RDS for SQL Server

Best Practices

Issue 01

Date 2025-07-17

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

1 Best Practices

This chapter describes best practices for working with RDS for SQL Server and provides operational guidelines that you can follow when using this service.

Table 1-1 RDS for SQL Server Best Practices

Reference	Description
Restoring Data from Backup Files to RDS for SQL Server DB Instances	Describes the version restrictions on RDS for SQL Server backup and restoration.
Migrating Data from a Self-Managed SQL Server Database on an ECS to an RDS for SQL Server DB Instance	Describes how to migrate a self-managed SQL Server database on an ECS to an RDS for SQL Server DB instance.
Modifying Parameters of RDS for SQL Server Instances	Describes how to modify parameter templates of RDS for SQL Server DB instances.
Supporting DMVs	Describes how to dynamically manage views through DMV on RDS for SQL Server.
Using the Import and Export Function to Migrate Data from a Local Database to an RDS for SQL Server DB Instance	Describes how to migrate an on-premises SQL Server database to an RDS for SQL Server DB instance.
Creating a Subaccount of rdsuser	Describes the permissions of the **rdsuser** account and how to create and manage IAM users under the **rdsuser** account.
Creating tempdb Files	Describes how to create tempdb temporary data files on RDS for SQL Server.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Reference	Description
Microsoft SQL Server Publication and Subscription	Describes how RDS for SQL Server provides the subscription function.
Installing a C# CLR Assembly in RDS for SQL Server	Describes how to add a c#CLR assembly on RDS for SQL Server.
Creating a Linked Server for an RDS for SQL Server DB Instance	Describes how to create a linked server to access another RDS for SQL Server DB instance.
Deploying SQL Server Reporting Services (SSRS) on RDS for SQL Server	Describes how to deploy SQL Server Reporting Services (SSRS) in RDS for SQL Server.
Shrinking an RDS for SQL Server Database	Describes how to use a stored procedure to shrink the size of the data and log files in a specified RDS for SQL Server database.
Using DAS to Create and Configure Agent Job and DBLink on the Master and Slave Databases for RDS for SQL Server Instances	Describes how to use DAS to create and configure agent jobs and DBLink on primary and standby RDS for SQL Server DB instances.
Creating a Job for Scheduled Instance Maintenance	Describes how to create a scheduled SQL agent job to re-create indexes, update statistics, and shrink the database.
Security Best Practices	Provides guidance on RDS for SQL Server security configurations.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2 Restoring Data from Backup Files to RDS for SQL Server DB Instances

This section describes how to use automated or manual backup files to restore a DB instance to the status when the backup was created.

Best Practices

- You can create a full backup file for your DB instance and use OBS and DRS to restore the backup file to an RDS for SQL Server DB instance.
- The restoration must be from an earlier database version to the same or a later version. The version of local backups must be earlier than or the same as the version of the destination DB instance to be restored.

  NOTE

  For example, if the local database version is Microsoft SQL Server 2012 Standard Edition, you can only restore the local backups to Standard or Enterprise Edition of Microsoft SQL Server 2014 or 2016. You cannot restore the local backups to any versions of Microsoft SQL Server 2008 or Web Editions of Microsoft SQL Server 2014 and 2016.
- For operation details on the RDS console, see **Restoring a DB Instance to a Point in Time** and **Restoring a DB Instance from a Backup**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

3 Migrating Data from a Self-Managed SQL Server Database on an ECS to an RDS for SQL Server DB Instance

Scenarios

- You have created a Microsoft SQL Server database on an ECS.
- The self-managed SQL Server database version on the ECS cannot be later than the version of the RDS for SQL Server DB instance.
- You have installed the SQL Server Management Studio (SSMS).

Procedure

Step 1 Create an ECS.

NOTE

The ECS and the RDS DB instance must be in the same region and VPC.

Step 2 Install Microsoft SQL Server 2008, 2012, or 2014 on the ECS.

NOTE

The Microsoft SQL Server installed on the ECS must be Standard or Enterprise Edition. It is recommended that the Microsoft SQL Server version be the same as the RDS DB instance version.

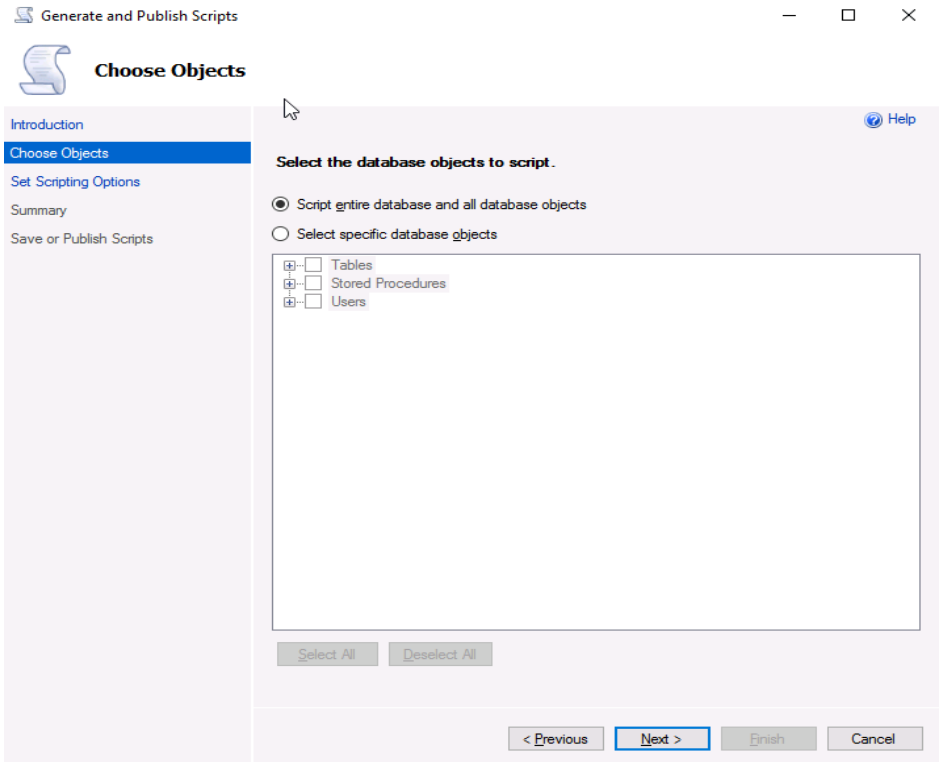
Step 3 Upload a local .bak file to the ECS and use Microsoft SQL Server to restore the local file to the RDS DB instance.

Step 4 Use the script generation tool provided by Microsoft SQL Server to generate a database structure script.

1. Right-click the database whose schema script needs to be generated and choose **Tasks** > **Generate Scripts**.
2. On the **Choose Objects** page, choose database objects to script, as shown in **Figure 3-1**. Then, click **Next**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-1 Choosing objects



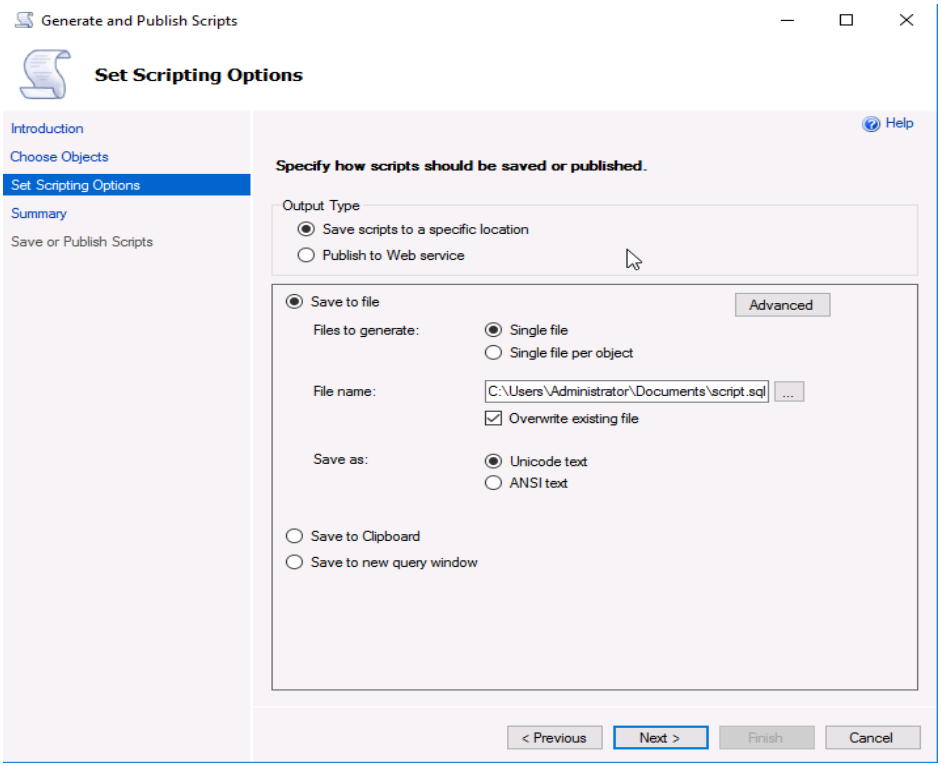
3. On the **Set Scripting Options** page, specify a directory for saving the script.

   NOTE

   You are advised to save the script locally and generate an SQL script for execution.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

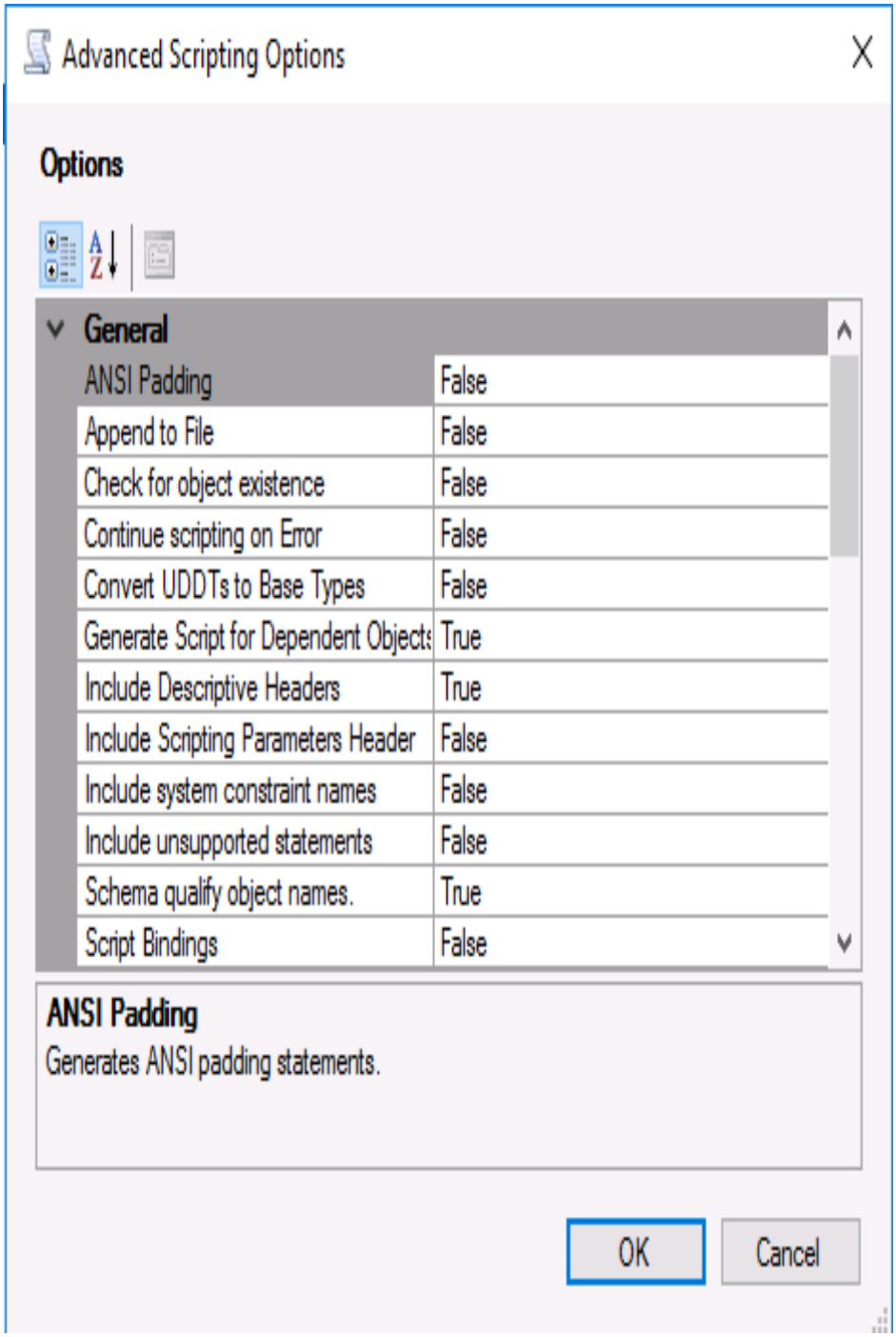
Figure 3-2 Specifying a directory for saving the script



4. Click **Advanced**. In the displayed **Advanced Scripting Options** dialog box, specify scripting options for triggers, indexes, unique keys, the primary key, and server version. Then, click **OK**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 3-3 Specifying advanced scripting options



NOTE

Generate Script for Dependent Objects indicates the script data type option.

5. Click **Next** to generate the script.

Step 5 Use the SSMS client to connect to the RDS DB instance and open the generated SQL script.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

> **NOTE**
>
> You need to create an empty database, and then use the script to create structures in the database.

Step 6 Use the import and export function provided by Microsoft SQL Server to migrate data.

1. Right-click the database where data is to be imported and choose **Tasks** > **Import Data**.
2. Click **Next**.
3. On the **Choose a Data Source** page, select a data source and click **Next**.
4. On the **Choose a Destination** page, select a destination database and click **Next**.
   - **Destination**: Select **SQL Server Native Client** (depending on your destination database type).
   - **Server name**: Enter the IP address and port number of the destination DB instance.
   - **Authentication**: Select **Use SQL Server Authentication**. Then, set **User name** to **rdsuser**, and **Password** to the password of **rdsuser**.
   - **Database**: Select the destination database where data is to be imported.
5. Select **Copy data from one or more tables or views** and click **Next**.
6. On the **Select Source Tables and Views** page, select the tables and views that you want to copy. Then, click **Edit Mappings**. In the displayed dialog box, select **Enable identity insert** and edit mappings based on your requirements.
7. Click **Next**.
8. Select **Run immediately** and click **Next**.
9. Click **Finish** to import data. You can view the progress. About 4,000 rows can be processed per second.

----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

4 Modifying Parameters of RDS for SQL Server Instances

You can modify parameters in custom parameter templates.

Each DB instance is assigned with a group of parameters when it is being created and modifications to these parameters do not affect other DB instances.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target DB instance.

Step 5 On the **Parameters** page, modify the parameters as required.

NOTE

- You cannot modify parameters in a default parameter template.
  - Each Microsoft SQL Server version has a unique default parameter template.
  - To apply a default parameter template to the current DB instance, choose **Parameter Templates** page in the navigation pane on the left, locate the target template on the **Default Templates** page, and click **Apply** in the **Operation** column.
- You can modify parameters in a custom template.
  - To create a custom template, choose **Parameter Templates** page in the navigation pane on the left, click **Create Parameter Template** on the **Custom Templates** page, and configure required information in the displayed dialog box. Then, click **OK**.
  - After you save modifications to the parameters in the custom template, you can apply this parameter template to multiple DB instances running corresponding versions.

You can modify the parameters listed in **Table 4-1** to improve DB instance performance.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 4-1 Parameters

Parameter	Description	Application Scenario
max degree of parallelism	Specifies the maximum degree of parallelism option. When an RDS for SQL Server DB instance runs on a computer with more than one microprocessor or CPU, RDS for SQL Server detects the best degree of parallelism (the number of processors used to run a single statement) for each parallel plan execution. The default value is **0**.	• If the DB instance is used for querying results, set the parameter to **0**. • If the DB instance is used for operations such as inserting, updating, and deleting, set the parameter to **1**.
max server memory (MB)	Specifies the server memory option. It is used to reconfigure the amount of memory (in MB) in the buffer pool used by a Microsoft SQL Server DB instance.	You are advised to retain the default value for this parameter. If you want to modify this parameter, the value of this parameter must be: • No less than 2 GB. • Not greater than 95% of the maximum memory of the DB instance.
user connections	Specifies the maximum number of simultaneous user connections allowed on Microsoft SQL Server. Default value: **1000**	• If this parameter is set to **0**, the number of connections to the DB instance is not limited. • Allowed values: Values excluding 1 to 10

- To save the modifications, click **Save**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

After the parameter values are modified, you can click **Change History** to view the modification details.

----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

5 Supporting DMVs

RDS for SQL Server supports dynamic management views (DMVs), which enables users to quickly find 10 SQL statements with the highest performance consumption.

Scenarios

- A performance bottleneck occurs and the database execution efficiency becomes low.
- The monitoring result shows that the CPU and I/O are high in some time segments.

Procedure

Step 1 Use the **rdsuser** account to connect to the target DB instance through a client and run the following statements on the management plane:

```
declare @DatabaseName nvarchar(100)
set @DatabaseName = 'Wisdom_TT_ODS'

select top 100
DB_NAME(st.dbid) as DBName, OBJECT_NAME(st.objectid,st.dbid) as ObjectName,
substring(st.text,(qs.statement_start_offset/2)+1,((case qs.statement_end_offset when -1 then
datalength(st.text) else qs.statement_end_offset end - qs.statement_start_offset)/2) + 1) as
Statement,
st.text as Query,
qp.query_plan,
plan_generation_num,
creation_time,
last_execution_time,
execution_count,
total_worker_time,
min_worker_time,
max_worker_time,
total_logical_reads,
min_logical_reads,
max_logical_reads,
total_elapsed_time,
min_elapsed_time,
max_elapsed_time,
total_rows,
min_rows,
max_rows,
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
total_worker_time/execution_count as avg_worker_time,          --Average CPU
duration
total_logical_reads/execution_count as avg_logical_reads,      --Average logical reads
total_elapsed_time/execution_count as avg_elapsed_time,        --Average total
duration
total_rows/execution_count as avg_rows,                        --Average data processing
rows
sql_handle,
plan_handle,
query_hash,
query_plan_hash
from sys.dm_exec_query_stats qs
cross apply sys.dm_exec_sql_text(plan_handle) st
cross apply sys.dm_exec_query_plan(plan_handle) qp
where st.dbid=DB_ID(@DatabaseName)
and text not like '%sys.%'and text not like '%[[]sys]%'
order by avg_worker_time desc
```

Step 2 You can view the SQL execution records and resource consumption details of the corresponding database in the query result.

----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

6 Using the Import and Export Function to Migrate Data from a Local Database to an RDS for SQL Server DB Instance

Scenarios

- You have created a local Microsoft SQL Server database.
- The local database version cannot be later than the version of the destination RDS for SQL Server DB instance.
- You want to migrate only tables instead of the whole database.

Procedure

Step 1 **Log in to the management console**.

Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane on the left, choose **Connectivity & Security**.

Step 6 In the **Connection Information** area, click **Bind** next to the **EIP** field.

Step 7 In the displayed dialog box, select an EIP and click **Yes**.

Step 8 Install the SSMS client locally and use the EIP to connect to the RDS DB instance.

NOTE

Click **here** to download the SSMS client.

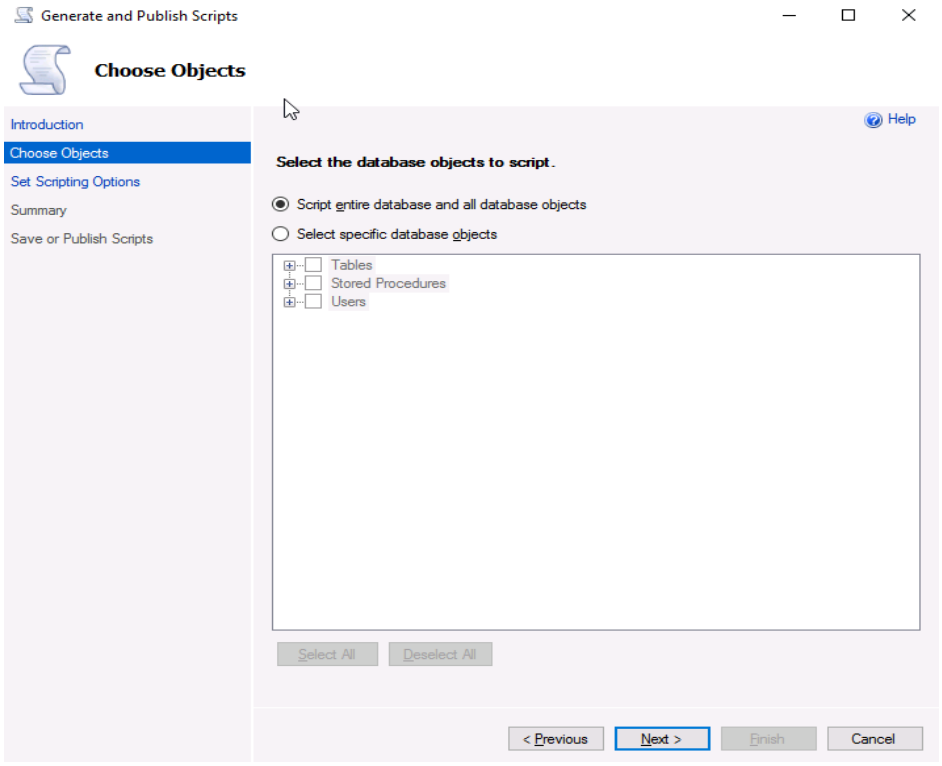
Step 9 Use the script generation tool provided by Microsoft SQL Server to generate a database structure script.

1. Right-click the database whose schema script needs to be generated and choose **Tasks** > **Generate Scripts**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2. On the **Choose Objects** page, choose database objects to script, as shown in Figure 6-1. Then, click **Next**.

Figure 6-1 Choosing objects



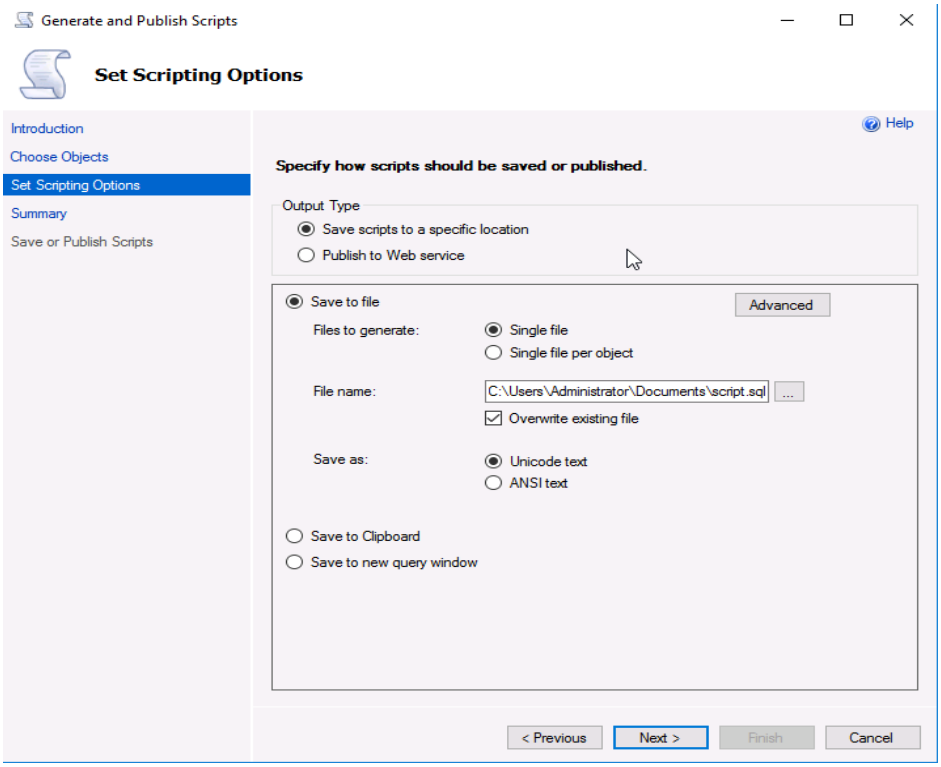
3. On the **Set Scripting Options** page, specify a directory for saving the script.

NOTE

You are advised to save the script locally and generate an SQL script for execution.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

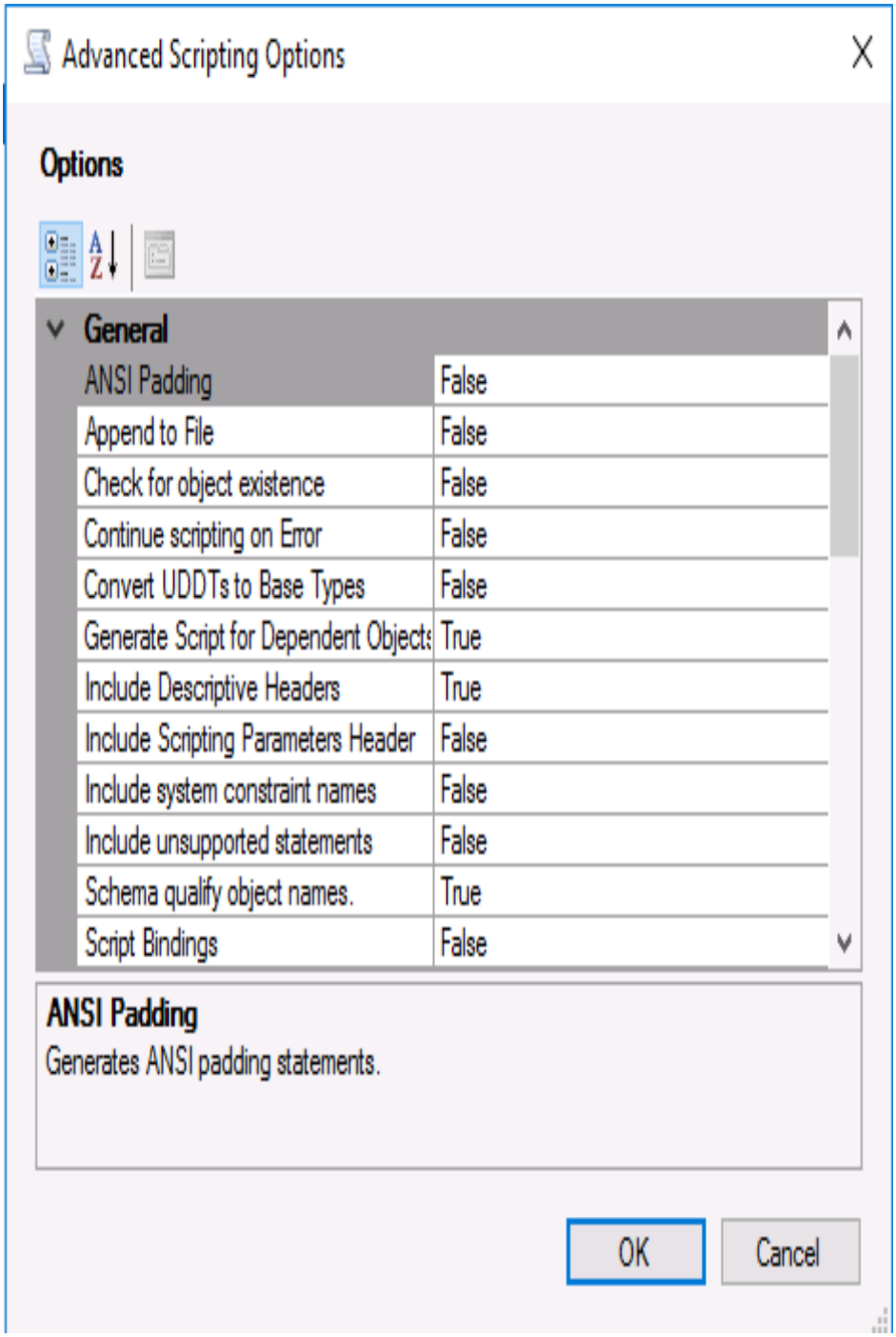
Figure 6-2 Specifying a directory for saving the script



4. Click **Advanced**. In the displayed **Advanced Scripting Options** dialog box, specify scripting options for triggers, indexes, unique keys, the primary key, and server version. Then, click **OK**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 6-3 Specifying advanced scripting options



NOTE

Generate Script for Dependent Objects indicates the script data type option.

5. Click **Next** to generate the script.

Step 10 Use the SSMS client to connect to the RDS DB instance and open the generated SQL script.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

> **NOTE**
>
> You need to create an empty database, and then use the script to create structures in the database.

Step 11 Use the import and export function provided by Microsoft SQL Server to migrate data.

1. Right-click the database where data is to be imported and choose **Tasks** > **Import Data**.
2. Click **Next**.
3. On the **Choose a Data Source** page, select a data source and click **Next**.
4. On the **Choose a Destination** page, select a destination database and click **Next**.
   - **Destination**: Select **SQL Server Native Client** (depending on your destination database type).
   - **Server name**: Enter the IP address and port number of the destination DB instance.
   - **Authentication**: Select **Use SQL Server Authentication**. Then, set **User name** to **rdsuser**, and **Password** to the password of **rdsuser**.
   - **Database**: Select the destination database where data is to be imported.
5. Select **Copy data from one or more tables or views** and click **Next**.
6. On the **Select Source Tables and Views** page, select the tables and views that you want to copy. Then, click **Edit Mappings**. In the displayed dialog box, select **Enable identity insert** and edit mappings based on your requirements.
7. Click **Next**.
8. Select **Run immediately** and click **Next**.
9. Click **Finish** to import data. You can view the progress. About 4,000 rows can be processed per second.

----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

7 Creating a Subaccount of rdsuser

Scenarios

This section describes how to create a subaccount and grant permissions to the subaccount. **Permissions of rdsuser** lists the permissions supported by **rdsuser**.

Prerequisites

You have created a database. For details, see **Creating a Database**.

Procedure

Step 1 Log in to the instance through DAS.

1. **Log in to the management console**.
2. Click  in the upper left corner and select a region.
3. Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.
4. Locate the target instance and click **Log In** in the **Operation** column.
5. On the displayed page, configure required parameters.

Table 7-1 Instance login

Parameter	Description
Login Username	Enter **rdsuser**.
Password	Enter the password of **rdsuser**. **NOTE** You can select **Remember Password** so that you can directly log in to the instance next time.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Collect Metadata Periodically	Enable this function as required. If this function is enabled: – DAS can store structure definition data such as database names, table names, and field names in instances, but does not store data in tables. – Metadata is collected in the early morning every day.
Show Executed SQL Statements	Enable this function as required. This function allows you to view executed SQL statements. You can re-execute an SQL statement without having to enter it again.

6. Click **Log In**.

Step 2 Create a subaccount.

1. On the main menu of the DAS console, choose **Account Management** > **Login Name**.
2. On the displayed page, click **Create Login Name**.
3. On the **Create Login Name** page, configure login information.

Table 7-2 Login information

Parameter	Description
Login Name	Enter a new login name.
Authentication Type	The value is fixed to **Microsoft SQL Server Authentication**.
Password	The password for the new login username must meet the following requirements: – It must contain at least three of the following: uppercase letters, lowercase letters, digits, and special characters ~! @#$^*-_=+?% – It must be 8 to 128 characters long. – It cannot contain the login username. – It cannot be a weak password.
Confirm Password	Enter the password again. **NOTE** For security purposes, select **Enforce Password Policy**.
Default Database	From the drop-down list, select a database the new login user will log in by default.
Default Language	Select a language for the new login user.

4. Click **Save**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

5. Click **Back to Login Name List**.
6. In the login name list, view the new login name.

Step 3 Grant permissions to the new login user.

NOTE

- Table 7-3 describes how to add a single permission. To add multiple permissions to the new login user at the same time, for example, to grant both read and write permissions, select both **db_datareader** and **db_datawriter** on the **Edit Database Role** page.
- For details about the permissions supported by **rdsuser**, see **Permissions of rdsuser**.

Table 7-3 Permissions that can be granted to a subaccount

Permission	Procedure
Database operation permission	1. Locate the new login name and click **Edit** in the **Operation** column. 2. On the displayed page, click the **User Mapping** tab. 3. In the **Users mapped to this login** list, click **Edit** in the row where both the user database and the new login username exist. 4. On the **Edit Database Role** page, select **db_owner** and click **OK**. 5. Click **Save**.
Server role permission	1. Locate the new login name and click **Edit** in the **Operation** column. 2. On the displayed page, click the **Server Roles** tab. 3. In the **Server Roles** list, select the desired server role. 4. Click **Save**.
Securable object permission	1. Locate the new login name and click **Edit** in the **Operation** column. 2. On the displayed page, click the **Securables** tab. 3. In the securables list, select the desired server permission. 4. Click **Save**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Permission	Procedure
Read-only permission	1. Locate the new login name and click **Edit** in the **Operation** column. 2. On the displayed page, click the **User Mapping** tab. 3. In the **Users mapped to this login** list, click **Edit** in the row where both the user database and the new login username exist. 4. On the **Edit Database Role** page, select **db_datareader** and click **OK**. 5. Click **Save**.
Write permission	1. Locate the new login name and click **Edit** in the **Operation** column. 2. On the displayed page, click the **User Mapping** tab. 3. In the **Users mapped to this login** list, click **Edit** in the row where both the user database and the new login username exist. 4. On the **Edit Database Role** page, select **db_datawriter** and click **OK**. 5. Click **Save**.

----End

Permissions of rdsuser

Table 7-4 Permissions of rdsuser

Name	Category	Permission
DB instance permissions	DB instance role permissions	[processadmin]
		[setupadmin]
	DB instance object permissions	ALTER ANY CONNECTION
		ALTER ANY LOGIN
		ALTER ANY SERVER ROLE
		ALTER SERVER STATE
		ALTER TRACE
		CONNECT ANY DATABASE
		CONTROL SERVER
		CONNECT SQL

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Name	Category	Permission
		CREATE ANY DATABASE
		SELECT ALL USER SECURABLES
		VIEW ANY DEFINITION
		VIEW ANY DATABASE
		VIEW SERVER STATE
	Database permissions	master: Public
		Msdb: Public SQLAgentUserRole
		Model: Public
		Rdsadmin: Public
		OtherDB: Db_Owner

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

8 Creating tempdb Files

Scenarios

The tempdb system database is a global resource that is available to all users connected to an instance of SQL Server or SQL Database. It is a temporary database that cannot store data permanently. It is used to process intermediate data for various requests in the instance. Physical properties of tempdb in SQL Server are classified into the primary data files (.mdf), secondary data files (.ndf), and log files (.ldf). **tempdb** is re-created every time SQL Server is started.

There may be some issues or even service interruption if applications frequently create and drop tempdb files, especially in high-concurrency scenarios.

Microsoft recommends that the tempdb files be divided into multiple files. Generally, the number of files depends on the number of vCPUs (logical). If the number of vCPUs is greater than eight, use eight data files and then if contention continues, increase the number of data files by multiples of 4 until the contention is reduced to acceptable levels or make changes to the workload/code.

For more information, see **tempdb Database** in the Microsoft official website.

Constraints

- By default, each RDS for SQL Server instance running SQL Server 2008, 2012, or 2014 Edition has one tempdb file, each instance running SQL Server 2016 Edition has four tempdb files, and each instance running SQL Server 2017 Edition has eight tempdb files.
- Each RDS for SQL Server instance has only one log file no matter which SQL Server Edition they run.

Application Scenario

You need to determine the number of tempdb files to be created based on the instance specifications and scenarios. The following uses an example to show how to create 8 tempdb files for a SQL Server 2014 Enterprise Edition instance with 32 vCPUs.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Prerequisites

- Visit the **Microsoft** website and obtain the SQL Server Management Studio installation package. Double-click the installation package and complete the installation as instructed.
- Create an instance with 32 vCPUs running Microsoft SQL Server 2014 Enterprise Edition. For details, see **Buying an RDS for SQL Server DB Instance**.

Procedure

Step 1 Start SQL Server Management Studio.

Step 2 Choose **Connect** > **Database Engine**. In the displayed dialog box, enter login information.

Figure 8-1 Connecting to the server

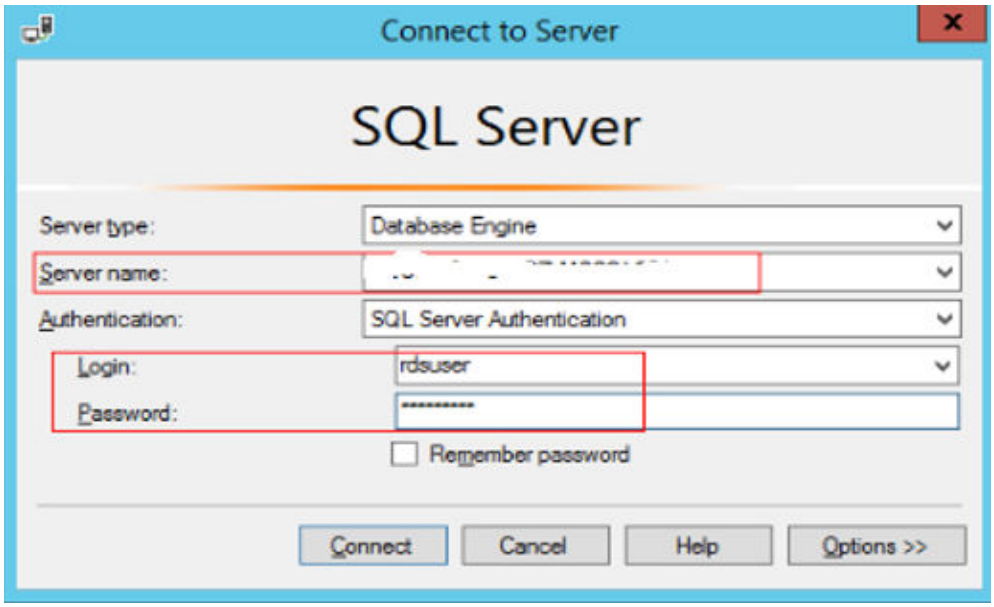


Table 8-1 Parameter description

Parameter	Description
Server name	Indicates the IP address and database port of the DB instance. Use a comma (,) to separate them. For example: x.x.x.x,8080. • The IP address is the EIP that has been bound to the DB instance. • The database port is that displayed on the **Connectivity & Security** page.
Authentication	Indicates the authentication mode. Select **SQL Server Authentication**.
Login	Indicates the database account used for accessing the instance. The default administrator account is **rdsuser**.

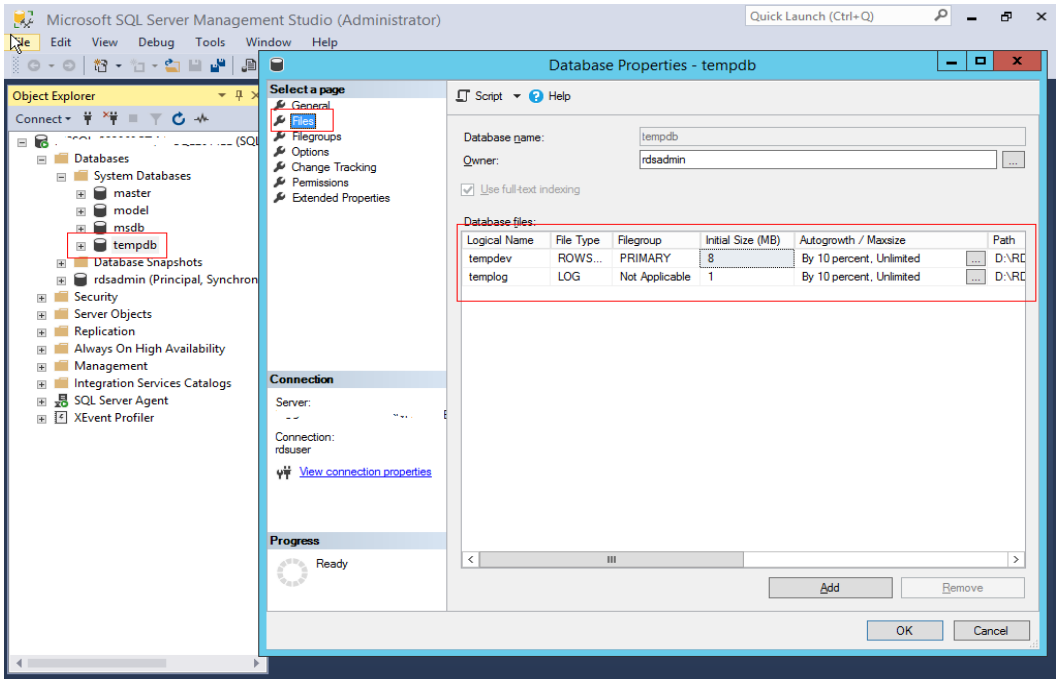
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
Password	Indicates the password of the database account.

Step 3 View the tempdb information.

- Choose **Databases** > **System Databases**> **tempdb**. Right-click **tempdb** and choose **Database Properties**. In the displayed dialog box, view the current tempdb information.

  Figure 8-2 Viewing current tempdb information



- You can also run the following SQL statements to query the tempdb information:

  SELECT name AS FileName,
  size*1.0/128 AS FileSizeInMB,
  CASE max_size
  WHEN 0 THEN 'Autogrowth is off.'
  WHEN -1 THEN 'Autogrowth is on.'
  ELSE 'Log file grows to a maximum size of 2 TB.'
  END,
  growth AS 'GrowthValue',
  'GrowthIncrement' =
  CASE
  WHEN growth = 0 THEN 'Size is fixed.'
  WHEN growth > 0 AND is_percent_growth = 0
  THEN 'Growth value is in 8-KB pages.'
  ELSE 'Growth value is a percentage.'

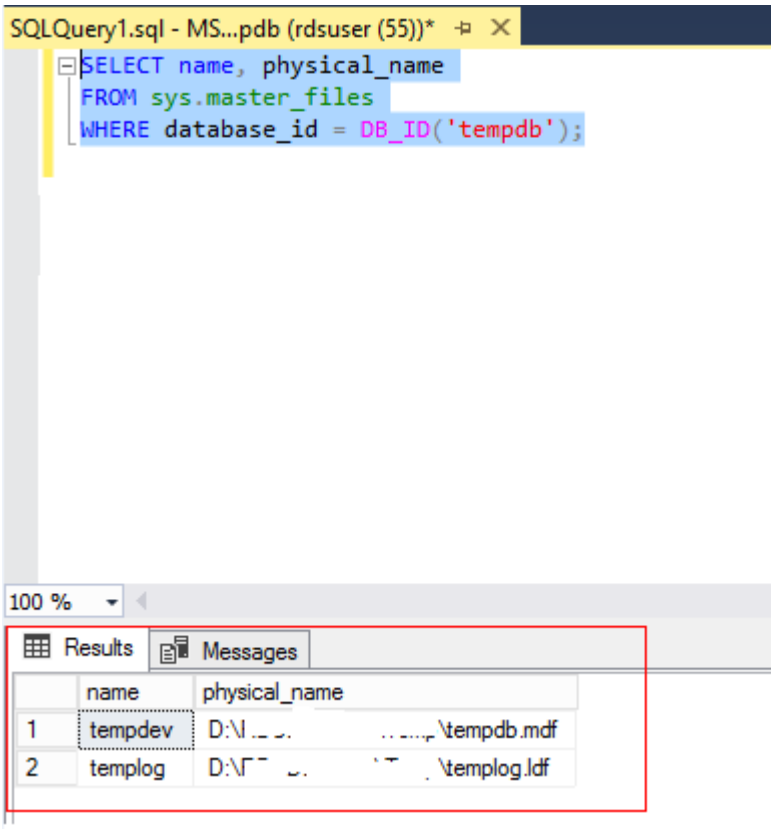
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
END
FROM tempdb.sys.database_files;
GO
```

Step 4 Run the following statements to query the tempdb file name of the current DB instance:

```
SELECT name, physical_name
FROM sys.master_files
WHERE database_id = DB_ID('tempdb');
```

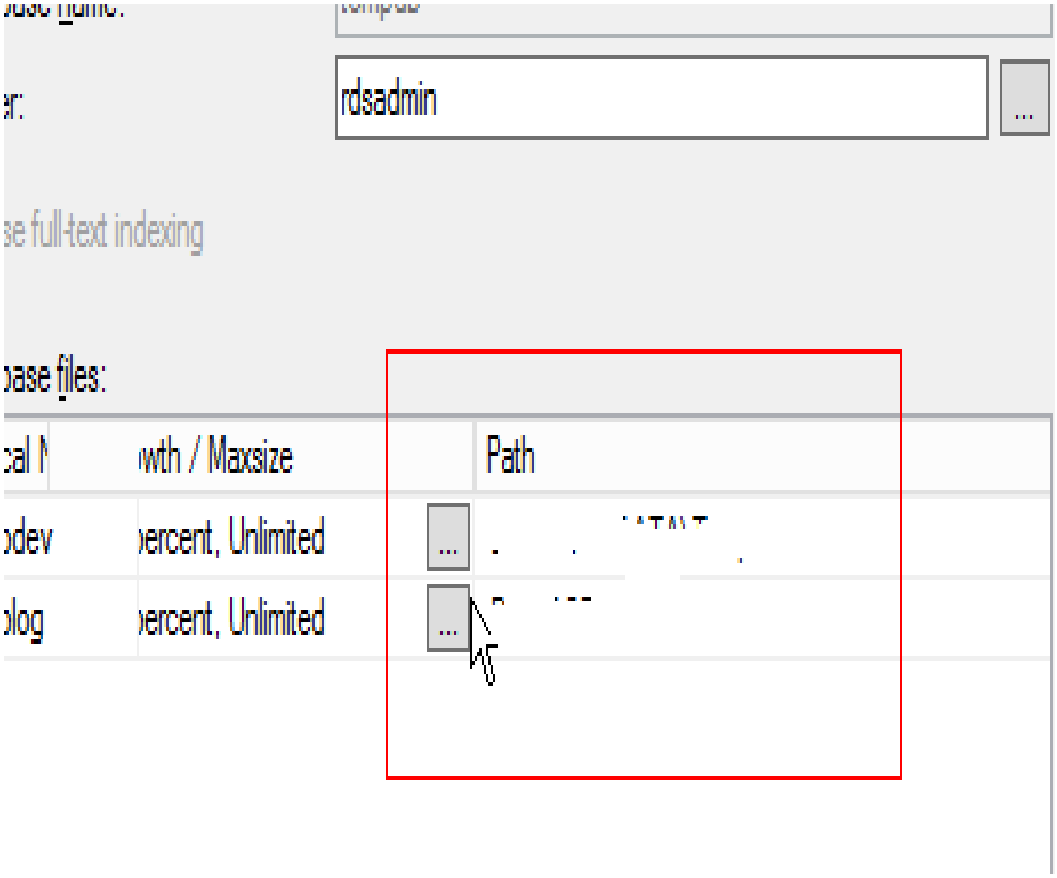
Figure 8-3 Viewing tempdb file names



Step 5 On the **Files** tab in **Step 3**, view the paths of tempdb files.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 8-4 Viewing tempdb paths

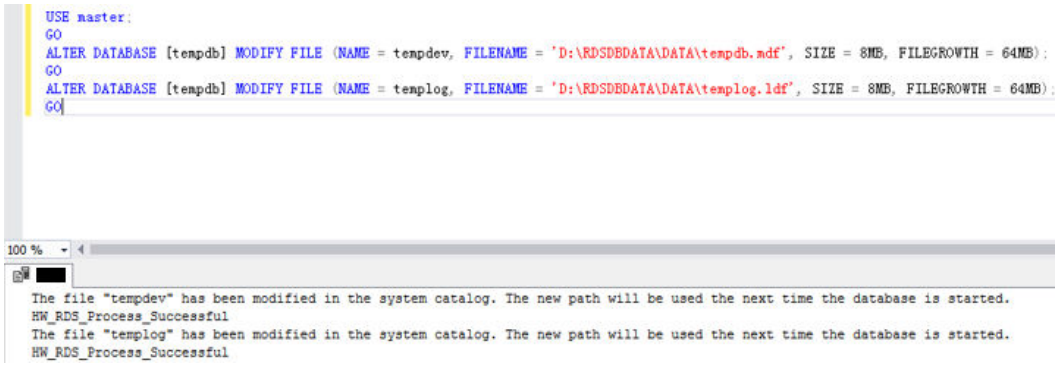


Step 6 Run the following statements to migrate the tempdb files to **D:\RDSDBDATA\DATA** and specify the initial size and growth as required.

```
USE master;
GO
ALTER DATABASE [tempdb] MODIFY FILE (NAME = tempdev, FILENAME = 'D:\RDSDBDATA\DATA\tempdb.mdf', SIZE = 8MB, FILEGROWTH = 64MB);
GO
ALTER DATABASE [tempdb] MODIFY FILE (NAME = templog, FILENAME = 'D:\RDSDBDATA\DATA\templog.ldf', SIZE = 8MB, FILEGROWTH = 64MB);
GO
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 8-5 Moving tempdb files



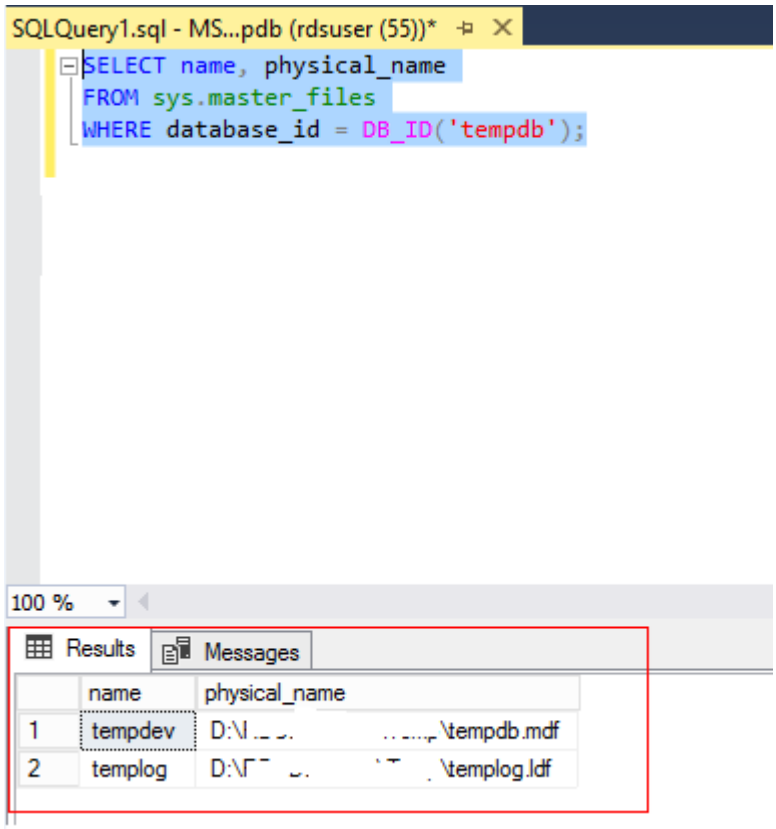
Step 7 On the **Instances** page of the RDS console, locate the target DB instance and choose **More** > **Reboot** in the **Operation** column to reboot the DB instance.

You can also click the target DB instance. On the displayed page, click **Reboot** in the upper right corner of the page.

Step 8 Run the following SQL statements to check whether the tempdb files are successfully migrated:

```
SELECT name, physical_name
FROM sys.master_files
WHERE database_id = DB_ID('tempdb');
```

Figure 8-6 Viewing the migration results



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Step 9 Configure the file name, initial size, and growth as required. Add tempdb files using either of the following methods:

- Adding tempdb files through SQL statements

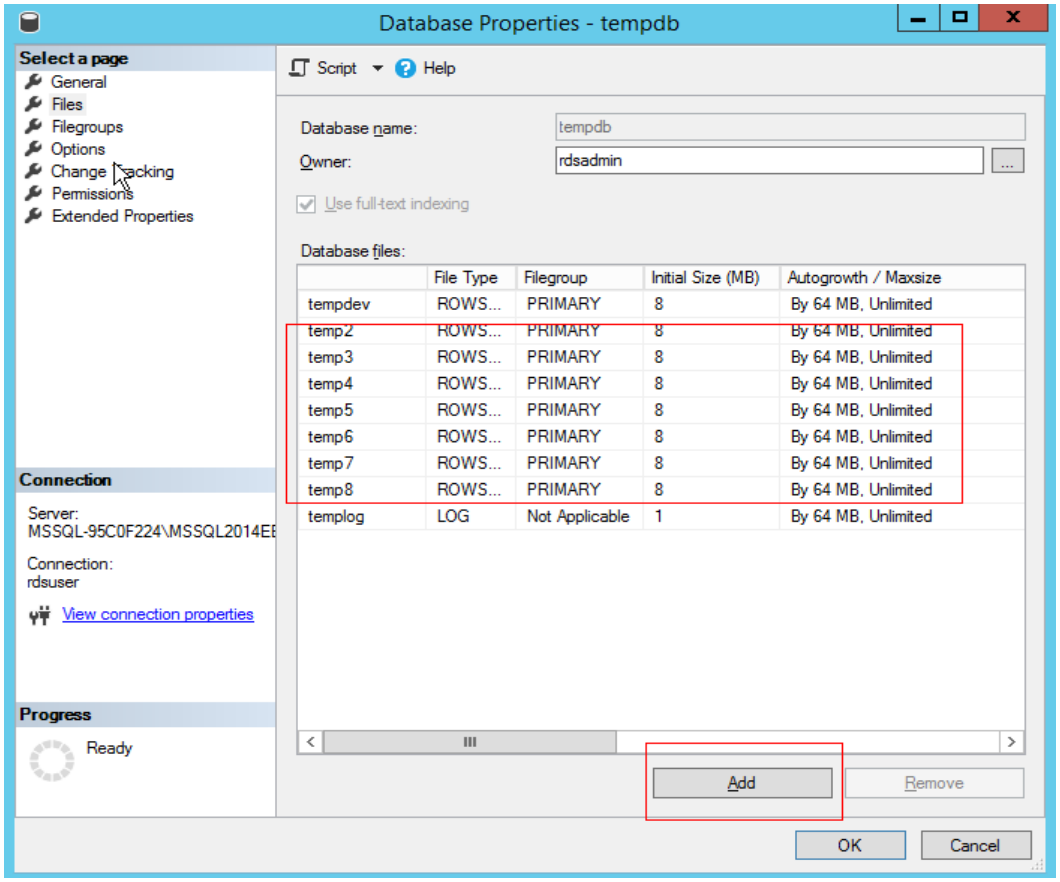
  Based on the number of vCPUs and tempdb files to be added, set the initial size to 8 MB and file growth to 64 MB.

  ```
  USE [master]
  GO
  ALTER DATABASE [tempdb] ADD FILE ( NAME = N'temp2', FILENAME = N'D:\RDSDBDATA\DATA\tempdb2.ndf', SIZE = 8MB, FILEGROWTH = 64MB)
  GO
  ALTER DATABASE [tempdb] ADD FILE ( NAME = N'temp3', FILENAME = N'D:\RDSDBDATA\DATA\tempdb3.ndf', SIZE = 8MB, FILEGROWTH = 64MB)
  GO
  ALTER DATABASE [tempdb] ADD FILE ( NAME = N'temp4', FILENAME = N'D:\RDSDBDATA\DATA\tempdb4.ndf', SIZE = 8MB, FILEGROWTH = 64MB)
  GO
  ALTER DATABASE [tempdb] ADD FILE ( NAME = N'temp5', FILENAME = N'D:\RDSDBDATA\DATA\tempdb5.ndf', SIZE = 8MB, FILEGROWTH = 64MB)
  GO
  ALTER DATABASE [tempdb] ADD FILE ( NAME = N'temp6', FILENAME = N'D:\RDSDBDATA\DATA\tempdb6.ndf', SIZE = 8MB, FILEGROWTH = 64MB)
  GO
  ALTER DATABASE [tempdb] ADD FILE ( NAME = N'temp7', FILENAME = N'D:\RDSDBDATA\DATA\tempdb7.ndf', SIZE = 8MB, FILEGROWTH = 64MB)
  GO
  ALTER DATABASE [tempdb] ADD FILE ( NAME = N'temp8', FILENAME = N'D:\RDSDBDATA\DATA\tempdb8.ndf', SIZE = 8MB, FILEGROWTH = 64MB)
  GO
  ```

- Adding tempdb files through SQL Server Management Studio On the **Files** tab in **Step 3**, click **Add**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 8-7 Adding tempdb files through the client

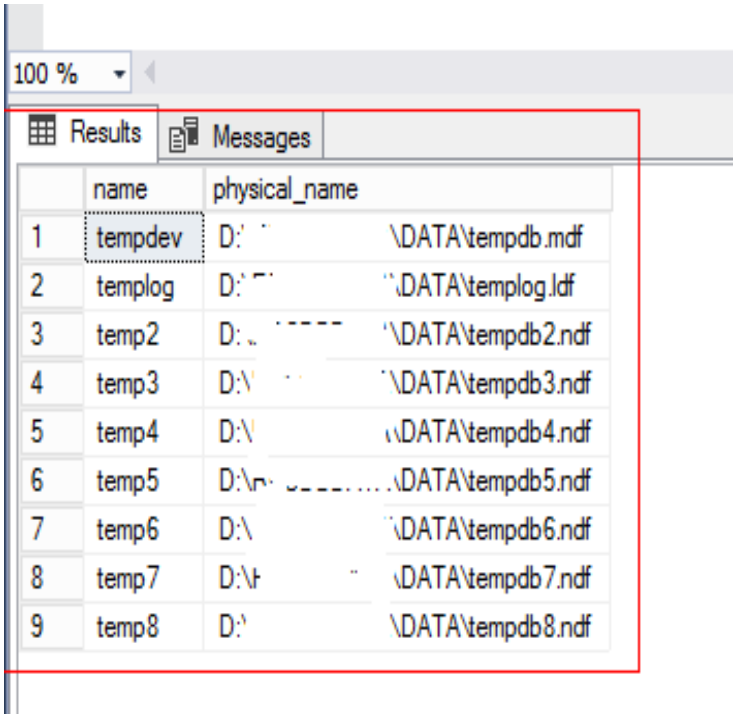


Step 10 After the configuration is complete, reboot the DB instance again by referring to **Step 7**.

Step 11 Repeat **Step 8** to check whether the tempdb files are successfully added.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 8-8 Viewing the added tempdb files



----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

9 Microsoft SQL Server Publication and Subscription

The publication and subscription function provided by Microsoft SQL Server uses the replication technology for data synchronization. This function makes it possible to split data read and write operations and synchronize offline and online data.

This section describes how to use SQL Server Management Studio (SSMS) to configure publication and subscription. You can create publications and subscriptions for RDS for SQL Server instances on the console. For details, see **Creating a Publication**.

Preparations

Environments

1. Local environment: a local database running Microsoft SQL Server 2014 Standard Edition in Windows
2. Online environment:
   - One single DB instance with 2 vCPUs and 16 GB of memory running Microsoft SQL Server 2014 Standard Edition, bound with an EIP
   - One primary/standby DB instance with 4 vCPUs and 8 GB of memory running Microsoft SQL Server 2014 Standard Edition, bound with an EIP

Environment Setup

- Publisher: maintains source data and distributes specific data to the distributor. In this example, use the local server as the publisher.
  a. Use SSMS to log in to the local database as user **sa**. Right-click the **Replication** folder and then click **Configure Distribution**. You can use your local server as the distributor or use another server as the distributor. Click **Next**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

NOTICE

- **sa** is the administrator account.
- The login account must have the **sysadmin** permission. Otherwise, publication and subscription cannot be configured.

b. Specify a root snapshot folder and click **Next**.

NOTE

Related agent permissions must be configured for the publication so that the agent account has the permissions to read from and write to the folder.

c. Specify the names and directories of the distribution database and log files, and click **Next**.

d. Specify the distributor for the publisher and then click **Next**.

e. Click **Finish** to complete the configuration.

- Configuring the agent account control file

  a. You need to add the agent account to the control property of the snapshot folder according to the folder directory. Otherwise, an error message will be displayed, indicating that the access is denied.

  b. Open the local SQL Server Configuration Manager, right-click the corresponding agent, choose **Properties** from the shortcut menu, and copy the account name.

  c. Return to the directory of the snapshot folder. Right-click the folder and choose **Properties** from the shortcut menu. In the displayed dialog box, choose **Security** > **Edit** > **Add**. Select the local path and agent account name, click **OK**, and select all permissions.

- Distributor: distributes data to specific subscribers. In this example, the distributor and publisher share the same server. Therefore, no extra configuration is required. For more information, see **Configure Distribution** at the official website.
- Subscriber: receives data from the distributor. Subscription includes push subscription and pull subscription.
  - Push subscription: The publisher propagates changes to a subscriber without a request from the subscriber. Changes can be pushed to subscribers continuously or on a frequently recurring schedule.
  - Pull subscription: The subscriber requests changes made at the publisher. The data is usually synchronized on demand or on a schedule rather than continuously. RDS for SQL Server instances do not support pull subscription. In this example, only push subscription can be configured.

  Before the subscription, ensure that the RDS for SQL Server instance can be accessed from the local server.

  Before configuring the local subscription, you need to configure the RDS instance information on the local server.

  a. Configure an alias name for the subscriber on the local server. The subscription service does not support IP address-based access. Therefore, you need to map the EIP of the RDS DB instance to an alias name. To obtain the alias name, log in to the RDS DB instance and run the following SQL statement:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
select  @@SERVERNAME
```

b. After obtaining the alias name, open the local SQL Server Configuration Manager, select the native clients, right-click **Aliases**, and choose **New Aliases** from the shortcut menu.

c. Enter related information and click **OK**.

Table 9-1 Parameter description

Parameter	Description
Alias Name	Alias name configured in **a**
Port No	Port number of the RDS instance
Server	EIP bound to the RDS instance

d. In **C:\Windows\System32\drivers\etc**, open the host file and add a mapping:

```
Server_address MSSQL-177FFD84\MSSQL2014STD
```

Creating a Publication

Step 1 Create a publication.

Expand the **Replication** folder, and then right-click the **Local Publications** folder. Click **New Publication**.

Step 2 Select **Transactional publication**.

Step 3 Select a table as the publication object.

Step 4 Add the object to be filtered for personalized publication.

Step 5 Create a snapshot to replicate the current state of the table. You can also set up a snapshot agent to execute the plan.

Step 6 Configure the agent security. You need to set the login account to the local **sa** account.

Step 7 Configure the publication name and click **Finish**.

Step 8 Check whether the publication is created by using the replication monitor.

----End

Creating a Subscription

Step 1 Right-click the publication for which you want to create one or more subscriptions, and then select **New Subscriptions**.

Step 2 Configure the required parameters and click **Next**.

Step 3 Select push subscription and click **Next**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Step 4 Click **Add Subscriber**. Both the SQL Server engine and non-SQL Server engine can be used as subscribers. In this example, use the RDS for SQL Server instance as the subscriber.

Step 5 Select a database as the subscription object.

Step 6 Configure the connection to the subscriber.

Step 7 Use a database account that is valid for a long time to ensure the subscription validity. You can use the account for logging in to the RDS for SQL Server instance. Then, click **OK**.

Step 8 Check whether the subscription is created successfully.

Step 9 Move the cursor to the publication to view the subscription information.

----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

10 Installing a C# CLR Assembly in RDS for SQL Server

Microsoft SQL Server provides assemblies to make database operations simple and convenient.

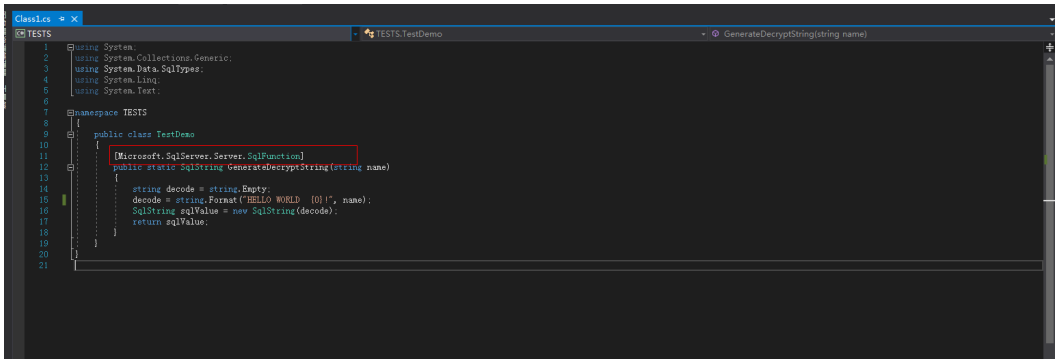
NOTE

When you restore data to a new or an existing DB instance, the **clr enabled** parameter is disabled by default. To use the CLR integration function, you need to enable **clr enabled** first. For details about how to enable the CLR integration function, see **Enabling CLR Integration**.

Procedure

Step 1 Create a C# function to compile a SQL Server DLL.

Figure 10-1 C# function code



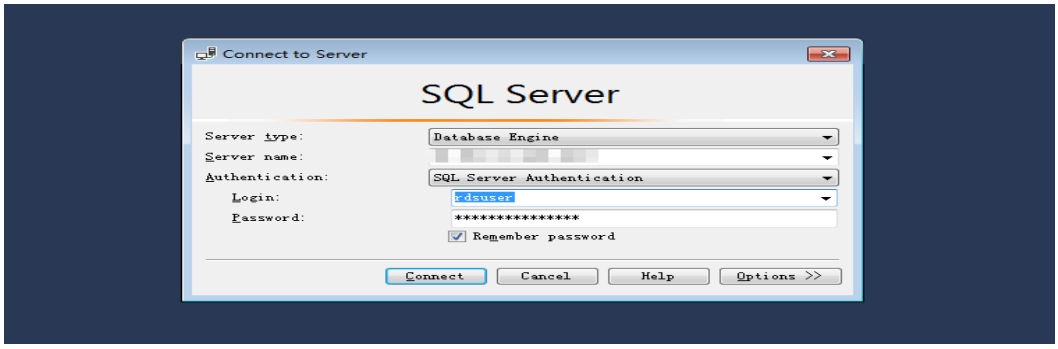
NOTICE

For more information about user-defined functions, see **CLR user-defined functions**.

Step 2 Use SQL Server Management Studio to connect to the database.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 10-2 Connecting to the server



Step 3 Select the target database and create the corresponding assembly.

NOTE

- Only the SAFE assembly (**Permission** set is **Safe**) can be created.
- The DLL file is saved in the hexadecimal format, as shown in **Figure 10-4**.

Figure 10-3 Creating an assembly

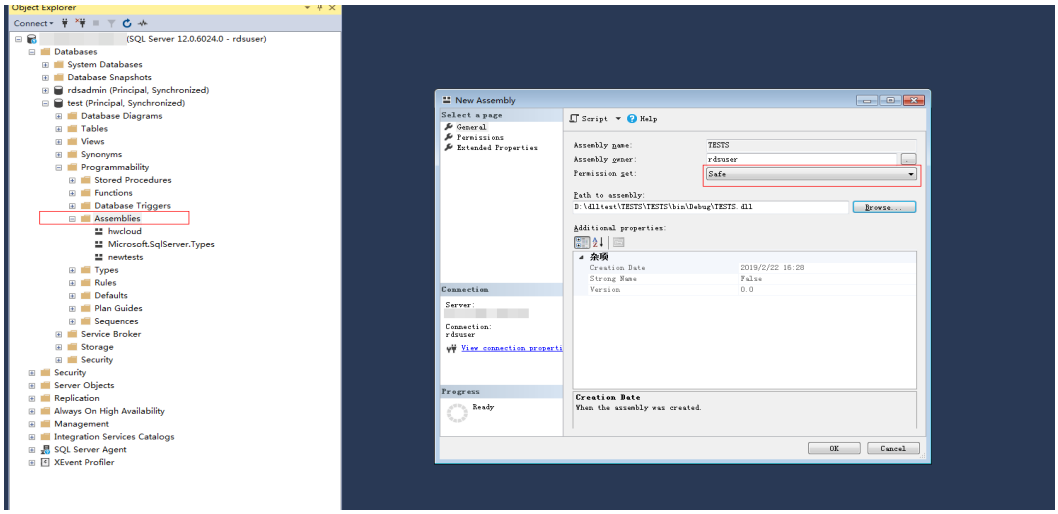
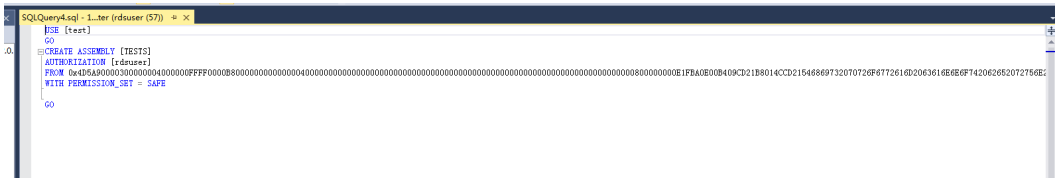


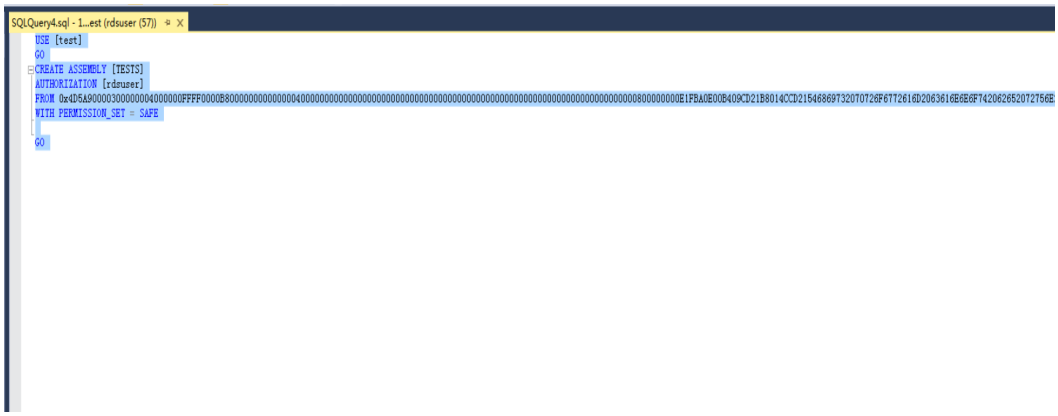
Figure 10-4 DLL file



Step 4 Execute the program. If the execution result is shown as **Figure 10-5**, the execution is successful. The TESTS assembly is added, as shown in **Figure 10-6**.

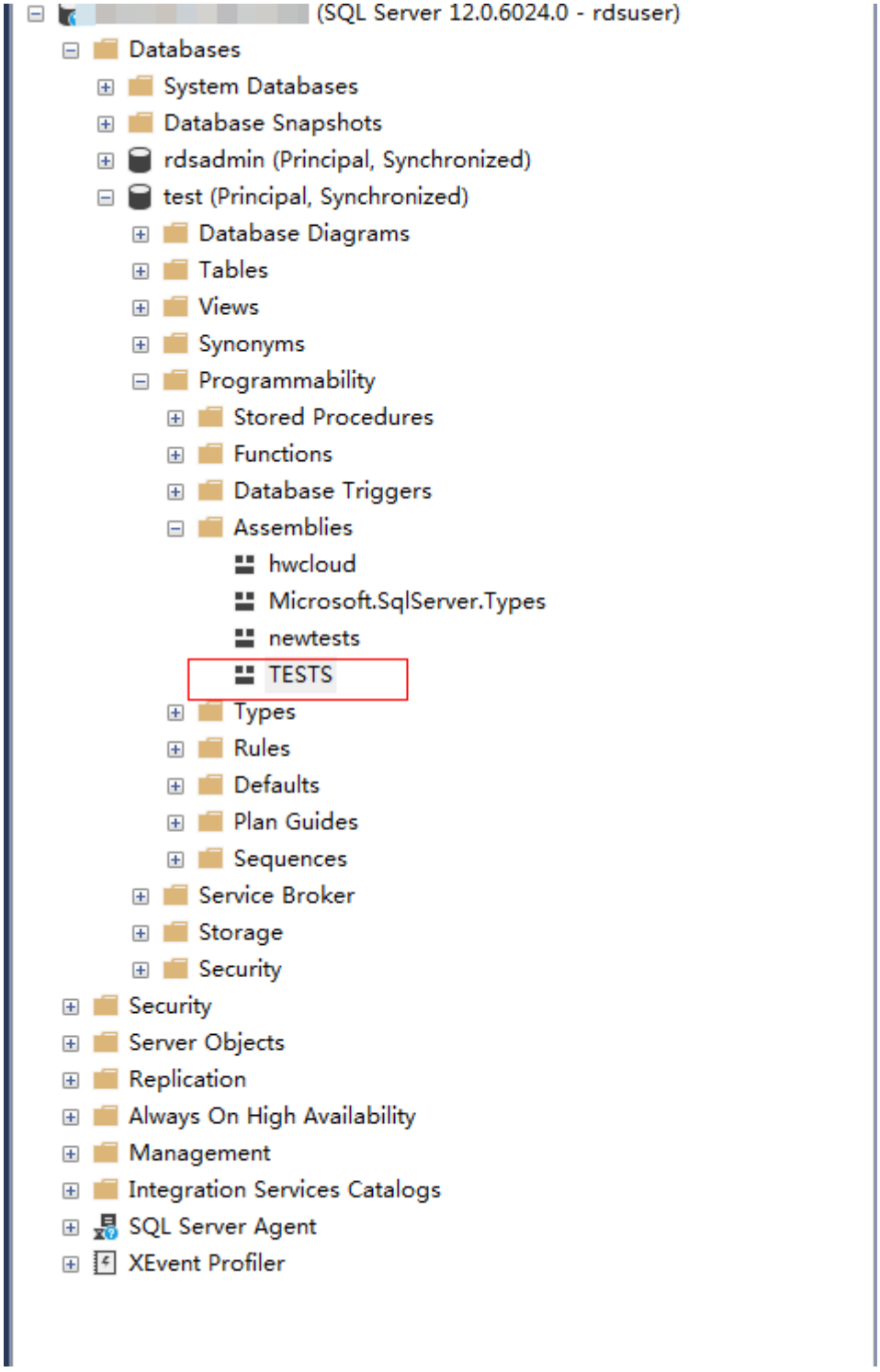
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 10-5 Execution result



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 10-6 TESTS assembly



----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

11 Creating a Linked Server for an RDS for SQL Server DB Instance

Create a linked server for RDS for SQL Server DB instance named 2 to access another RDS for SQL Server DB instance named 1.

Step 1 Enable distributed transactions of the two DB instances by referring to **Distributed Transactions** and add the peer-end host information to each other. For on-premises servers or ECSs, see **Resolving Names on Remote Servers (ECSs)**.

NOTE

RDS for SQL Server instance 2 and RDS for SQL Server instance 1 are in the same VPC. If the ECS and RDS instances are not in the same VPC or you are creating an on-premises server for an RDS instance, bind an EIP to the RDS instance. For details, see **Binding and Unbinding an EIP**.

Step 2 In DB instance 1, create database dbtest1 as user **rdsuser**.

Step 3 In DB instance 2, run the following SQL statements to create a linked server as user **rdsuser**:

```
USE [master]
GO
EXEC master.dbo.sp_addlinkedserver @server = N'TEST_SERVERNAME', @srvproduct=N'SQLServer',
@provider=N'SQLOLEDB', @datasrc=N'192.168.***.***,1433'
EXEC master.dbo.sp_addlinkedsrvlogin @rmtsrvname = N'TEST_SERVERNAME', @locallogin = NULL ,
@useself = N'false', @rmtuser = N'rdsuser', @rmtpassword = N'********'
GO
```

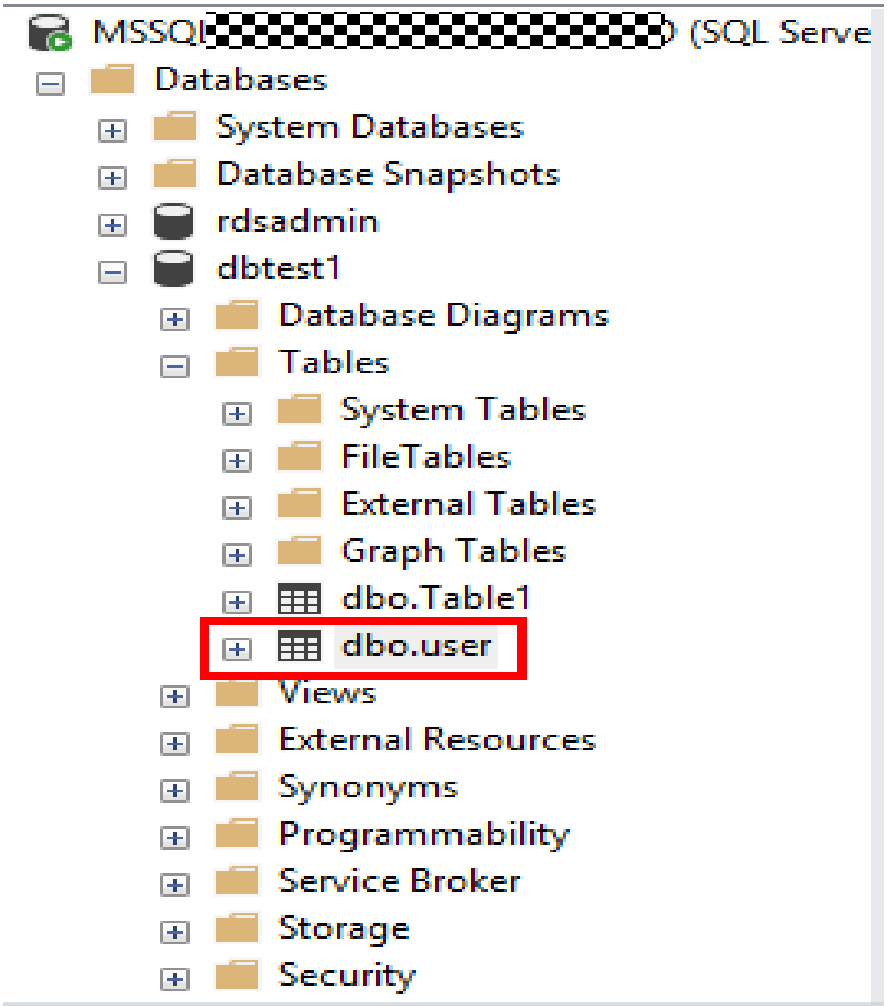
Table 11-1 Parameter description

Parameter	Description
@server	The name of the linked server.
@srvproduct	The product name of the data source. Use the default value **SQL Server**.
@provider	Use the default value.
@datasrc	The IP address and port of the DB instance to be accessed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Description
@rmtsrvname	The name of the linked server.
@locallogin	The login name on the local server. Use the default value **NULL**.
@useself	Whether to connect to the linked server by simulating the local login name or username and password. Enter **false**, which indicates that the linked server is connected through the username and password.
@rmtuser	The username (**rdsuser**).
@rmtpassword	The password for the username.

Step 4 After the DBLink is created, you can view the databases created in DB instance 1 on the linked server.

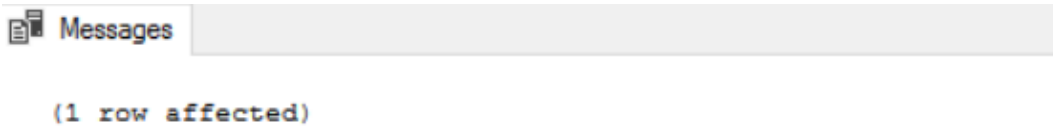


Step 5 Run the following commands to check whether the data is successfully inserted, as shown in **Figure 11-1**:

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
begin tran
set xact_abort on
INSERT INTO [LYNTEST].[dbtest1].[dbo].[user1]
([id],[lname],[rname])
VALUES('19','w','x')
GO
commit tran
```

Figure 11-1 Insert result



----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

12 Deploying SQL Server Reporting Services (SSRS) on RDS for SQL Server

You can use SSRS to make various simple or complex reports. In addition, RDS provides the subscription function for you to subscribe to reports. This section describes how to deploy SSRS on RDS for SQL Server.

Scenarios

Microsoft SQL Server contains server components such as the SQL Server database engine, SSRS, and SQL Server Analysis Services (SSAS). The SQL Server database engine is a standard relational database component. RDS for SQL Server is a Platform-as-a-Service (PaaS) service that provides this database engine. However, other components, such as SSRS, are not provided as PaaS services on Huawei Cloud. To use SSRS on Huawei Cloud, you need to create a Windows-based ECS before installing and configuring SSRS.

SSRS has been separated from the Microsoft SQL Server component package and become an independent component service since SQL Server 2017. To migrate SSRS to the cloud, download the component from the Microsoft official website, install it on a Windows-based ECS, and use RDS for SQL Server as the backend database.

Prerequisites

- You have **created an RDS for SQL Server instance**.
- You have created a Windows-based ECS. (The ECS and RDS DB instance must be in the same VPC, security group, and subnet.)

Procedure

Step 1 Download **SSRS** and install it on the ECS.

Step 2 After the installation is complete, click **Configure Report Server**.

Step 3 In Report Server Configuration Manager, configure **Server Name** and click **Connect**.

Step 4 In the navigation pane on the left, click **Service Account** and **Web Service URL** and configure parameters based on your service requirements.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

NOTE

For details, see the **official documentation**.

Step 5 Configure the report server.

1. In the navigation pane on the left, click **Database**. On the right of the page, click **Change Database** to create a report server database on the ECS.
2. In the displayed dialog box, select **Create a new report server database** and click **Next**.

   If a local report database is available, you can use Data Replication Service (DRS) to **migrate the full backup files** of the local report database to the RDS for SQL Server instance.
3. Configure the connection information of the RDS for SQL Server instance. Set **Server Name** to the RDS for SQL Server instance address in the format of *IP address,port*. Use a comma (,) to separate the IP address and port. Set **Username** to **rdsuser**. Click **Test Connection**. After the connection test is successful, click **Next**.
4. Enter the database name, select a language for the script, and then click **Next**.
5. Configure the credentials for the account **rdsuser** to connect to the report server and click **Next**.
6. Confirm the report server information and click **Next**.
7. After the configuration is successful, click **Finish**.

NOTE

For details, see the **official documentation**.

Step 6 In the navigation pane on the left, click **Web Portal URL** and click **Apply**. After the operation is complete, click the URL to access the web page of the report server.

Step 7 In the upper right corner, choose **New** > **Data Source**.

Step 8 Configure the parameters as follows:

Table 12-1 Parameter description

Category	Parameter	Description
Properties	Name	Name of the data source. The name cannot contain the following characters: / @ $ & * + = < > : ' , ? | \
	Description	Description of the data source, which is used to identify different data sources.
	Hide	If this parameter is selected, the data source is hidden.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Category	Parameter	Description
	Enable	If this parameter is selected, the data source is enabled.
Connections	Type	Type of the data source. Select **Microsoft SQL Server**.
	Connection String	Domain name and database name of the RDS for SQL Server instance in the following format: Data Source=<*Floating IP address of the RDS for SQL Server instance*, *port of the RDS for SQL Server instance*>; Initial Catalog=<*Database name*>
Login	Data Source Login	Select **Use the following credentials**.
	Credential Type	Select **Database username and password**.
	Username	Account of the RDS for SQL Server instance
	Password	Password of the database account

Step 9 Click **Test Connection**. After the connection test is successful, click **Create**.

Step 10 After the data source is created, design reports using Report Builder or Visual Studio.

For details, see **Report Builder in SQL Server**.

----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

13 Shrinking an RDS for SQL Server Database

Scenarios

You can use a stored procedure to shrink the size of the data and log files in a specified RDS for SQL Server database.

Prerequisites

An RDS for SQL Server DB instance has been connected. You can connect to the DB instance through a SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**

Constraints

- The database can be shrunk only when the database file size exceeds 50 MB. Otherwise, the following message is displayed:

  ```
  Cannot shrink file '2' in database 'master' to 6400 pages as it only contains 256 pages.
  ```

- Transactions running at the row version control-based isolation level may prevent shrinking operations. To solve this problem, perform the following steps:
  a. Terminate the transactions that prevent shrinking.
  b. Terminate the shrinking operation. All completed tasks will be retained.
  c. Do not perform any operations and wait until the blocking transactions are complete.

Best Practices

When you plan to shrink a database, consider the following:

- A shrink operation is most effective after an operation that creates lots of unused space, such as a database reboot.
- Most databases require some free space to be available for regular day-to-day operations. If you shrink a database repeatedly and notice that the database size grows again, this indicates that the space that was shrunk is required for regular operations. In these cases, repeatedly shrinking the database is a wasted operation.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- A shrink operation does not preserve the fragmentation state of indexes in the database, and generally increases fragmentation to a degree. This is another reason not to repeatedly shrink the database.

Procedure

Step 1 Run the following command to shrink a database:

EXEC [master].[dbo].[rds_shrink_database] @DBName='*myDbName***';**

Table 13-1 Parameter description

Parameter	Description
myDbName	Name of the database to be shrunk. If this parameter is not specified, all databases are shrunk by default.

The following figure shows the execution result set. Each result corresponds to the information about each file in the specified database (or all databases).

Figure 13-1 Result set

Results | Messages

	DbId	FileId	CurrentSize	MinimumSize	UsedPages	EstimatedPages
1	1	1	688	512	512	512

Table 13-2 Result set parameter description

Column Name	Description
DbId	Database ID of the current shrink file.
FileId	File ID of the current shrink file.
CurrentSize	Number of 8 KB pages occupied by the file.
MinimumSize	Minimum number of 8 KB pages occupied by the file. The value indicates the minimum size or the initial size of the file.
UsedPages	Number of 8 KB pages used by the file.
EstimatedPages	Number of 8 KB pages that the database engine estimates the file can be shrunk to.

Step 2 After the command is successfully executed, the following information is displayed:

```
HW_RDS_Process_Successful: Shrink Database Done.
```

----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Fault Rectification

If the file size does not change after the database is shrunk, run the following SQL statement to check whether the file has sufficient available space:

SELECT name, size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS int)/ 128.0 AS AvailableSpaceInMB FROM sys.database_files;

Example

1. Run the following command to shrink the **dbtest2** database:

   EXEC [master].[dbo].[rds_shrink_database] @DBName = 'dbtest2';

   The command output is as follows:

   Figure 13-2 Execution result

   ```
   [Shrink Start] Date and time: 2020-03-19 15:51:07

   Start to shrink files in database [dbtest2], current file id is 1...
   DBCC execution completed. If DBCC printed error messages, contact your system administrator.
   Shrink file (id: 1) in database [dbtest2] done!

   Start to shrink files in database [dbtest2], current file id is 2...
   DBCC execution completed. If DBCC printed error messages, contact your system administrator.
   Shrink file (id: 2) in database [dbtest2] done!

   [Shrink End] Date and time: 2020-03-19 15:51:08

   HW_RDS_Process_Successful : Shrink Database done.
   ```

2. Run the following command to shrink all databases:

   EXEC [master].[dbo].[rds_shrink_database];

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

14 Using DAS to Create and Configure Agent Job and DBLink on the Master and Slave Databases for RDS for SQL Server Instances

Scenarios

Data Admin Service (DAS) is a one-stop database management platform that allows you to manage databases on a web console. It offers database development, O&M, and intelligent diagnosis, facilitating your database usage and maintenance. Currently, DAS supports primary/standby switchover of RDS for SQL Server databases, facilitating synchronization between master and slave databases.

Logging In to DAS

Step 1 **Log in to the management console**.

Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance and click **Log In** in the **Operation** column.

Alternatively, click the instance name on the **Instances** page. On the displayed **Basic Information** page, click **Log In** in the upper right corner of the page.

Step 5 On the displayed login page, enter the correct username and password and click **Log In**.

----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Creating a Job for Data Synchronization to the Slave Database

Step 1 Create a job on the master database.

On the DAS console, choose **SQL Operations** > **SQL Query** on the top menu bar. In the msdb database, run the following commands to create a job:

NOTE

If a job has been created on the master database, skip this step.

```
USE [msdb]
GO
DECLARE @jobId BINARY(16)
EXEC  msdb.dbo.sp_add_job @job_name=N'hwtest',
		@enabled=1,
		@notify_level_eventlog=0,
		@notify_level_email=2,
		@notify_level_page=2,
		@delete_level=0,
		@category_name=N'[Uncategorized (Local)]',
		@owner_login_name=N'rdsuser', @job_id = @jobId OUTPUT
select @jobId
GO
EXEC msdb.dbo.sp_add_jobserver @job_name=N'hwtest', @server_name = N'*******'
GO
USE [msdb]
GO
EXEC msdb.dbo.sp_add_jobstep @job_name=N'hwtest', @step_name=N'select orders',
		@step_id=1,
		@cmdexec_success_code=0,
		@on_success_action=1,
		@on_fail_action=2,
		@retry_attempts=0,
		@retry_interval=0,
		@os_run_priority=0, @subsystem=N'TSQL',
		@command=N'select * from orders;',
		@database_name=N'test',
		@flags=0
GO
USE [msdb]
GO
EXEC msdb.dbo.sp_update_job @job_name=N'hwtest',
		@enabled=1,
		@start_step_id=1,
		@notify_level_eventlog=0,
		@notify_level_email=2,
		@notify_level_page=2,
		@delete_level=0,
		@description=N'',
		@category_name=N'[Uncategorized (Local)]',
		@owner_login_name=N'zf1',
		@notify_email_operator_name=N'',
		@notify_page_operator_name=N''
GO
```

Run the following SQL statements to check whether the job has been created:

```
use [msdb]

select * from msdb.dbo.sysjobs where name ='hwtest';
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Step 2 Switch to the slave database.

NOTE

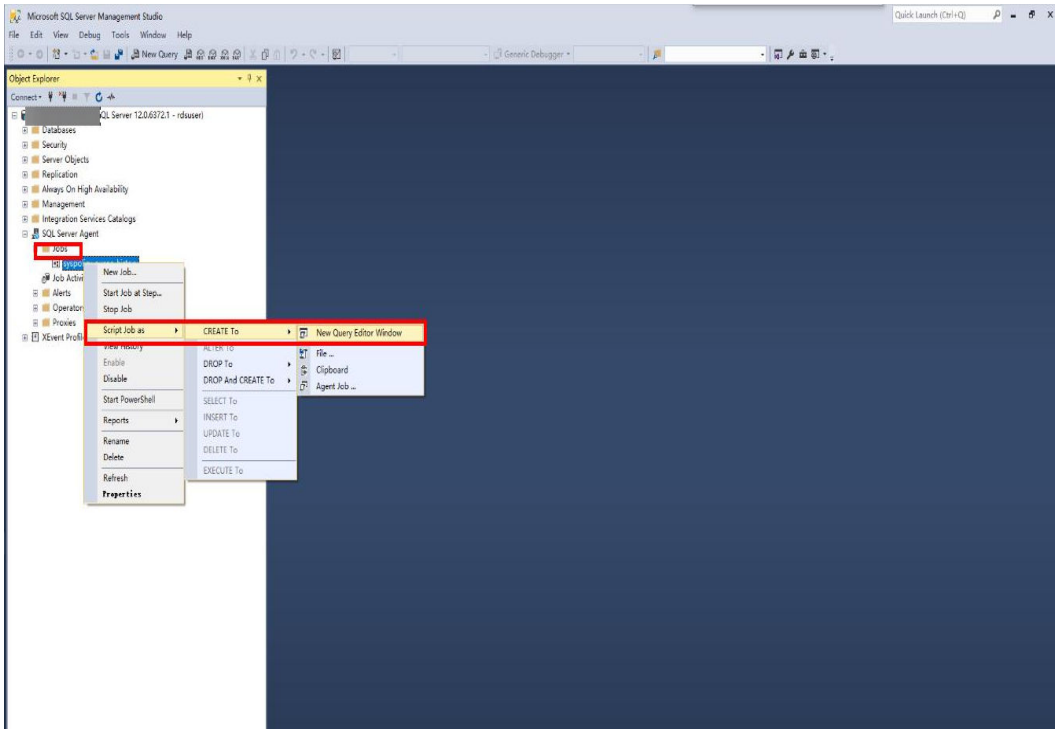
Currently, RDS for SQL Server does not support job synchronization between the master and slave databases. Therefore, you need to create a job on the slave database and synchronize the job. Click **Switch SQL Execution Node** next to **Master** to switch to the slave database.

Step 3 Run the commands in **Step 1** to create a job on the slave database.

Alternatively, use SQL Server Management Studio (SSMS) to export the created job to the editor window, copy the job to the SQL query window, and then run the commands in **Step 1** to create a job on the slave database.

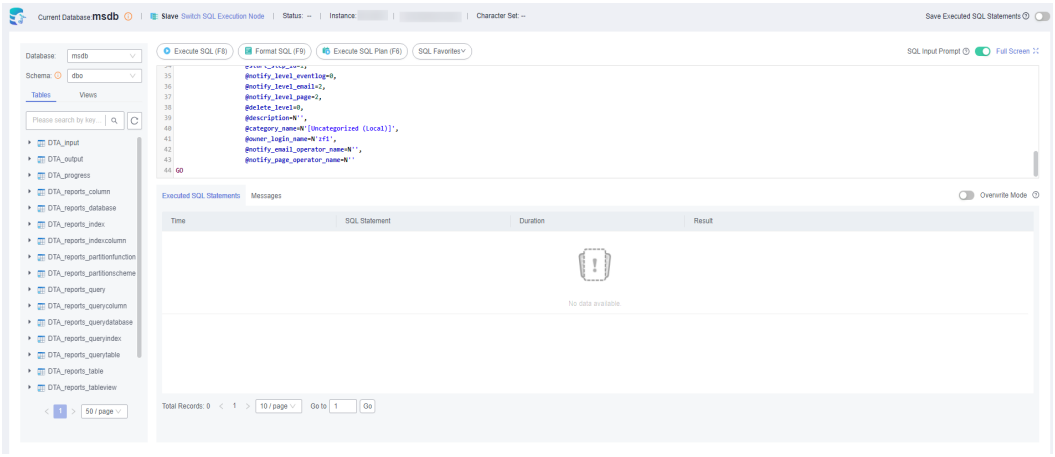
If the job fails to be created, you are advised to delete the job first and then create the job again.

Figure 14-1 Exporting a job



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 14-2 Using the DAS console to create a job on the slave database



Run the following SQL statements to delete the job:

USE [msdb]

GO

EXEC msdb.dbo.sp_delete_job @job_name=N'hwtest', @delete_unused_schedule=1

GO

----End

Creating a DBLink for Data Synchronization to the Slave Database

DAS allows you to create linked servers to synchronize data between master and slave databases.

NOTE

Check whether the MSDTC is configured by referring to **Creating a Linked Server for an RDS for SQL Server DB Instance**.

Step 1 Create a DBLink on the master database.

USE [master]

GO

EXEC master.dbo.sp_addlinkedserver @server = N'TEST', @srvproduct=N'mytest', @provider=N'SQLOLEDB', @datasrc=N'abcd'

EXEC master.dbo.sp_addlinkedsrvlogin @rmtsrvname = N'TEST', @locallogin = NULL , @useself = N'False', @rmtuser = N'rdsuser', @rmtpassword = N'******'**

GO

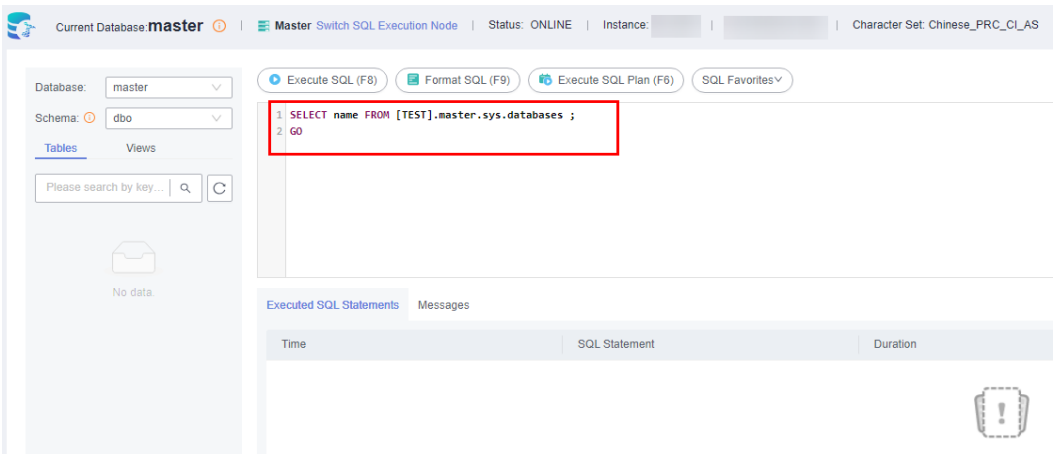
After the creation is successful, the corresponding DB instance or databases can be connected to view data verification. Run the following statements to query databases:

SELECT name FROM [TEST].master.sys.databases ;

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
GO
```

Figure 14-3 Database query



Step 2 Create a DBLink on the slave database.

On the DAS console, click **Switch SQL Execution Node** next to **Master** and run the SQL statement for creating a DBLink.

NOTE

If the current DB instance and the interconnected database are not in the same VPC or distributed transactions are enabled with an EIP bound to the primary DB instance, the query statement cannot be executed on the standby DB instance. This step is used only to synchronize the DBLink configuration. After a switchover or failover, the DBLink can be used.

----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

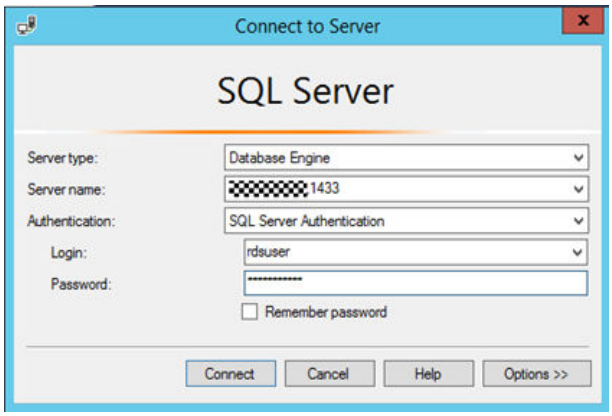
15 Creating a Job for Scheduled Instance Maintenance

Scenarios

After a DB instance runs for a period of time, the system performance deteriorates because index fragments increase and statistics are not updated in a timely manner. You are advised to create a SQL agent job to periodically re-create indexes, update statistics, and shrink the database.

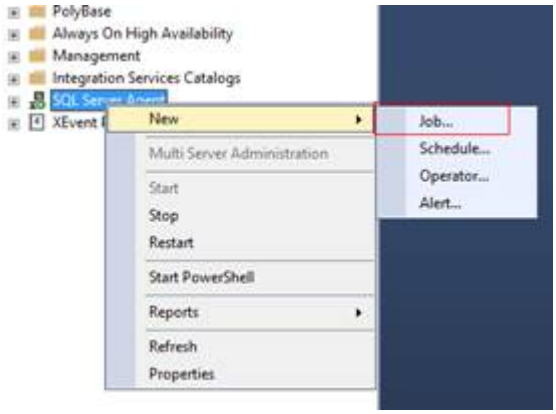
Creating an Index Re-creating Job

Step 1 Start the SQL Server Management Studio client and log in to it as user **rdsuser**.

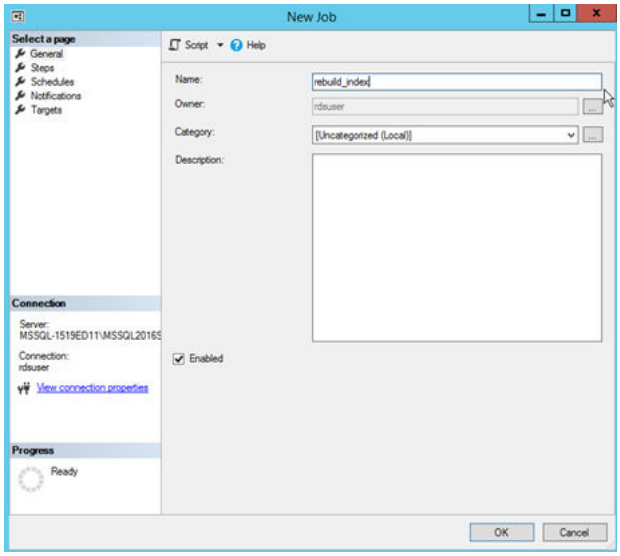


Step 2 Right-click **SQL Server Agent** and choose **New** > **Job** to create an SQL agent job.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

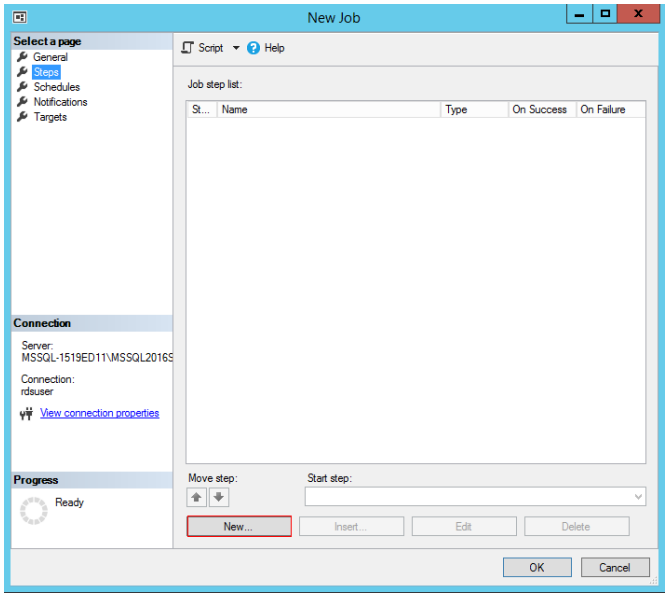


Step 3 Enter the name and description, and click **OK**.



Step 4 Select **Steps** and click **New** to add an execution step.

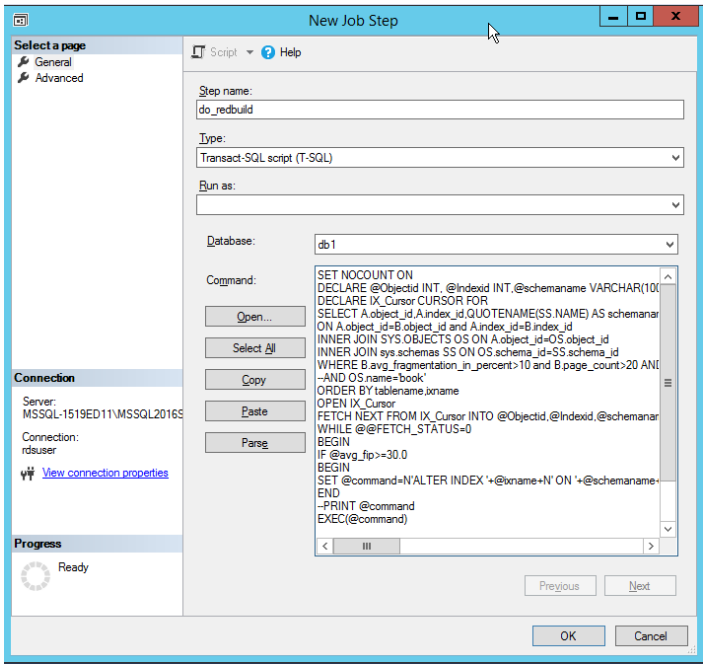
Figure 15-1 Adding an execution step



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Step 5 Enter the step name, type, and command, and click **OK**. Set **Command** to the SQL statements that need to be executed periodically. When the number of index fragments reaches a specified value, for example, 30%, the index can be recreated.

Figure 15-2 Configure parameters



Run the following SQL statement to recreate the index because the number of index fragments of all tables in the specified **dbname** exceeds 30%:

```
use [dbname]
SET NOCOUNT ON
DECLARE @Objectid INT, @Indexid INT,@schemaname VARCHAR(100),@tablename
VARCHAR(300),@ixname VARCHAR(500),@avg_fip float,@command VARCHAR(4000)
DECLARE IX_Cursor CURSOR FOR
SELECT A.object_id,A.index_id,QUOTENAME(SS.name) AS
schemaname,QUOTENAME(OBJECT_NAME(B.object_id,B.database_id))as
tablename ,QUOTENAME(A.name) AS ixname,B.avg_fragmentation_in_percent AS avg_fip FROM
sys.indexes A inner join sys.dm_db_index_physical_stats(DB_ID(),NULL,NULL,NULL,'LIMITED') AS B
ON A.object_id=B.object_id and A.index_id=B.index_id
INNER JOIN sys.objects OS ON A.object_id=OS.object_id
INNER JOIN sys.schemas SS ON OS.schema_id=SS.schema_id
WHERE B.avg_fragmentation_in_percent>10 and B.page_count>20 AND A.index_id>0 AND A.is_disabled<>1
--AND OS.name='book'
ORDER BY tablename,ixname
OPEN IX_Cursor
FETCH NEXT FROM IX_Cursor INTO @Objectid,@Indexid,@schemaname,@tablename,@ixname,@avg_fip
WHILE @@FETCH_STATUS=0
BEGIN
IF @avg_fip>=30.0
BEGIN
SET @command=N'ALTER INDEX '+@ixname+N' ON '+@schemaname+N'.'+ @tablename+N' REBUILD ';
END
--PRINT @command
EXEC(@command)
FETCH NEXT FROM IX_Cursor INTO @Objectid,@Indexid,@schemaname,@tablename,@ixname,@avg_fip
END
CLOSE IX_Cursor
DEALLOCATE IX_Cursor
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

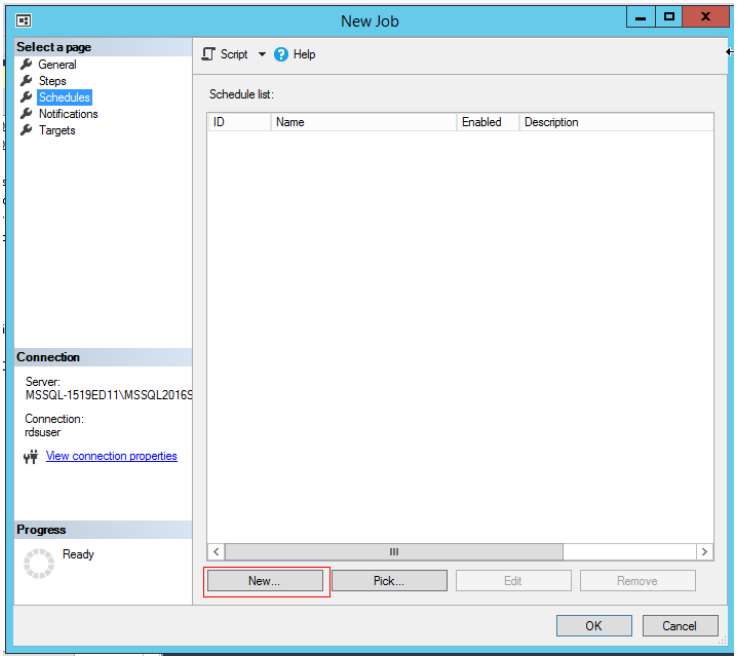
NOTE

In the preceding SQL statements, you only need to change the value of Use [**dbname**] in the first line to the specified database name.

If you need to execute the SQL statements for all databases, modify the SQL statements to add cyclic execution for all databases.

Step 6 Select **Schedules** and click **New** to add a scheduled execution plan.

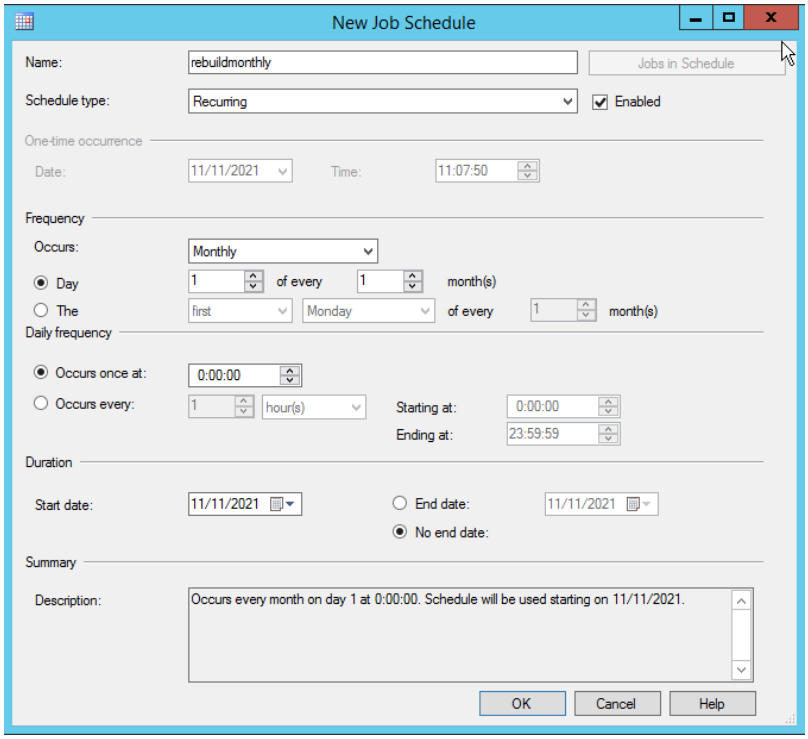
Figure 15-3 Adding a scheduled execution plan



Step 7 Add a schedule that is executed once a month, modify the daily frequency and duration, and click **OK**.

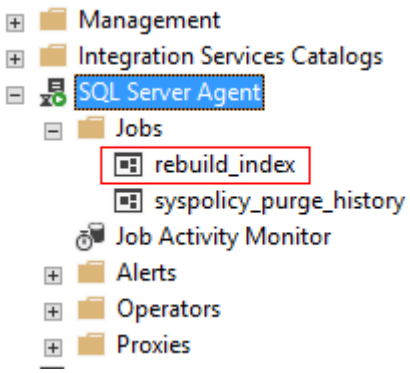
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 15-4 Configuring a scheduled execution plan



Step 8 Check that the job has been created.

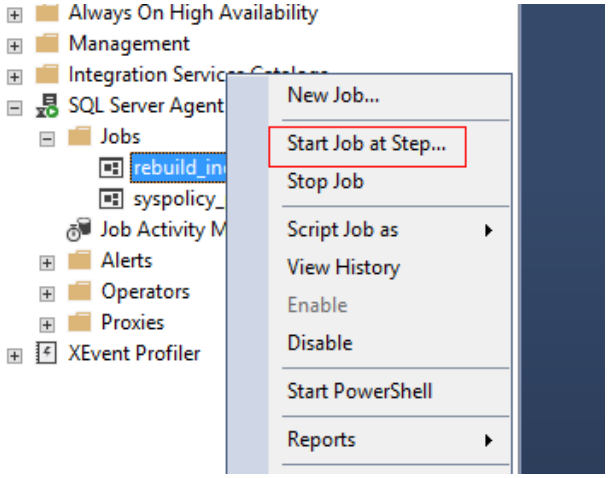
Figure 15-5 job



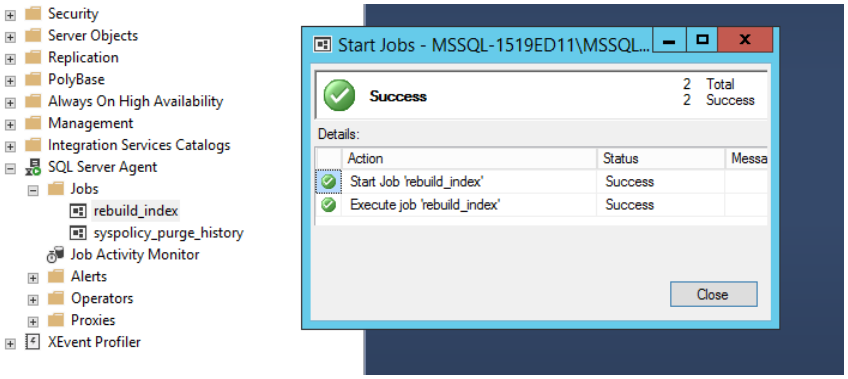
Step 9 Right-click the job and choose **Start Job at Step** to manually run the job.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 15-6 Running a job



Step 10 Check whether the job can run properly. If the job runs normally, the maintenance job for periodically recreating the indexes of the **db1** database has been created.



----End

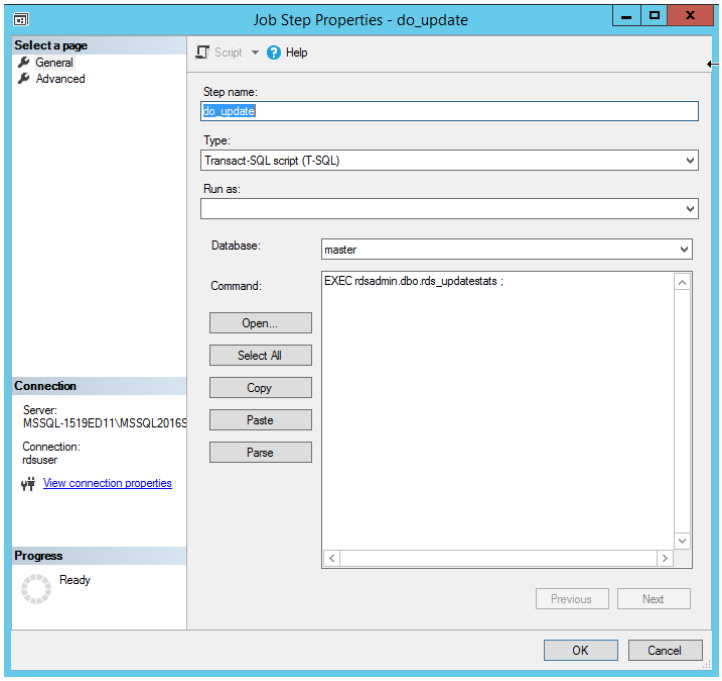
Updating Statistics

Step 1 Perform **Step 1** to **Step 4**.

Step 2 Enter the step name, type, and command, and click **OK**. Set **Command** to the stored procedure for updating statistics. For details, see **Updating Database Statistics**.

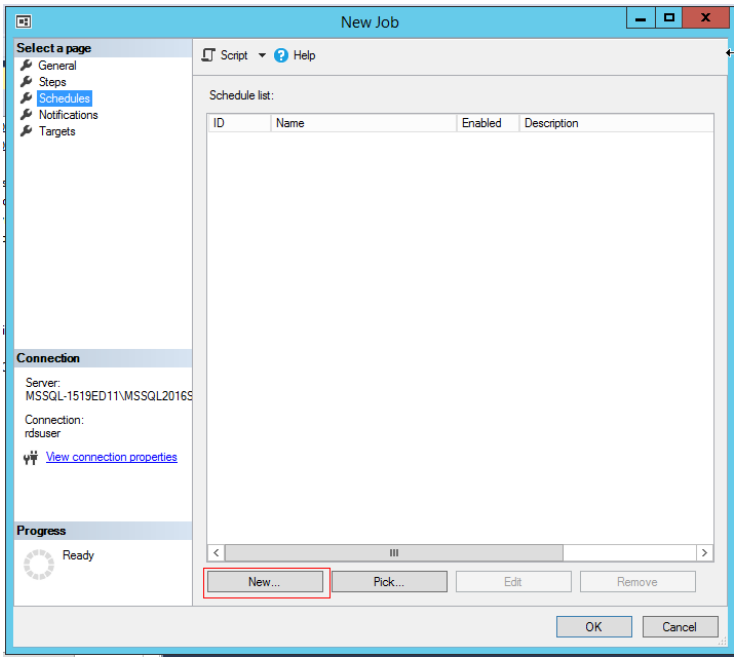
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 15-7 Updating statistics



Step 3 Select **Schedules** and click **New** to add a scheduled execution plan.

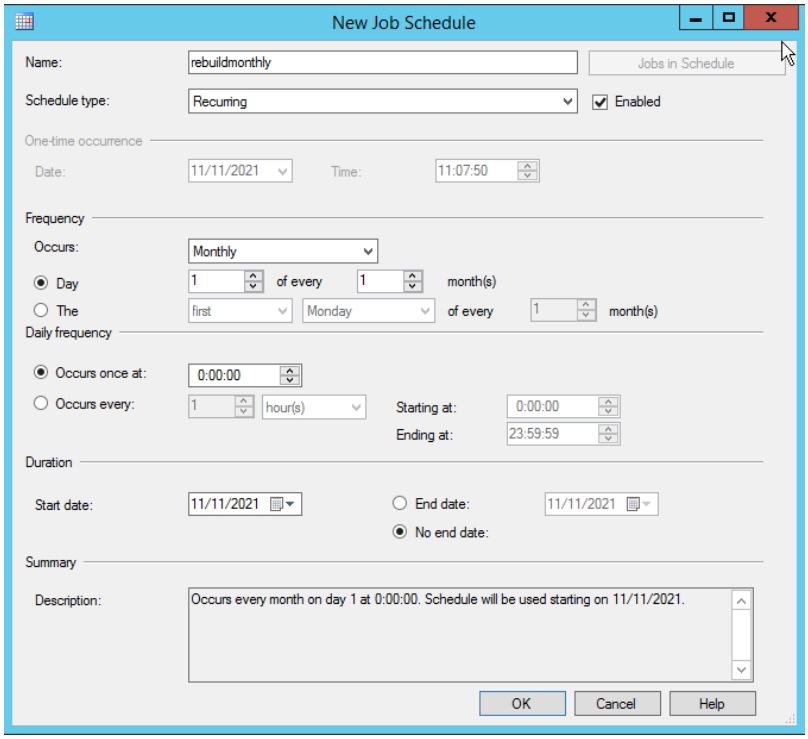
Figure 15-8 Adding a scheduled execution plan



Step 4 Add a schedule that is executed once a month, modify the daily frequency and duration, and click **OK**.

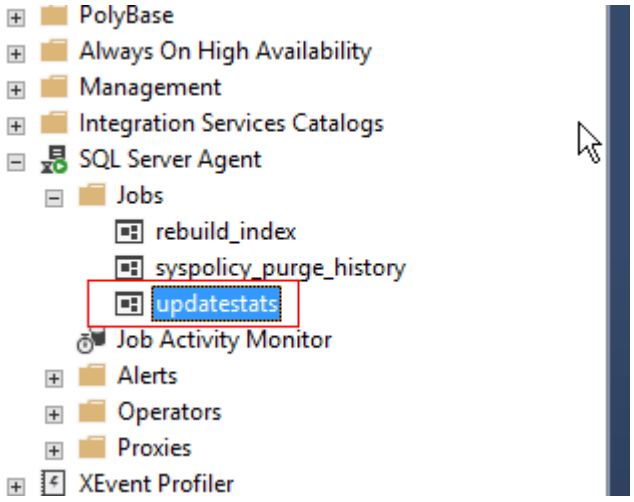
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 15-9 Configuring a scheduled execution plan



Step 5 Check that the job has been created.

Figure 15-10 Updating statistics job



Step 6 Right-click the job and choose **Start Job at Step** to manually run the job.

----End

Shrinking the Database Periodically

Step 1 Perform **Step 1** to **Step 4**.

Step 2 Enter the step name, type, and command, and click **OK**. Set **Command** to the SQL commands for shrinking the database.

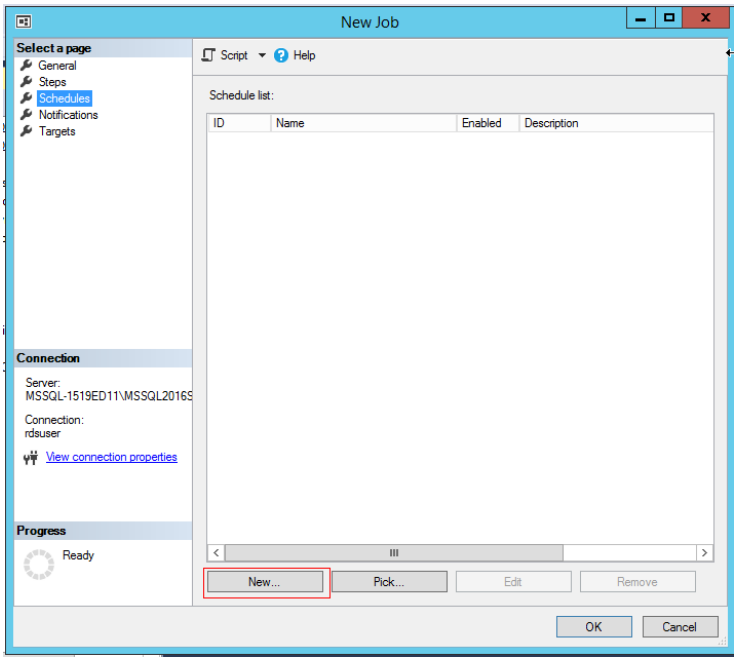
```
EXEC [master].[dbo].[rds_shrink_database_log] @dbname='myDbName';
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Set **@dbname** to the database name.

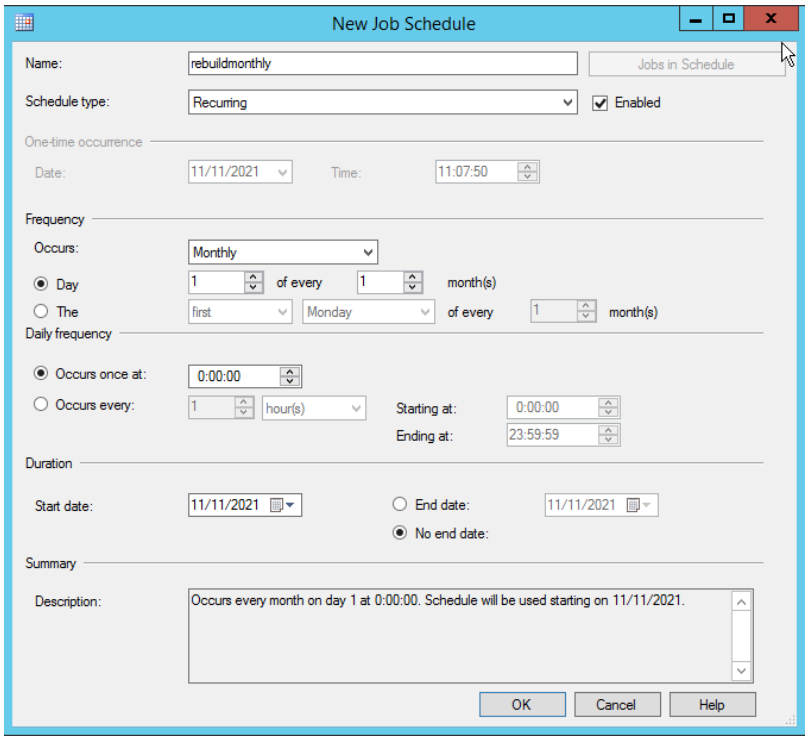
Step 3 Select **Schedules** and click **New** to add a scheduled execution plan.

Figure 15-11 Adding a scheduled execution plan



Step 4 Add a schedule that is executed once a month, modify the daily frequency and duration, and click **OK**.

Figure 15-12 Configuring a scheduled execution plan



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Step 5 Right-click the job and choose **Start Job at Step** to manually run the job.

----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

16 Using Extended Events

Extended event permissions are available now. You can use **rdsuser** to manage extended events or grant extended event permissions to other users.

For more information, see **Quickstart: Extended Events in SQL Server**.

Constraints

- All RDS for SQL Server 2008 versions do not support extended events because Microsoft SQL Server 2008 does not support extended events.
- The etw_classic_sync_target type is not available for extended event targets.
- When an extended event is created or updated, only the path **D:\RDSDBDATA\Log\error** is supported. The file name can be customized.

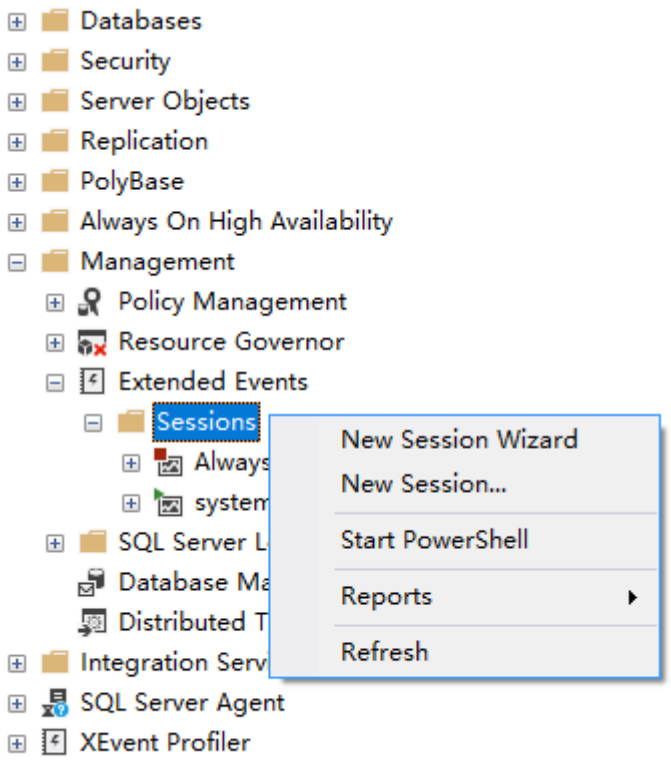
Creating an Extended Event

Step 1 Start the SQL Server Management Studio (SSMS) client and log in to it as user **rdsuser**.

Step 2 Choose **Management** > **Sessions** > **New Session**.

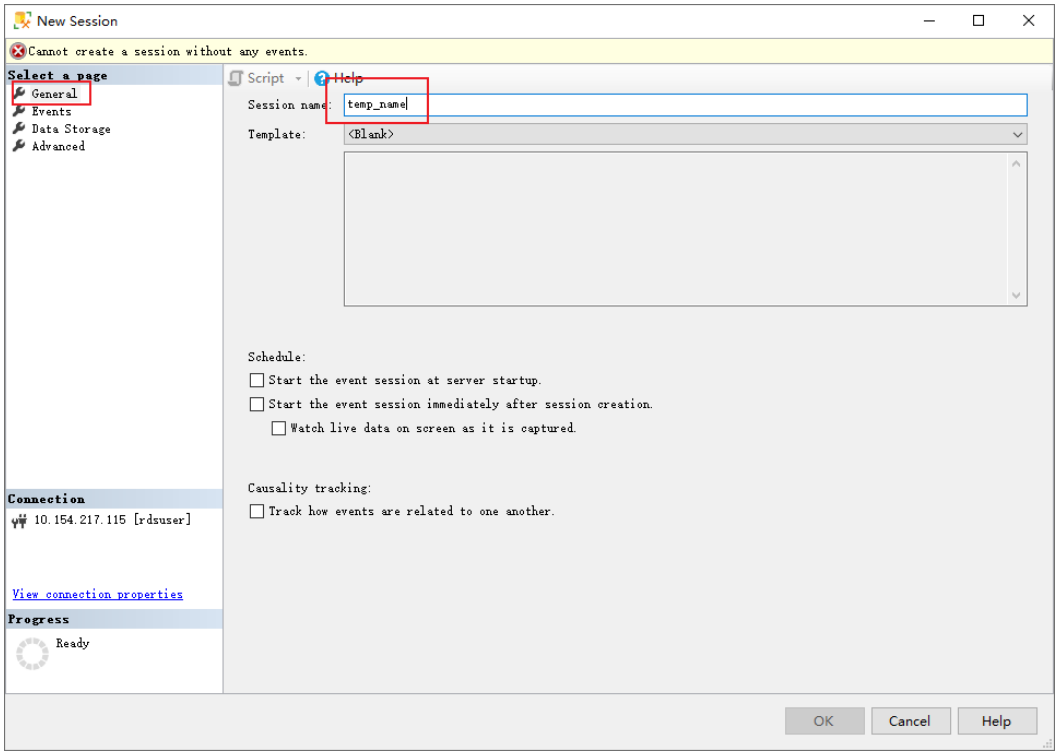
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 16-1 Creating an extended event



Step 3 Click **General** and enter a session name.

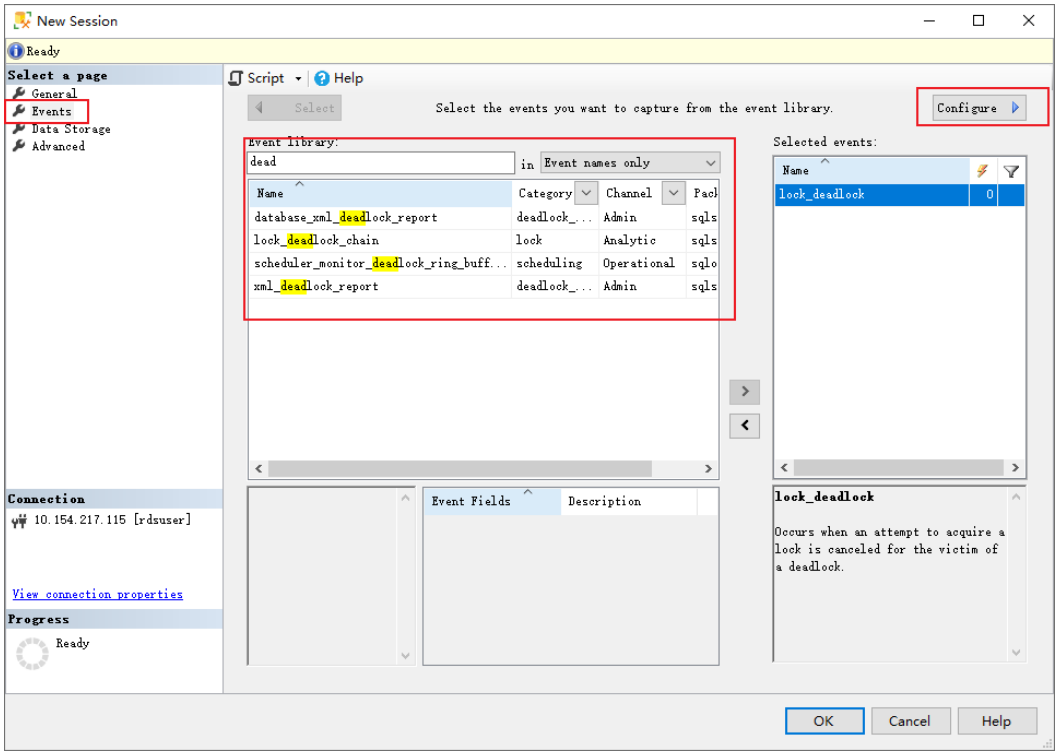
Figure 16-2 Entering a session name



 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

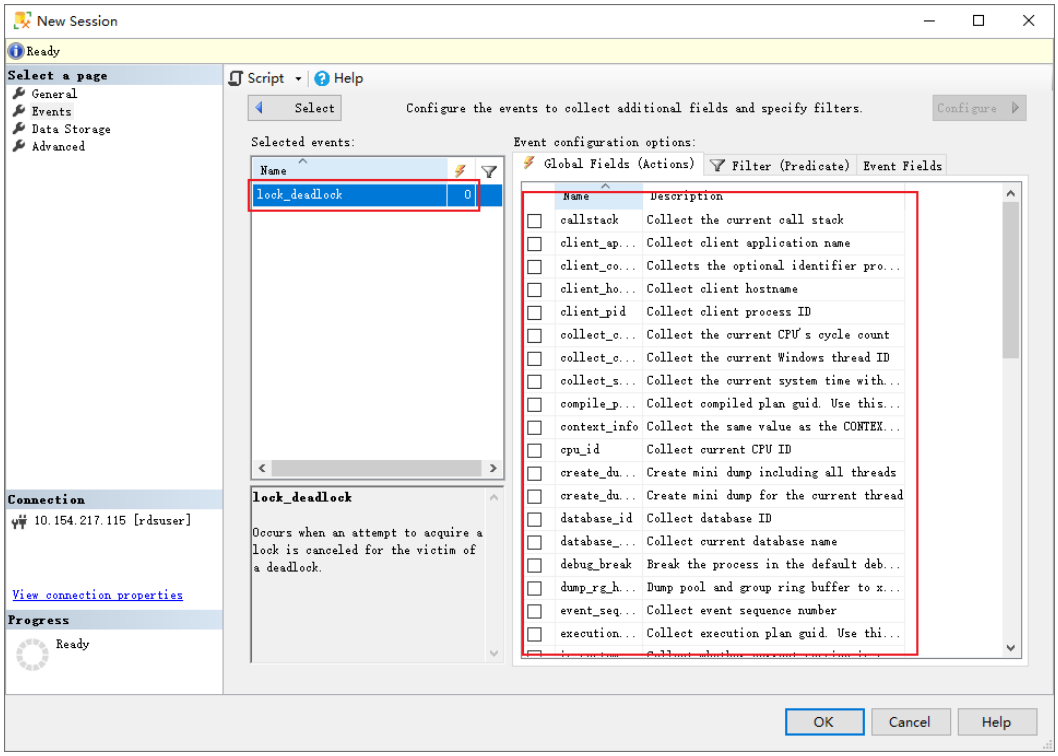
Step 4 Click **Events** and select an event.

Figure 16-3 Selecting an event



Step 5 Click **Configure** on the page displayed in **Step 4**.

Figure 16-4 Configuring an event



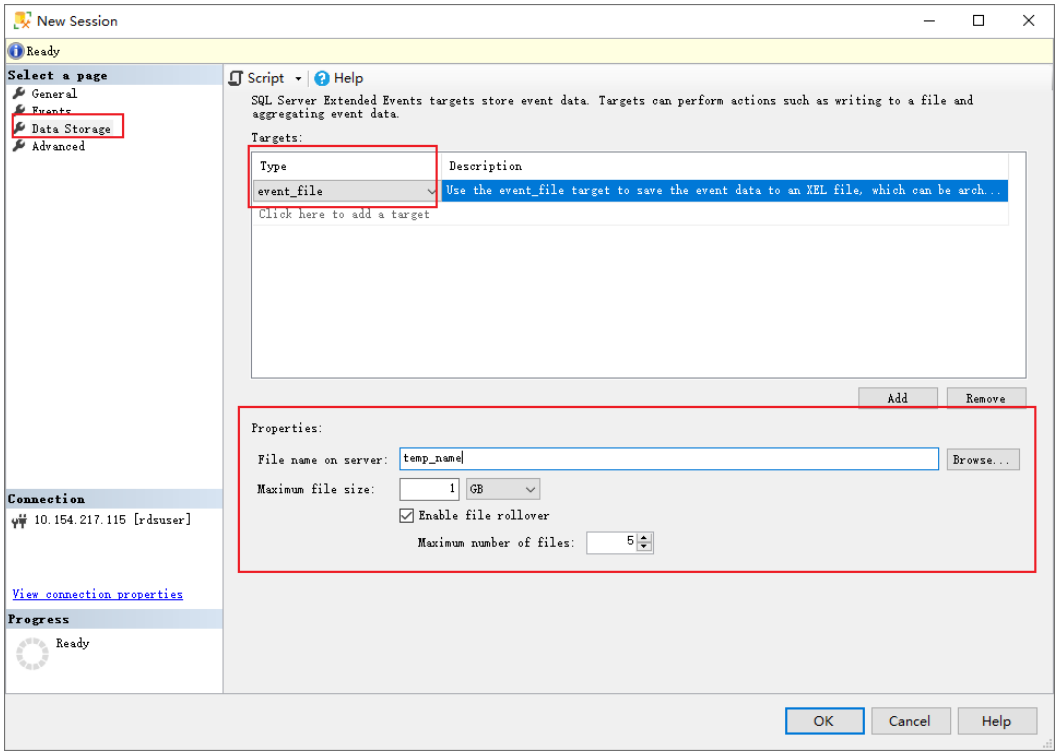
 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Step 6 Click **Data Storage** to configure the data storage.

NOTE

The file name can be customized. There is no need to click **Browse** because you can only browse the file system of the client where the SSMS is located, but not the file system of the RDS for SQL Server server. RDS for SQL Server supports the **D:\RDSDBDATA\Log\error** path only, so you only need to change the file name.

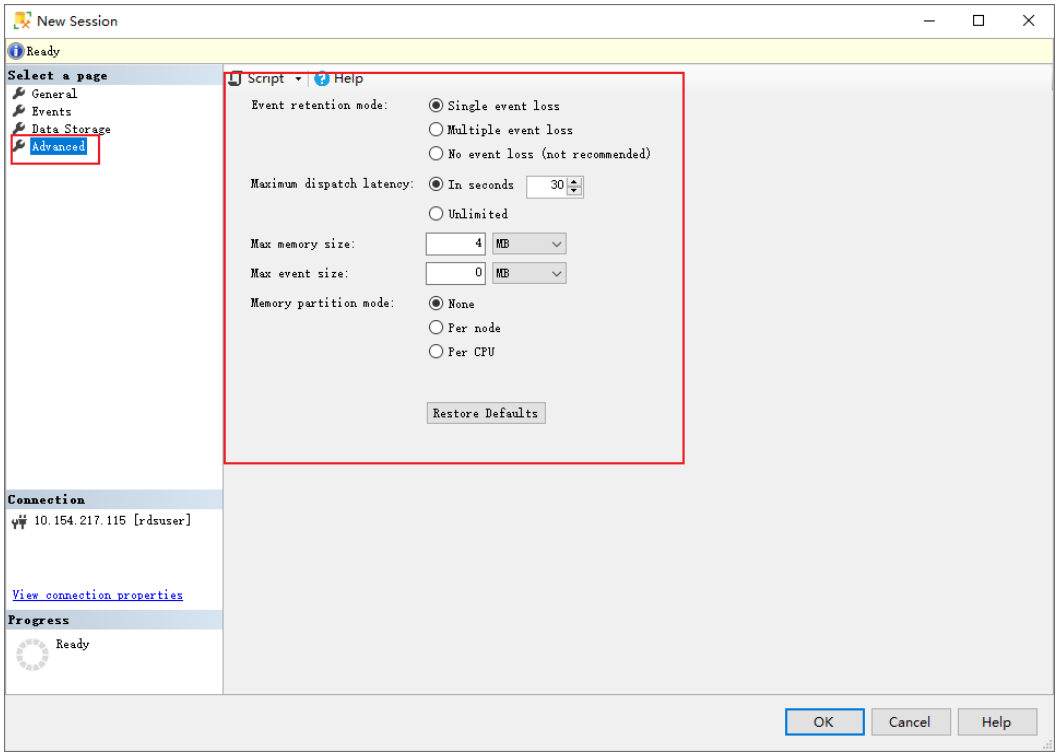
Figure 16-5 Configuring the data storage



Step 7 Click **Advanced** to configure the file generation policy.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Figure 16-6 Configuring the file generation policy



Step 8 Use the script to generate SQL statements. After confirming that the SQL statements are correct, run the SQL statements to create an extended event.

```
-- Example SQL statements generated
CREATE EVENT SESSION [temp_name] ON SERVER
ADD EVENT sqlserver.lock_deadlock(
ACTION(sqlserver.session_id,sqlserver.sql_text,sqlserver.username))
ADD TARGET package0.event_file(SET filename=N'temp_name')
GO
```

----End

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

17 Security Best Practices

With its powerful transaction processing capability, rich functions, and wide support for enterprise-level applications, SQL Server has earned a good reputation in the industry and has become one of the preferred relational databases among many enterprises. RDS for SQL Server is a cloud-based database service that is reliable, scalable, easy to manage, and immediately ready for use. It aims to provide customers with more efficient and secure data management solutions.

To help secure your workload on RDS for SQL Server instances, follow these best practices as needed.

- **Using HA Instances**
- **Enabling Data Backup**
- **Avoiding Binding an EIP to Your Instance**
- **Avoiding Using the Default Port**
- **Periodically Resetting Passwords of Database Accounts**
- **Periodically Checking and Deleting Roles That Are No Longer Used**
- **Configuring Least-Privilege Permissions for Database Accounts**
- **Avoiding Using the sysadmin Permission**
- **Enabling Forcible Encryption**
- **Using TDE**
- **Configuring the Maximum Degree of Parallelism**
- **Using DBSS for Full Audit**

Using HA Instances

Each RDS for SQL Server cluster or primary/standby instance can perform a failover and promote the standby node to primary if the primary node fails. This ensures high availability and service continuity and minimizes downtime and data loss.

Enabling Data Backup

When you create an RDS for SQL Server instance, an automated backup policy is enabled by default with a retention period of seven days. You can change the backup retention period as required. RDS for SQL Server instances support

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

automated backups and **manual backups**. You can periodically back up your instance. If the instance fails or data is damaged, **restore it using backups** to ensure data reliability.

Avoiding Binding an EIP to Your Instance

Do not deploy your RDS for SQL Server instance on the Internet or in a demilitarized zone (DMZ). Instead, deploy it on a Huawei Cloud private network and use routers or firewalls to control access to your instance. To prohibit unauthorized access and DDoS attacks from the Internet, do not bind an EIP to your instance. If your instance already has an EIP, you are advised to unbind it. If you do need an EIP, **configure security group rules** to restrict the source IP addresses that can access your instance.

Avoiding Using the Default Port

The default port of RDS for SQL Server is **1433**, which is vulnerable to malicious attacks. To avoid this risk, **change the port** for your DB instance.

Periodically Resetting Passwords of Database Accounts

Periodically resetting passwords is one important measure to enhance system and application security. This practice not only lowers the chances of password exposure but also helps you meet compliance requirements, mitigate internal risks, and boost your awareness of security. By doing so, you can significantly improve account security and protect sensitive data and systems from potential security threats. For details, see **Resetting a Password for a Database Account**.

Periodically Checking and Deleting Roles That Are No Longer Used

Check whether all roles are mandatory. Every unknown role must be reviewed to ensure that it is used properly. If any role is no longer used, delete it.

Configuring Least-Privilege Permissions for Database Accounts

RDS for SQL Server allows you to grant role-based permissions to a database account for data and command access. You are advised to create **database accounts** and configure least-privilege permissions for the accounts. If any account permission does not meet the role requirements, update the account permission or **delete** the account. RDS for SQL Server has **built-in accounts**. They are used to provide background O&M services for DB instances. Do not use or delete them.

Avoiding Using the sysadmin Permission

The sysadmin permission in RDS for SQL Server is the highest permission. Misusing this permission can lead to security O&M failures. This could result in data corruption, inability to restore data, or difficulties in performing a primary/standby switchover.

To improve the security and stability of RDS for SQL Server, do not use the sysadmin permission. The principle of least privilege, refined permission management, and enhanced audit and monitoring can significantly reduce

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

security risks, protect data and systems, and improve overall security management.

Enabling Forcible Encryption

Forcible encryption ensures that all data transmitted between each client and RDS for SQL Server is encrypted. In this way, data can be effectively prevented from being eavesdropped or tampered with during transmission, improving both data privacy and security. Once forcible encryption is enabled, all clients automatically communicate with RDS for SQL Server in encryption mode. Using this function eliminates the need to configure each client individually, making security management easier.

Using TDE

RDS for SQL Server uses **Transparent Data Encryption (TDE)** to protect data at rest by encrypting data files and backup files. Encryption in TDE is transparent to applications and meets compliance requirements. However, using TDE may affect the performance and storage space.

After TDE is enabled for an RDS for SQL Server instance, data files are encrypted and do not need to be encrypted again using disk encryption. TDE is recommended over disk encryption because disk encryption affects database and OS performance.

Configuring the Maximum Degree of Parallelism

To optimize query performance and resource utilization, you can adjust the maximum degree of parallelism of your RDS for SQL Server instance by setting **max degree of parallelism**. Setting this parameter to a large value can cause lock blocking, and setting it to a small value can cause insufficient resource utilization. You are advised to set it based on the resource usage and instance specifications. It is recommended to set the initial value to half of the vCPUs of your DB instance or adjust the value based on the actual load. For details about how to modify a parameter, see **Modifying RDS for SQL Server Instance Parameters**.

Using DBSS for Full Audit

Audit logs can capture detailed records that auditors usually need to meet compliance regulations. For example, DDL audit is enabled for RDS for SQL Server instances by default to track changes in server settings, database, and table structures. In addition, you are advised to use Database Security Service (DBSS) to obtain full audit logs. The audit logs contain DML audit content and can be stored for 180 days or longer.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.