Importing a BACPAC file for a SQL database using SSMS
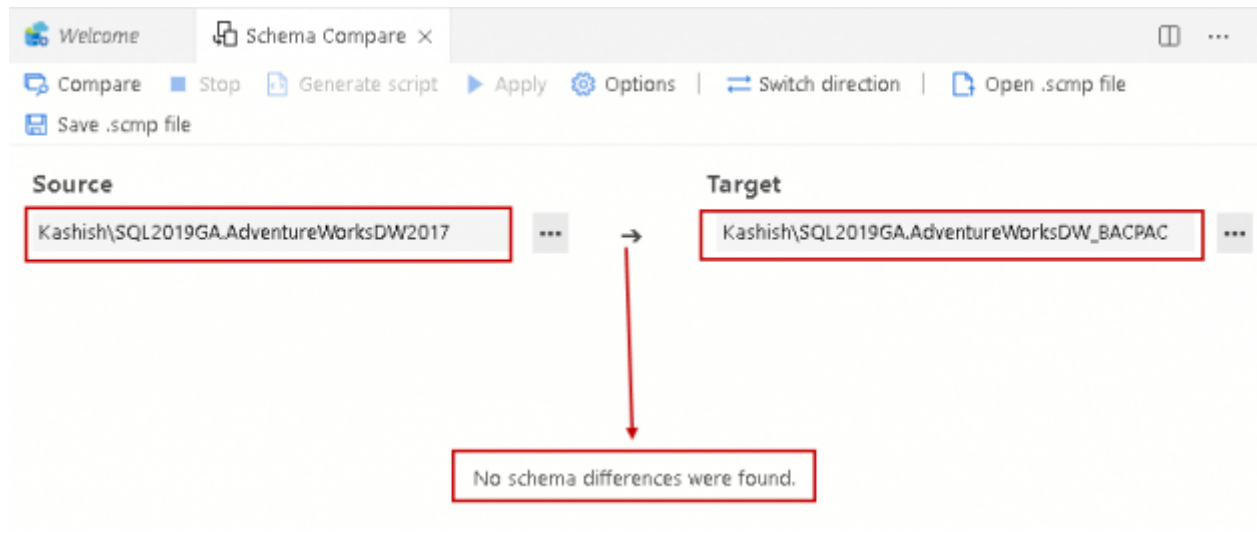
This article gives you an overview of BACPAC package and its usage in SQL Database data refresh (data import and export) using SQL Server management studio.

Introduction

Sometimes we get the requirement of refreshing few tales from other copies of the SQL Database. You can use various methods for this task as following:

- Generate data script
- Data import and export wizard

SQL Server does not support restoring individual tables from database backups. You can restore database backup with another database and import data from that database. For large databases, we cannot use this approach due to space and time constraints. You require space for storing the backup file as well as data and log files.

Another problem is the firewall issues. Many times, organizations do not allow firewalls between production and test environment. In this case, it becomes difficult to copy production data to lower the environment using a jump server using import and export wizards and backup methods as well. In this article, we will look at an alternative method of using a BACPAC package for refreshing tables.

Overview of a DACPAC and BACPAC package

DACPAC is an acronym of **D**ata-Tier **A**ppli**c**ation **Pac**kage. It is a logical database entity defining all database objects such as tables, views, users and logins. It enables developers and database administrators to create a single package file consisting of database objects.

We call this package a DACPAC package.

Usually, developers use SQL Server Data Tools (SSDT) for writing t-SQL scripts and making changes as per the requirement. They can generate a DACPAC package and send it to DBAs for deploying in multiple environments. Similarly, DBA can give the DACPAC copy of a production database to developers for their development purpose.

We can also use DACPAC packages for schema comparison as well between different copies of it. You can use the Azure Data Studio SQL Server Schema Compare extension for this.

Similarly, BACPAC (**Bac**kup **pac**kage) consists of schema and data for SQL Server tables. We can export database tables in a BACPAC package and give it to developers for refreshing data per their convenience.

We get options for DACPAC and BACPAC package in SSMS.

Note: I would recommend always using the latest version of SQL Server Management Studio. The latest version is SSMS 18.4.

Connect to a SQL instance and expand databases in the object explorer. Right-click on a database in SSMS and you get data-tier application options as shown below.
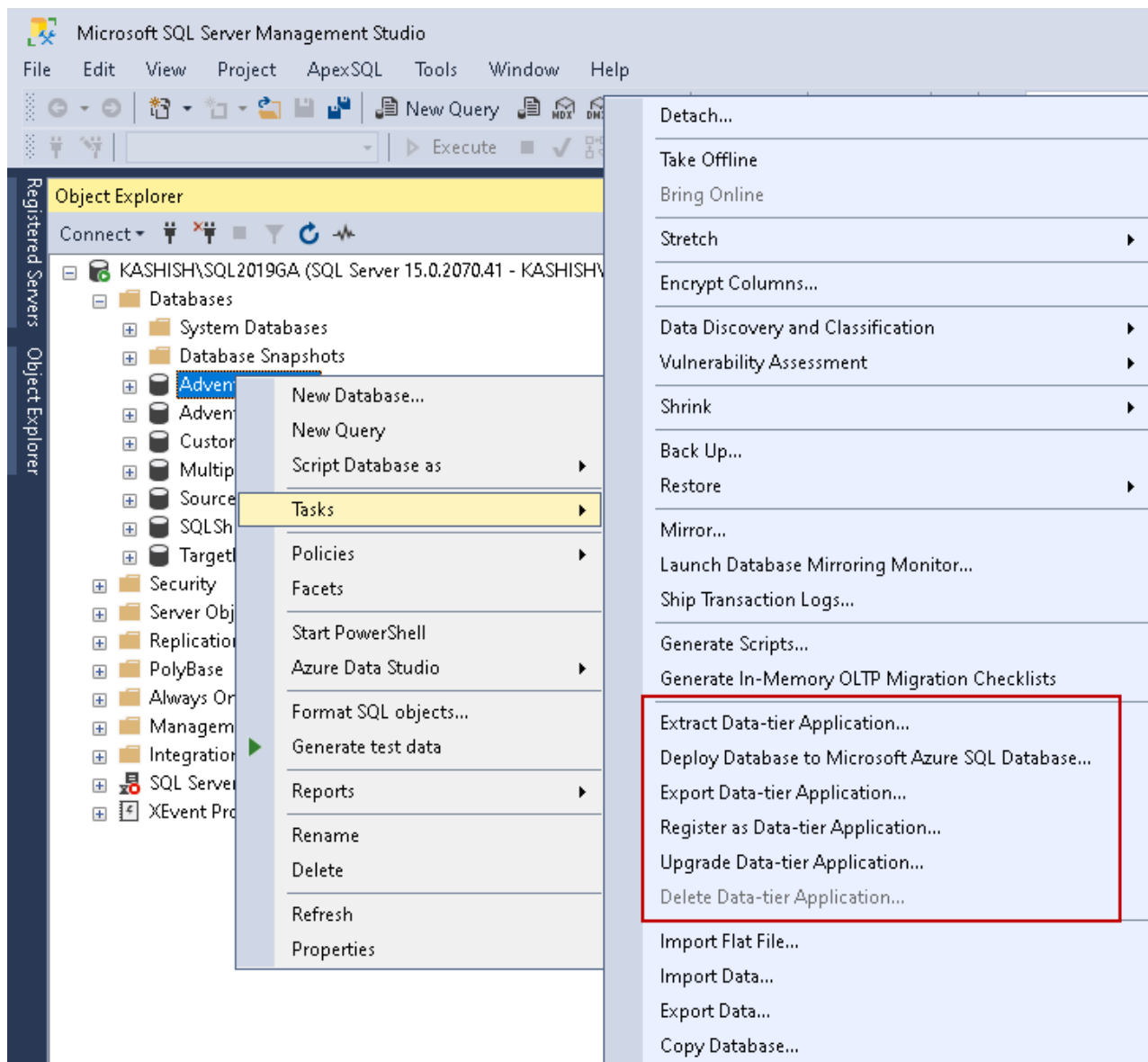
- **Export Data-tier application:** We can extract schema and data as well in a BACPAC file using this option
- **Extract Data-tier application:** We use this wizard for extracting a database in a DACPAC package. This DACPAC package does not contain data
- **Upgrade Data-tier application:** We can upgrade a previously deployed database using this upgrade data tier option

## Export Data-tier application wizard for BACPAC package

Let's export data and schema for a database and create another copy of a database in the same instance for this demonstration.

We use the following steps for doing this task.

Right-click on the source SQL Database and click on **Export Data-tier application.** It launches an introduction page. This introduction page defies the summary and high-level steps for this wizard. The high-level steps are as follows:

- Specify export settings

- Review the export summary
- Check the result of the operation
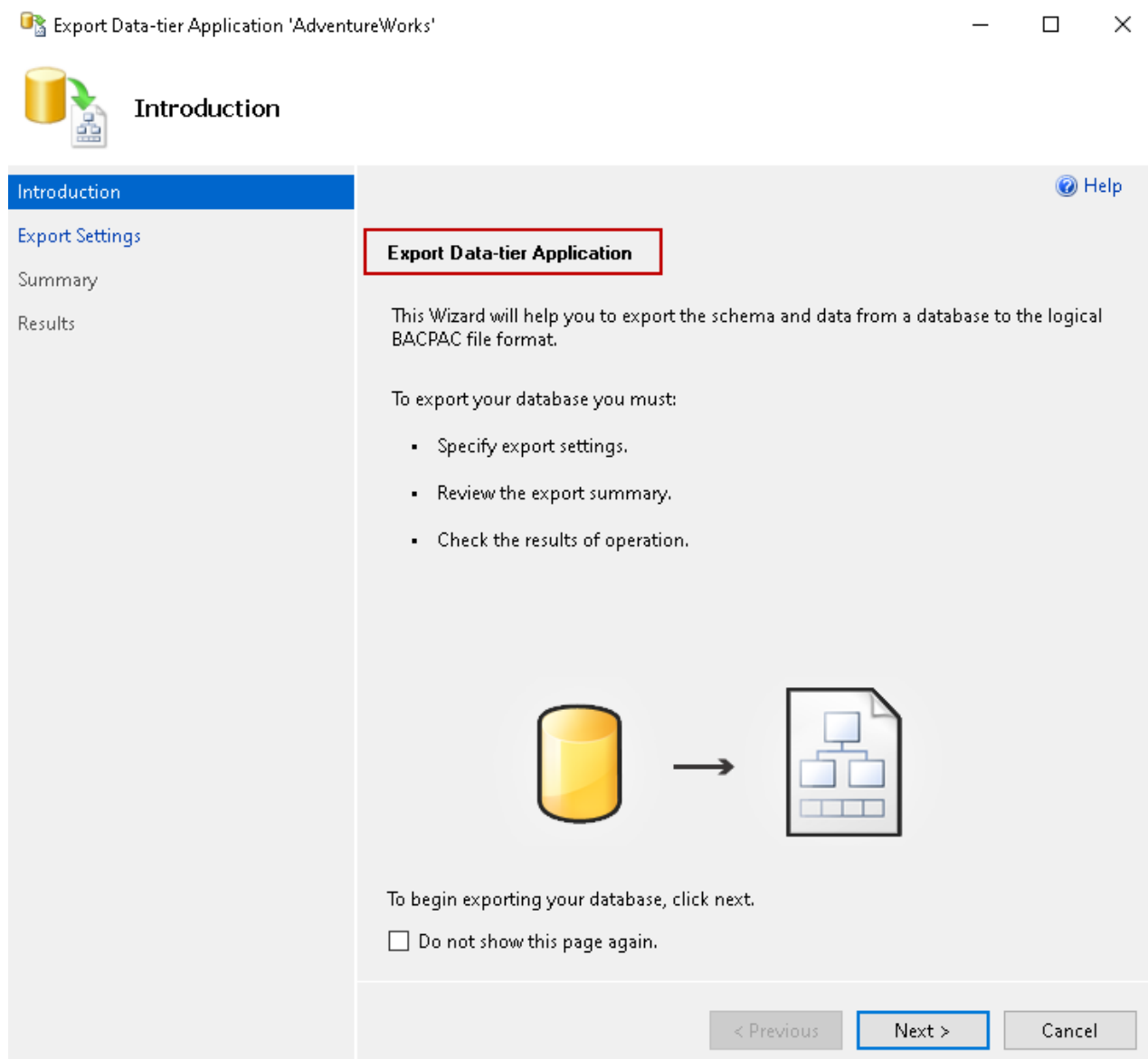
If we do not require this introduction page from now onwards, we can check **Do not show this page again** option.

Click next, and it takes you to the **Export setting** page.

The export setting page allows us to configure either a local disk directory or Microsoft Azure storage container. For Microsoft Azure, you should have a valid storage account and container for storing the BACPAC file.

Let's use a local disk directory, as shown below.

By default, it extracts all schema, tables (with data)available in the database. We can also filter selected objects in the advanced page select all database schema, tables in the advanced tab of this page.

Click on the advanced tab and review the objects. It groups objects per schema such as dbo.

Export Data-tier Application 'AdventureWorksDW2017'

Export Settings

Introduction
Export Settings
Summary
Results

⑦ Help

Export Settings

This operation will create a BACPAC file that contains the logical contents of your database. To continue, specify the location where you want the BACPAC file to be created, and then click Next. To specify a subset of tables to export, use the Advanced option.

Settings | Advanced

Select tables for data export:

☐ ☑ Select All
　☐ ☑ dbo
　　☑ AdventureWorksDWBuildVersion
　　☑ DatabaseLog
　　☑ DimAccount
　　☑ DimCurrency
　　☑ DimCustomer
　　☑ DimDate
　　☑ DimDepartmentGroup
　　☑ DimEmployee
　　☑ DimGeography
　　☑ DimOrganization
　　☑ DimProduct
　　☑ DimProductCategory
　　☑ DimProductSubcategory
　　☑ DimPromotion
　　☑ DimReseller
　　☑ DimSalesReason
　　☑ DimSalesTerritory

< Previous | Next > | Cancel

We are not using any object filtering in BACPAC export. It will create an export for all tables in the database. Click on Next and review the configurations.
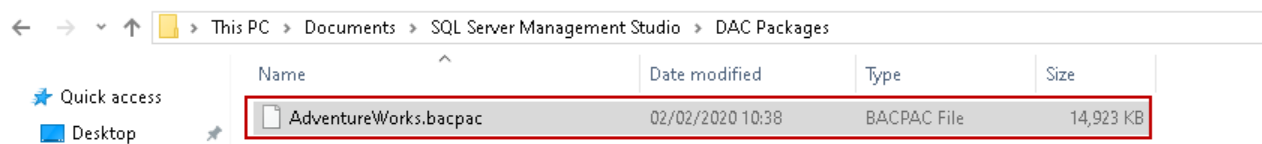
Click Finish, and it starts the export of the database. You can see the success message for each step.
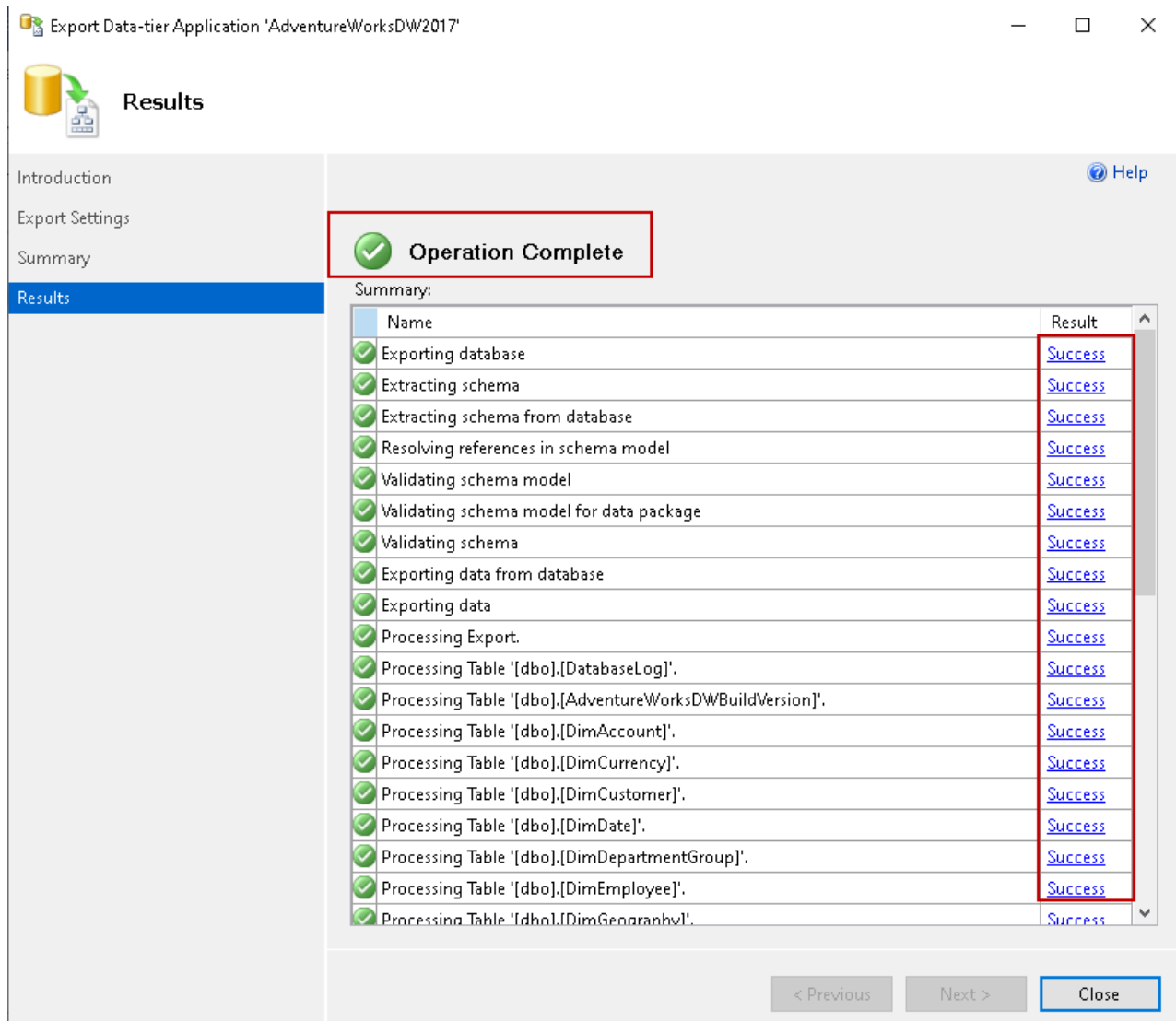
We can browse to the directory and verify a BACPAC extension file exists.

Import a BACPAC file using SSMS Import Data-tier Application wizard

We can copy this BACPAC file to destination SQL instance having either on-premise or Azure SQL Database. Right-click on the connected destination instance and launch **Import Data-tier applications** wizard.

The high-level steps for importing a BACPAC package are:

- Specify the BACPAC import file
- Import Configuration
- Import and verify data

Click next, and it takes you to the import settings page. Here, we specify the path of a BACPAC file either from the local disk or Microsoft Azure Storage container. We used local disk storage during the import process so, select the local disk option and browse to the directory containing required BACPAC package.

In the next page, we specify a setting for the new database. We cannot use an existing SQL Database for importing a BACPAC package.

It also requires input for the data and log file of this new database. By default, it shows you default data and log file location of connected SQL instance. We can change it per requirement.

As highlighted earlier, we cannot use an existing database for BACPAC file import. We get the following error message if the database already exists.

Modify the new database name. In my demonstration, I specified AdventureWorksDW_BACPAC name for SQL Database. Click next and view summary.

The next page is a summary page showcasing your configurations. I recommend you always review it before proceeding. You can easily navigate back and make the required changes.
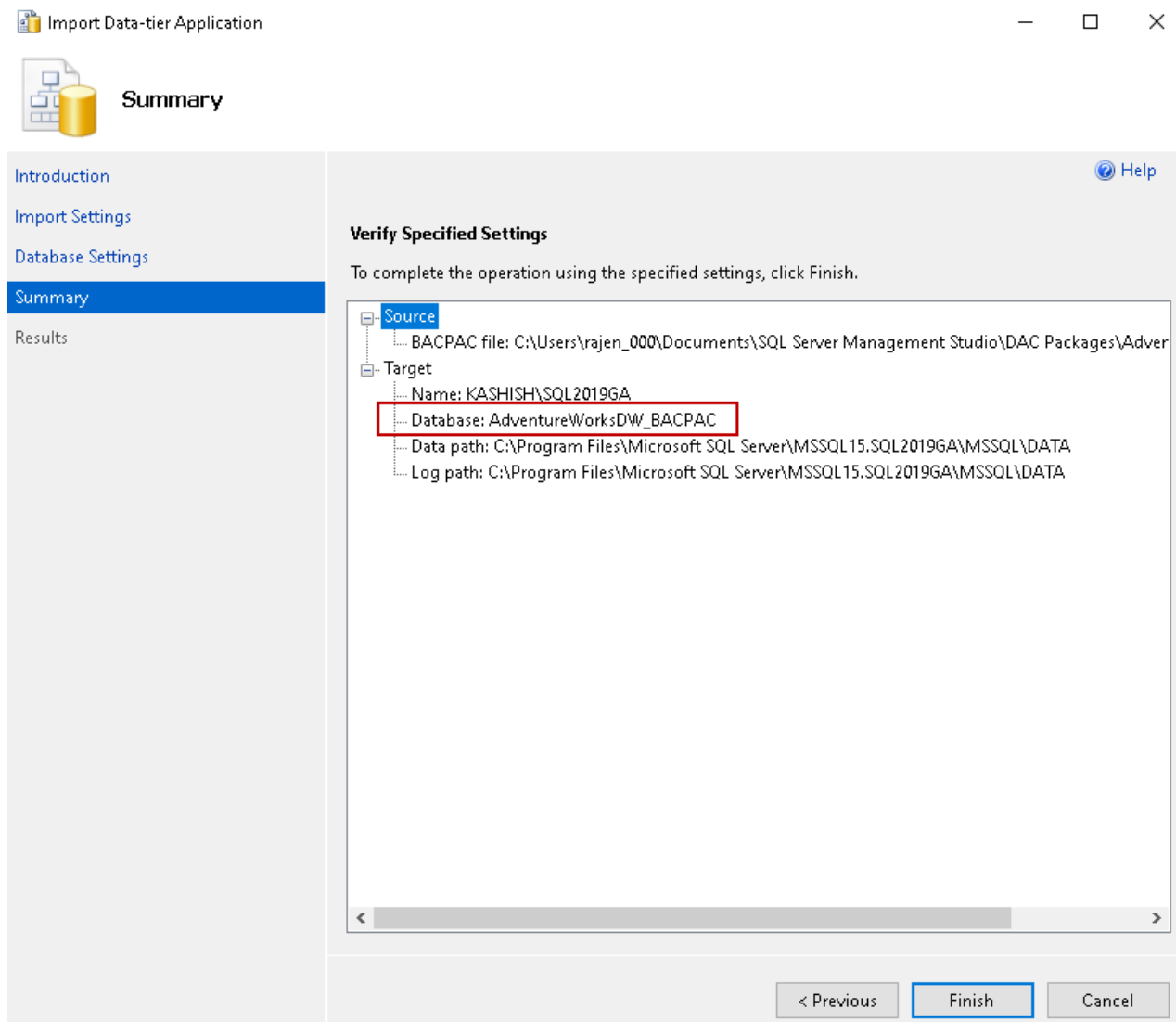
Click Finish, and you can see the following steps.

- Creates a new database
- Imports schema and data into this new database
- Disables indexes
- Inserts data into tables
- Enable indexes

Import Data-tier Application — □ ×

Progress

Introduction

Import Settings

Database Settings

Summary

Results

? Help

Importing database

	Name	Status	
✓	Creating database on target	Success	
✓	Creating deployment plan	Success	
✓	Initializing deployment	Success	
✓	Verifying deployment plan	Success	
✓	Analyzing deployment plan	Success	
⟳	Importing package schema and data into database	In Progress	
⟳	Updating database	In Progress	
⟳	Importing data	In Progress	
⟳	Processing Import.	In Progress	
✓	Disabling indexes.	Success	
✓	Disabling index 'PK_DatabaseLog_DatabaseLogID'.	Success	
✓	Disabling index 'AK_DimCurrency_CurrencyAlternateKey'.	Success	
✓	Disabling index 'IX_DimCustomer_CustomerAlternateKey'.	Success	
✓	Disabling index 'AK_DimDate_FullDateAlternateKey'.	Success	
✓	Disabling index 'AK_DimProduct_ProductAlternateKey_StartDate'.	Success	
✓	Disabling index 'AK_DimProductCategory_ProductCategoryAlternateKey'.	Success	

⊙ Fewer details

< Previous   Next >   Cancel

15/19
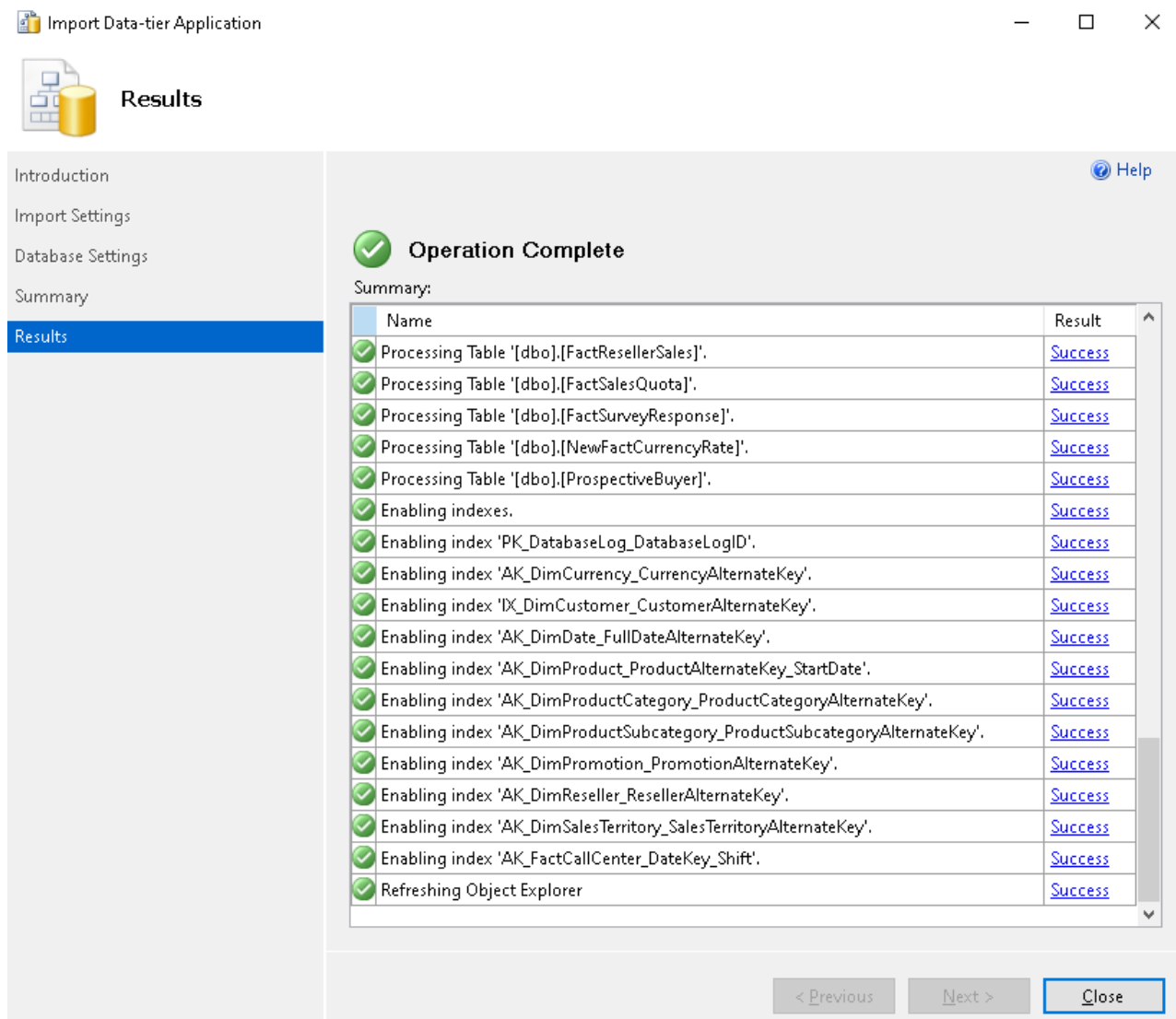
Refresh the object explorer, and you see the newly restored database.

Let's compare the record counts for [dbo].[DimDate] tables in both **AdventureWorksDW2017** and **AdventureWorksDW_BACPAC** databases.

In the following screenshot, verify that it contains the same record count in both the SQL Databases.

```
□SELECT COUNT(*) as [AdventureWorksDW2017]
    FROM [AdventureWorksDW2017].[dbo].[DimDate]

□SELECT COUNT(*)as [AdventureWorksDW_BACPAC]
    FROM [AdventureWorksDW_BACPAC].[dbo].[DimDate]
```

110 %

▦ Results ▤▤ Messages

	AdventureWorksDW2017
1	3652

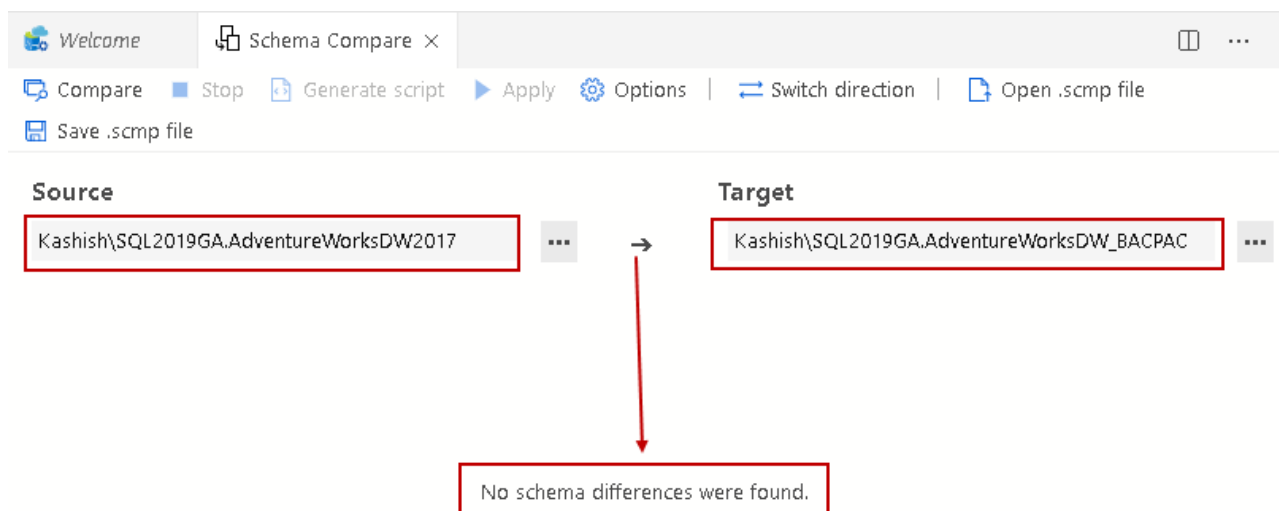|	AdventureWorksDW_BACPAC
1	3652
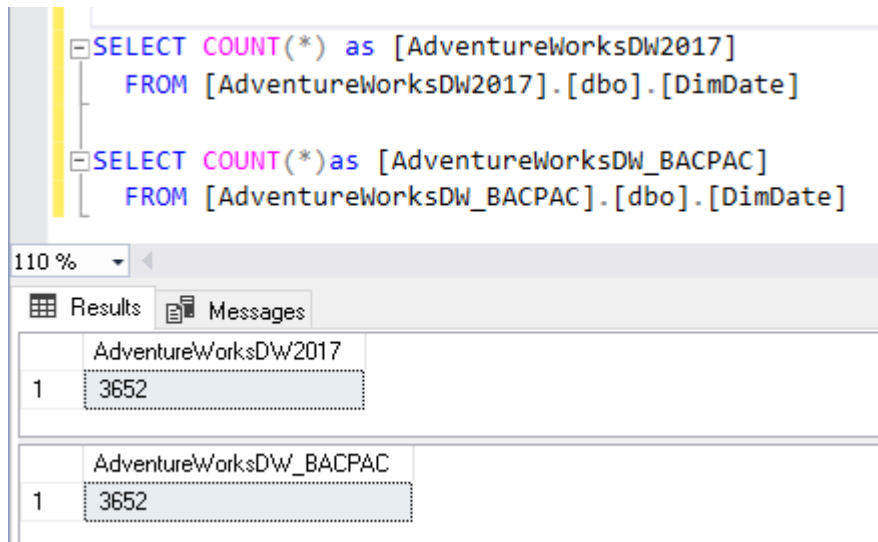
Validating schema of the database imported from a BACPAC package

Let's take a step further and compare the schema in both the databases using schema compare extension of Azure Data Studio. I am not elaborating the steps here for this. You can refer to this article Schema Compare extension for a detailed explanation.

As you can see, there is no schema difference between the source (AdventureWorksDW2017), and the target database restored from BACPAC (AdventureWorksDW2017_ BACPAC).

🞰 Welcome   🔄 Schema Compare ✕   ▢ ⋯

🔄 Compare   ■ Stop   ▣ Generate script   ▶ Apply   ⚙ Options | ⇄ Switch direction | 🗋 Open .scmp file
🖫 Save .scmp file

Source   **Target**

| Kashish\SQL2019GA.AdventureWorksDW2017 | ⋯ | → | Kashish\SQL2019GA.AdventureWorksDW_BACPAC | ⋯ |

No schema differences were found.

You have data in a new database. Now, you can use import and export wizard as explained in this article How to import/export data to SQL Server using the SQL Server Import and Export Wizard.

For example, let's say we require a refresh of[dbo].[ProspectiveBuyer] the table in the AdventureWorks2017DW database. We can use the following steps for it.

- Create a BACPAC file from the source AdventureWorks2017DW database for the [dbo].[ProspectiveBuyer] table as steps shown above
- Copy this BACPAC file from the source server to the destination server
- Import this BACPAC file and create a new SQL Database on the destination instance

- Use Import and export wizard in copying data from a new database to source database

Conclusion

In this article, we explored the usage of a BACPAC package for export and import of SQL database tables data. It is an alternate method for data refresh; however, you should review the possible approaches and choose what fits best for your requirements.

See more

To boost SQL coding productivity, check out these SSMS add-ins including T-SQL formatting, refactoring, auto-complete, text and data search, snippets and auto-replacements, SQL code and object comparison, multi-db script comparison, object decryption and more

Watch Video At: https://youtu.be/UC-felOMDdk
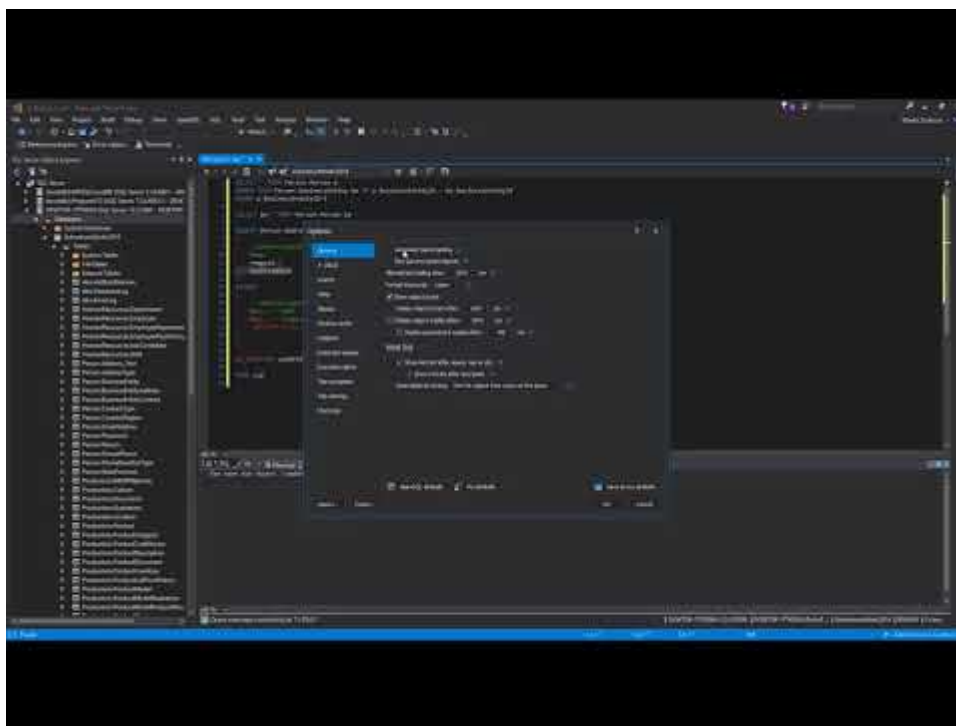
Rajendra Gupta

As an MCSA certified and Microsoft Certified Trainer in Gurgaon, India, with 13 years of experience, Rajendra works for a variety of large companies focusing on performance optimization, monitoring, high availability, and disaster recovery strategies and implementation. He is the author of hundreds of authoritative articles on SQL Server, Azure, MySQL, Linux, Power BI, Performance

tuning, AWS/Amazon RDS, Git, and related technologies that have been viewed by over 10m readers to date.

He is the creator of one of the biggest free online collections of articles on a single topic, with his 50-part series on SQL Server Always On Availability Groups. Based on his contribution to the SQL Server community, he has been recognized with various awards including the prestigious "Best author of the year" continuously in 2020 and 2021 at SQLShack.

Raj is always interested in new challenges so if you need consulting help on any subject covered in his writings, he can be reached at rajendra.gupta16@gmail.com

View all posts by Rajendra Gupta