekte Antwort: D

Erläuterungen:

Um den konfigurierten Sicherungsplan der produktiv verwendeten Datenbank nicht zu

beeinträchtigen, sollte eine Kopiesicherung durchgeführt werden.

Eine Kopiesicherung ist eine SQL Server-Sicherung, die unabhängig von der Sequenz von herkömmlichen SQL Server-Sicherungen erstellt wird. Normalerweise wird beim Erstellen einer Sicherung die Datenbank geändert, und außerdem beeinflusst dies die Art und Weise, wie spätere Sicherungen wiederhergestellt werden. Manchmal kann es sich jedoch als nützlich erweisen, eine Datensicherung für einen bestimmten Zweck vorzunehmen, ohne die allgemeinen Sicherungs- und Wiederherstellungsprozeduren für die Datenbank zu beeinflussen. Kopiesicherungen eignen sich für diesen Zweck.

Die folgenden Typen von Kopiesicherungen sind verfügbar:

Vollständige Kopiesicherungen (alle Wiederherstellungsmodelle). Eine Kopiesicherung kann nicht als differenzielle Basis oder differenzielle Sicherung dienen und wirkt sich nicht auf die differenzielle Basis aus. Die Wiederherstellung einer vollständigen Kopiesicherung entspricht der Wiederherstellung jeder anderen vollständigen Sicherung.

Protokollkopiesicherungen (nur vollständiges und massenprotokolliertes Wiederherstellungsmodell). Eine Protokollkopiesicherung behält den vorhandenen Protokollarchivpunkt bei und wirkt sich daher nicht auf die Sequenz von regulären Protokollsicherungen aus. Protokollkopiesicherungen sind normalerweise nicht nötig. Erstellen Sie stattdessen eine neue routinemäßige Protokollsicherung (mithilfe von WITH NORECOVERY), und verwenden Sie dann diese Sicherung zusammen mit allen vorherigen Protokollsicherungen, die für die Wiederherstellungssequenz erforderlich sind. Eine Protokollkopiesicherung ist manchmal jedoch auch für das Ausführen einer Onlinewiederherstellung nützlich.

Nach einer Kopiesicherung wird das Transaktionsprotokoll nie abgeschnitten.

So erstellen Sie eine Kopiesicherung:

SQL Server Management Studio

Wählen Sie im Dialogfeld Datenbank sichern auf der Seite Allgemein die Option Kopiesicherung aus.

Transact-SQL

Die grundlegende Transact-SQL-Syntax lautet wie folgt:

Für eine vollständige Kopiesicherung:

BACKUP DATABASE database_name TO backup_device ... WITH COPY_ONLY ...

Für eine Protokollkopiesicherung:

BACKUP LOG database_name TO backup_device ... WITH COPY_ONLY ...

PowerShell

Verwenden Sie das Backup-SqlDatabase-Cmdlet mit dem -CopyOnly-Parameter.

Erweiterte Antwortmöglichkeiten:

Hätten Sie es auch gewußt, wenn mehr als die gezeigten 6 Antwortmöglichkeiten zur Auswahl stünden?

A: Massenprotokolliert

B: CONTINUE_AFTER_ERROR

C: Einfach

D: DBO_ONLY

E: COPY_ONLY

F: SKIP
G: RESTART
H: Transaktionsprotokoll
I: NO_CHECKSUM
J: NORECOVERY
K: Differenziell
L: STANDBY
M: CHECKSUM
N: Vollständig

4. Sie sind als Datenbankadministrator für das Unternehmen it-pruefungen.de tätig. Sie administrieren eine Microsoft SQL Server 2012 Instanz. Die Instanz hostet eine Datenbank für die Buchhaltung. Die Dateien der Datenbank sind in einem Storage Area Network (SAN) gespeichert.

Die Datenbank weist die folgenden Merkmale auf:

Die Datendatei mit einer Größe von zwei Terrabyte (TB) befindet sich auf einer dedizierten LUN, die auf dem Server als Laufwerk D: eingebunden ist.

Das Transaktionsprotokoll mit einer Größe von 10 Gigabyte (GB) befindet sich auf einer dedizierten LUN, die auf dem Server als Laufwerk E: eingebunden ist.

Auf Laufwerk D: ist ein Terrabyte freier Speicher verfügbar.

Auf Laufwerk E: sind fünf Gigabyte freier Speicher verfügbar.

Während der Geschäftszeiten werden fortlaufend Änderungen am Datenbestand der Datenbank durchgeführt. Die Geschäftszeiten sind von Montags bis Freitags jeweils von 09:00 Uhr bis 17:00 Uhr. Täglich werden ca. fünf Prozent der Daten geändert.

Die Buchhaltung lädt täglich zwischen 11:15 Uhr und 15:15 Uhr große CSV-Dateien in die Datenbank. Für die Aktion wird entweder das Hilfsprogramm BCP oder die Transact-SQL Anweisung BULK INSERT verwendet. Bei jedem Import werden der Datenbank ca. 3 GB Daten hinzugefügt. Der Datenimport muss in möglichst kurzer Zeit abgeschlossen werden.

Jeden Sonntag um 10:00 Uhr wird eine vollständige Sicherung der Datenbank durchgeführt. Zusätzliche Sicherungsvorgänge werden während der Geschäftszeiten im zwei-Stunden-Rhythmus (11:00, 13:00, 15:00 und 17:00) durchgeführt.

Sie müssen sicherstellen, dass die Datenbank mit möglichst geringem Datenverlust wiederherstellbar ist.

Welches Wiederherstellungsmodell verwenden Sie?

(Diese Aufgabe gehört zu einer Reihe von Fragestellungen, für die dieselben Antworten zur Auswahl stehen. Im Hilfetext finden Sie erweiterte Antwortmöglichkeiten.)

- A.Einfach
- B.Differenziell
- C.Vollständig
- D.CHECKSUM
- E.Massenprotokolliert

F.Transaktionsprotokoll

Korrekte Antwort: C

Erläuterungen:

Mit Wiederherstellungsmodellen wird die Wartung des Transaktionsprotokolls gesteuert. Es stehen drei Wiederherstellungsmodelle zur Verfügung: einfach, vollständig und massenprotokolliert. Für eine Datenbank wird im Allgemeinen das vollständige oder das einfache Wiederherstellungsmodell verwendet.

Einfaches Wiederherstellungsmodell

Das einfache Wiederherstellungsmodell minimiert den Verwaltungsaufwand für das Transaktionsprotokoll, da das Transaktionsprotokoll nicht gesichert wird. Beim einfachen Wiederherstellungsmodell besteht im Falle einer Beschädigung der Datenbank ein erheblich größeres Risiko für Datenverluste. Nur die Daten seit der letzten Sicherung der verloren gegangenen Daten können wiederhergestellt werden. Deshalb sollten die Sicherungsintervalle im Rahmen des einfachen Wiederherstellungsmodells so kurz sein, dass es nicht zu umfangreichen Datenverlusten kommen kann. Die Intervalle sollten jedoch lang genug sein, dass durch den Sicherungsaufwand die produktive Arbeit nicht eingeschränkt wird. Der Aufwand kann gering gehalten werden, wenn die Sicherungsstrategie differenzielle Sicherungen einschließt.

Das einfache Wiederherstellungsmodell ist generell für Benutzerdatenbanken, z. B. für Test- und Entwicklungsdatenbanken, geeignet oder für Datenbanken, die weitgehend schreibgeschützte Daten enthalten, wie z. B. ein Data Warehouse. Das einfache Wiederherstellungsmodell ist für Produktionssysteme, bei denen der Verlust kürzlich vorgenommener Änderungen nicht akzeptabel ist, nicht geeignet. In solchen Fällen empfiehlt sich die Verwendung des vollständigen Wiederherstellungsmodells.

Vollständiges und massenprotokolliertes Wiederherstellungsmodell

Das vollständige Wiederherstellungsmodell und das massenprotokollierte Wiederherstellungsmodell bieten im Vergleich zum einfachen Wiederherstellungsmodell einen besseren Schutz für Daten. Diese Wiederherstellungsmodelle beruhen auf dem Sichern des Transaktionsprotokolls, um eine vollständige Wiederherstellbarkeit bereitzustellen und den Datenverlust für ein breites Spektrum an Zwischenfällen zu verhindern.

Vollständiges Wiederherstellungsmodell

Stellt das normale Datenbankwartungsmodell für Datenbanken bereit, bei denen die Beständigkeit von Transaktionen notwendig ist.

Protokollsicherungen sind erforderlich. Bei diesem Modell werden alle Transaktionen vollständig protokolliert, und die Transaktionsprotokolleinträge werden aufbewahrt, bis sie gesichert wurden. Beim vollständigen Wiederherstellungsmodell kann eine Datenbank bis zur Fehlerquelle wiederhergestellt werden, vorausgesetzt, dass das Protokollfragment nach dem Fehler gesichert werden kann. Das vollständige Wiederherstellungsmodell unterstützt auch die Wiederherstellung einzelner Datenseiten.

Massenprotokolliertes Wiederherstellungsmodell

Bei diesem Wiederherstellungsmodell werden die meisten Massenvorgänge

massenprotokolliert. Es dient ausschließlich als Ergänzung zum vollständigen Wiederherstellungsmodell. Bei bestimmten umfangreichen Massenvorgängen, wie dem Massenimport oder der Indexerstellung, wird durch temporäres Wechseln zum massenprotokollierten Wiederherstellungsmodell die Leistung gesteigert und die Speicherplatzbelegung reduziert. Protokollsicherungen sind weiterhin erforderlich. Beim massenprotokollierten Wiederherstellungsmodell werden wie beim vollständigen Wiederherstellungsmodell die Transaktionsprotokolleinträge so lange aufbewahrt, bis sie gesichert wurden. Dies wird durch umfangreichere Protokollsicherungen und ein höheres Risiko für Datenverluste aufgewogen, da das massenprotokollierte Wiederherstellungsmodell die Wiederherstellung bis zu einem bestimmten Zeitpunkt nicht unterstützt.

Im folgenden Beispiel wird das Wiederherstellungsmodell für die Beispieldatenbank AdventureWorks2008R2 mit Vollständig festgelegt:

```
USE master;
```

```
GO
```

```
ALTER DATABASE AdventureWorks2008R2
```

```
SET RECOVERY FULL;
```

```
GO
```

Erweiterte Antwortmöglichkeiten:

Hätten Sie es auch gewußt, wenn mehr als die gezeigten 6 Antwortmöglichkeiten zur Auswahl stünden?

A: Massenprotokolliert

B: CONTINUE_AFTER_ERROR

C: Einfach

D: DBO_ONLY

E: COPY_ONLY

F: SKIP

G: RESTART

H: Transaktionsprotokoll

I: NO_CHECKSUM

J: NORECOVERY

K: Differenziell

L: STANDBY

M: CHECKSUM

N: Vollständig

5. Sie sind als Datenbankadministrator für das Unternehmen it-pruefungen.de tätig. Sie administrieren eine Microsoft SQL Server 2012 Instanz. Die Instanz hostet eine Datenbank für die Buchhaltung. Die Dateien der Datenbank sind in einem Storage Area Network (SAN) gespeichert.

Die Datenbank weist die folgenden Merkmale auf:

Die Datendatei mit einer Größe von zwei Terrabyte (TB) befindet sich auf einer dedizierten LUN, die auf dem Server als Laufwerk D: eingebunden ist.

Das Transaktionsprotokoll mit einer Größe von 10 Gigabyte (GB) befindet sich auf einer dedizierten LUN, die auf dem Server als Laufwerk E: eingebunden ist.

Auf Laufwerk D: ist ein Terrabyte freier Speicher verfügbar.

Auf Laufwerk E: sind fünf Gigabyte freier Speicher verfügbar.

Während der Geschäftszeiten werden fortlaufend Änderungen am Datenbestand der Datenbank durchgeführt. Die Geschäftszeiten sind von Montags bis Freitags jeweils von 09:00 Uhr bis 17:00 Uhr. Täglich werden ca. fünf Prozent der Daten geändert.

Die Buchhaltung lädt täglich zwischen 11:15 Uhr und 15:15 Uhr große CSV-Dateien in die Datenbank. Für die Aktion wird entweder das Hilfsprogramm BCP oder die Transact-SQL Anweisung BULK INSERT verwendet. Bei jedem Import werden der Datenbank ca. 3 GB Daten hinzugefügt. Der Datenimport muss in möglichst kurzer Zeit abgeschlossen werden.

Jeden Sonntag um 10:00 Uhr wird eine vollständige Sicherung der Datenbank durchgeführt. Zusätzliche Sicherungsvorgänge werden während der Geschäftszeiten im zwei-Stunden-Rhythmus (11:00, 13:00, 15:00 und 17:00) durchgeführt.

Sie müssen sicherstellen, dass die Größe der Sicherungen so klein wie möglich ausfällt.

Welchen Sicherungstyp führen Sie alle zwei Stunden aus?

(Diese Aufgabe gehört zu einer Reihe von Fragestellungen, für die dieselben Antworten zur Auswahl stehen. Im Hilfetext finden Sie erweiterte Antwortmöglichkeiten.)

A.STANDBY

B.Differenziell

C.Vollständig

D.CHECKSUM

E.Massenprotokolliert

F.Transaktionsprotokoll

Korrekte Antwort: F

Erläuterungen:

In Betracht kommen hier differenzielle Sicherungen (BACKUP DATABASE ... WITH DIFFERENTIAL) und Transaktionsprotokollsicherungen (BACKUP LOG).

Differenzielle Sicherung

Eine differenzielle Sicherung basiert auf der aktuellsten, zuvor ausgeführten vollständigen Datensicherung. Mit einer differenziellen Sicherung werden nur die Daten erfasst, die sich seit dieser vollständigen Sicherung geändert haben. Die vollständige Sicherung, auf der eine differenzielle Sicherung basiert wird, wird als Basis der differenziellen Sicherung bezeichnet. Vollständige Sicherungen, mit Ausnahme von Kopiesicherungen, können als Basis für eine Reihe differenzieller Sicherungen dienen, einschließlich Datenbanksicherungen, Teilsicherungen und Dateisicherungen. Die Basissicherung für eine differenzielle Dateisicherung kann in einer vollständigen Sicherung, einer Dateisicherung oder einer Teilsicherung enthalten sein.

Bei einer differenziellen Sicherung wird der Status aller Blöcke (Sammlungen von acht physisch zusammenhängenden Seiten) erfasst, die sich zwischen dem

Erstellungszeitpunkt der differenziellen Datenbank und dem Erstellungszeitpunkt der differenziellen Sicherung geändert hat. Das bedeutet, dass die Größe einer bestimmten differenziellen Sicherung von der Menge an Daten abhängt, die seit der Basissicherung geändert wurden. Grundsätzlich gilt: Je älter die Basis, desto größer die neue differenzielle Sicherung. In einer Reihe differenzieller Sicherungen enthält ein häufig aktualisierter Block möglicherweise bei jeder differenziellen Sicherung unterschiedliche Daten.

Eine differenzielle Sicherung wird relativ bald erstellt, nachdem die zugehörige Basis erheblich kleiner geworden ist als die differenzielle Basis. Hierdurch wird Speicherplatz gespart und der Zeitaufwand für die Sicherung reduziert. Im Laufe der Zeit wird aufgrund der in der Datenbank vorgenommenen Änderungen der Unterschied zwischen der Datenbank und einer bestimmten differenziellen Basis jedoch immer größer. Je mehr Zeit zwischen einer differenziellen Sicherung und der zugehörigen Basis liegt, desto umfangreicher wird die differenzielle Sicherung ausfallen. Dies bedeutet, dass letztendlich differenzielle Sicherungen nahezu die Größe der differenziellen Basis erreichen können. Bei einer großen differenziellen Sicherung geht der Vorteil des schnelleren Sicherns und des geringeren Umfangs verloren.

Transaktionsprotokollsicherungen

Transaktionsprotokollsicherungen sind nur für SQL Server-Datenbanken relevant, die das vollständige oder das massenprotokollierte Wiederherstellungsmodell verwenden.

Vor dem Erstellen von Protokollsicherungen muss bereits mindestens eine vollständige Sicherung durchgeführt worden sein. Danach kann das Transaktionsprotokoll jederzeit gesichert werden, es sei denn, das Protokoll wurde gerade gesichert. Es empfiehlt sich, Protokollsicherungen häufig auszuführen, damit auf diese Weise die Gefahr von Datenverlusten verringert und das Abschneiden des Transaktionsprotokolls angewendet werden kann. In der Regel erstellt ein Datenbankadministrator von Zeit zu Zeit eine vollständige Datenbanksicherung, z. B. einmal pro Woche, sowie optional in kürzeren Abständen eine Reihe von differenziellen Datenbanksicherungen, z. B. täglich. Unabhängig von den Datenbanksicherungen sichert der Datenbankadministrator das Transaktionsprotokoll sehr häufig, z. B. alle 10 Minuten. Der optimale Abstand zwischen Sicherungen ist abhängig von Faktoren wie der Wichtigkeit der Daten, der Größe der Datenbank und der Arbeitsauslastung des Servers.

Erweiterte Antwortmöglichkeiten:

Hätten Sie es auch gewußt, wenn mehr als die gezeigten 6 Antwortmöglichkeiten zur Auswahl stünden?

- A: Massenprotokolliert
- B: CONTINUE_AFTER_ERROR
- C: Einfach
- D: DBO_ONLY
- E: COPY_ONLY
- F: SKIP
- G: RESTART
- H: Transaktionsprotokoll
- I: NO_CHECKSUM

J: NORECOVERY
K: Differenziell
L: STANDBY
M: CHECKSUM
N: Vollständig

6. Sie sind als Datenbankadministrator für das Unternehmen it-pruefungen.de tätig. Sie administrieren einen Microsoft SQL Server 2012 Datenbankserver mit dem Namen SQL1. Der Server hostet eine Datenbank mit dem Namen CB1.

Sie müssen das Auftreten von Deadlocks bei der Ausführung bestimmter gespeicherter Prozeduren untersuchen und die entsprechenden Anweisungen aufzeichnen, um sie später auf einem Testserver erneut abzuspielen.

Was erstellen Sie?

(Diese Aufgabe gehört zu einer Reihe von Fragestellungen, für die dieselben Antworten zur Auswahl stehen. Im Hilfetext finden Sie erweiterte Antwortmöglichkeiten.)

- A. Eine Datenbank-Überwachungsspezifikation
- B. Eine Server-Überwachungsspezifikation
- C. Eine Warnung
- D. Eine SQL Server Profiler Ablaufverfolgung
- E. Einen Ressourcenpool
- F. Eine Sitzung für erweiterte Ereignisse

Korrekte Antwort: D

Erläuterungen:

SQL Server Profiler ist eine funktionsreiche Benutzeroberfläche zum Erstellen und Verwalten von Ablaufverfolgungen sowie zum Analysieren und Wiedergeben von Ablaufverfolgungsergebnissen. Die Ereignisse werden in einer Ablaufverfolgungsdatei gespeichert, die später analysiert oder beim Versuch, ein Problem zu diagnostizieren, zur Wiedergabe einer bestimmten Reihe von Schritten verwendet werden kann.

SQL Server Profiler wird z. B. für folgende Aktivitäten verwendet:

- Schrittweises Untersuchen problematischer Abfragen, um die Ursache des Problems zu ermitteln.

- Suchen und Diagnostizieren von Abfragen, die sehr langsam ausgeführt werden.

- Erfassen der Transact-SQL-Anweisungen, die zu einem Problem geführt haben.

Mithilfe der gespeicherten Ablaufverfolgung kann das Problem auf einem Testserver repliziert werden, um es dort zu diagnostizieren.

- Überwachen der Leistung von SQL Server zur Optimierung der Arbeitsauslastung.

- Korrelieren von Leistungsindikatoren zur Diagnose von Problemen.

Erweiterte Antwortmöglichkeiten:

Hätten Sie es auch gewußt, wenn mehr als die gezeigten 6 Antwortmöglichkeiten zur Auswahl stünden?

- A: Eine Datenbank-Überwachungsspezifikation
- B: Eine Richtlinie
- C: Eine Warnung
- D: Eine SQL Server Profiler Ablaufverfolgung
- E: Einen Ressourcenpool
- F: Eine Sitzung für erweiterte Ereignisse
- G: Eine Server-Überwachungsspezifikation

7. Sie sind als Datenbankadministrator für das Unternehmen it-pruefungen.de tätig. Sie administrieren einen Microsoft SQL Server 2012 Datenbankserver mit dem Namen SQL1. Der Server hostet eine Datenbank mit dem Namen CB1.

Sie müssen verhindern, dass Benutzer Serverüberwachungen auf SQL1 deaktivieren.

Was erstellen Sie?

(Diese Aufgabe gehört zu einer Reihe von Fragestellungen, für die dieselben Antworten zur Auswahl stehen. Im Hilfetext finden Sie erweiterte Antwortmöglichkeiten.)

- A. Eine Datenbank-Überwachungsspezifikation
- B. Eine Richtlinie
- C. Eine Warnung
- D. Eine SQL Server Profiler Ablaufverfolgung
- E. Eine Server-Überwachungsspezifikation
- F. Eine Sitzung für erweiterte Ereignisse

Korrekte Antwort: B

Erläuterungen:

Die richtlinienbasierte Verwaltung ist ein System zum Verwalten einer oder mehrerer Instanzen von SQL Server 2012 und kann Sie beispielsweise in den folgenden Szenarien unterstützen:

Eine Unternehmensrichtlinie verhindert das Aktivieren von Datenbank-E-Mail oder SQL Mail. Eine Richtlinie wird erstellt, um den Serverstatus dieser zwei Funktionen zu überprüfen. Ein Administrator vergleicht den Serverstatus mit der Richtlinie. Wenn der Serverstatus die Bedingungen der Richtlinie nicht einhält, wählt der Administrator den Konfigurationsmodus, und die Richtlinie ändert den Serverstatus so, dass er den Bedingungen der Richtlinie entspricht.

Für die AdventureWorks2012 -Datenbank gilt eine Benennungskonvention, die vorschreibt, dass alle gespeicherten Prozeduren mit den Buchstaben AW_ beginnen. Eine Richtlinie wird erstellt, mit der diese Richtlinie erzwungen werden soll. Ein Administrator testet diese Richtlinie und erhält eine Liste gespeicherter Prozeduren, die diese Richtlinie nicht einhalten. Wenn zukünftig gespeicherte Prozeduren diese Benennungskonvention nicht einhalten, erzeugt die Erstellungsanweisung für die gespeicherte Prozedur einen Fehler.

Richtlinien werden mit Management Studio erstellt und verwaltet. Der Vorgang umfasst

folgende Schritte:

Wählen Sie ein Facet der richtlinienbasierten Verwaltung aus, das die zu konfigurierenden Eigenschaften enthält.

Definieren Sie eine Bedingung, die den Status eines Verwaltungsfacets angibt.

Definieren Sie eine Richtlinie, die die Bedingung, zusätzliche Bedingungen zum Filtern der Zielsätze und den Auswertungsmodus enthält.

Überprüfen Sie, ob eine Instanz von SQL Server in Übereinstimmung mit der Richtlinie ist.

Erweiterte Antwortmöglichkeiten:

Hätten Sie es auch gewußt, wenn mehr als die gezeigten 6 Antwortmöglichkeiten zur Auswahl stünden?

- A: Eine Datenbank-Überwachungsspezifikation
- B: Eine Richtlinie
- C: Eine Warnung
- D: Eine SQL Server Profiler Ablaufverfolgung
- E: Einen Ressourcenpool
- F: Eine Sitzung für erweiterte Ereignisse
- G: Eine Server-Überwachungsspezifikation

8. Sie sind als Datenbankadministrator für das Unternehmen it-pruefungen.de tätig. Sie administrieren eine Microsoft SQL Server 2012 Datenbank mit dem Namen DB1.

DB1 enthält ein Schema mit dem Namen Customers. Das Schema enthält eine Tabelle mit dem Namen Regions.

Ein Benutzer mit dem Namen Tom ist Mitglied einer Rolle mit der Bezeichnung Sales. Tom verfügt über die SELECT-Berechtigung für die Tabelle Regions. Die Rolle Sales verfügt über die SELECT-Berechtigung für das Schema Customers.

Sie müssen sicherstellen, dass sowohl der Rolle Sales als auch Tom das Ausführen der SELECT-Anweisung für alle Tabellen im Schema Customers verweigert wird.

Welche Transact-SQL Anweisung führen Sie aus?

(Diese Aufgabe gehört zu einer Reihe von Fragestellungen, für die dieselben Antworten zur Auswahl stehen. Im Hilfetext finden Sie erweiterte Antwortmöglichkeiten.)

- A. REVOKE SELECT ON Schema::Customers FROM Tom
- B. DENY SELECT ON Object::Regions FROM Tom
- C. EXEC sp_addrolemember 'Sales', 'Tom '
- D. DENY SELECT ON Object::Regions FROM Sales
- E. REVOKE SELECT ON Object::Regions FROM Tom
- F. DENY SELECT ON Schema::Customers FROM Sales

Korrekte Antwort: F

Erläuterungen:

Tom hat die SELECT-Berechtigung einmal über die Mitgliedschaft in der Rolle Sales und

einmal direkt über seinen Benutzer erhalten. Wird der Rolle Sales die SELECT-Berechtigung für das Schema Sales verweigert, wirkt sich dies auf alle Tabellen (und andere Objekte) des Schemas aus.

Da ein verweigertes Recht Vorrang vor einem gewährten Recht hat, wird die SELECT-Berechtigung, die Tom direkt für die Tabelle Sales erhalten hat, außer Kraft gesetzt.

Erweiterte Antwortmöglichkeiten:

Hätten Sie es auch gewußt, wenn mehr als die gezeigten 6 Antwortmöglichkeiten zur Auswahl stünden?

- A: REVOKE SELECT ON Schema::Customers FROM Tom
- B: DENY SELECT ON Object::Regions FROM Tom
- C: EXEC sp_addrolemember 'Sales', 'Tom '
- D: DENY SELECT ON Object::Regions FROM Sales
- E: REVOKE SELECT ON Object::Regions FROM Tom
- F: DENY SELECT ON Schema::Customers FROM Sales
- G: DENY SELECT ON Schema::Customers FROM Tom
- H: EXEC sp_droprolemember 'Sales', 'Tom'
- I: REVOKE SELECT ON Object::Regions FROM Sales
- J: REVOKE SELECT ON Schema::Customers FROM Sales

9.Sie sind als Datenbankadministrator für das Unternehmen it-pruefungen.de tätig. Sie administrieren eine Microsoft SQL Server 2012 Datenbank mit dem Namen DB1.

DB1 enthält ein Schema mit dem Namen Customers. Das Schema enthält eine Tabelle mit dem Namen Regions.

Ein Benutzer mit dem Namen Tom ist Mitglied einer Rolle mit der Bezeichnung Sales. Tom verfügt über die SELECT-Berechtigung für die Tabelle Regions. Die Rolle Sales verfügt über die SELECT-Berechtigung für das Schema Customers.

Sie müssen sicherstellen, dass Tom das Ausführen der SELECT-Anweisung für alle Tabellen im Schema Customers verweigert wird.

Welche Transact-SQL Anweisung führen Sie aus?

(Diese Aufgabe gehört zu einer Reihe von Fragestellungen, für die dieselben Antworten zur Auswahl stehen. Im Hilfetext finden Sie erweiterte Antwortmöglichkeiten.)

- A.DENY SELECT ON Schema::Customers FROM Tom
- B.DENY SELECT ON Object::Regions FROM Tom
- C.EXEC sp_droprolemember 'Sales', 'Tom'
- D.DENY SELECT ON Object::Regions FROM Sales
- E.REVOKE SELECT ON Object::Regions FROM Tom
- F.DENY SELECT ON Schema::Customers FROM Sales

Korrekte Antwort: A

Erläuterungen:

Tom hat die SELECT-Berechtigung einmal über die Mitgliedschaft in der Rolle Sales und einmal direkt über seinen Benutzer erhalten. Wird Tom die SELECT-Berechtigung für das Schema Sales verweigert, wirkt sich dies auf alle Tabellen des Schemas Regions aus.

Da ein verweigertes Recht Vorrang vor einem gewährten Recht hat, wird die SELECT-Berechtigung, die Tom direkt für die Tabelle Sales erhalten hat, ebenfalls außer Kraft gesetzt.

Erweiterte Antwortmöglichkeiten:

Hätten Sie es auch gewußt, wenn mehr als die gezeigten 6 Antwortmöglichkeiten zur Auswahl stünden?

- A: REVOKE SELECT ON Schema::Customers FROM Tom
- B: DENY SELECT ON Object::Regions FROM Tom
- C: EXEC sp_addrolemember 'Sales', 'Tom '
- D: DENY SELECT ON Object::Regions FROM Sales
- E: REVOKE SELECT ON Object::Regions FROM Tom
- F: DENY SELECT ON Schema::Customers FROM Sales
- G: DENY SELECT ON Schema::Customers FROM Tom
- H: EXEC sp_droprolemember 'Sales', 'Tom'
- I: REVOKE SELECT ON Object::Regions FROM Sales
- J: REVOKE SELECT ON Schema::Customers FROM Sales