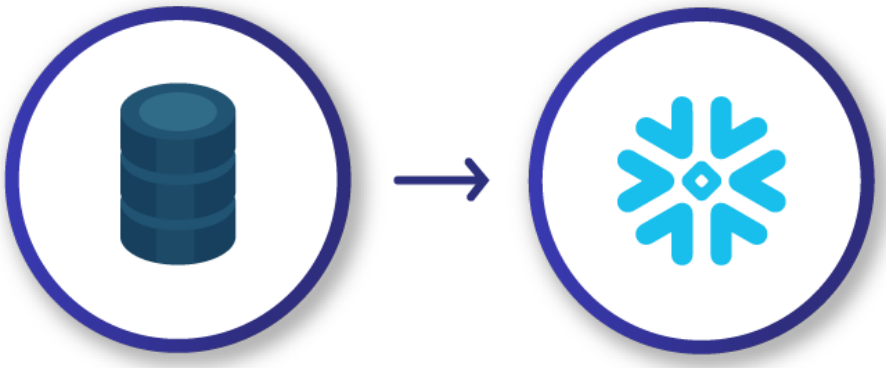
Database Migration to Snowflake: A Complete How-to Guide



Summary

Introduction

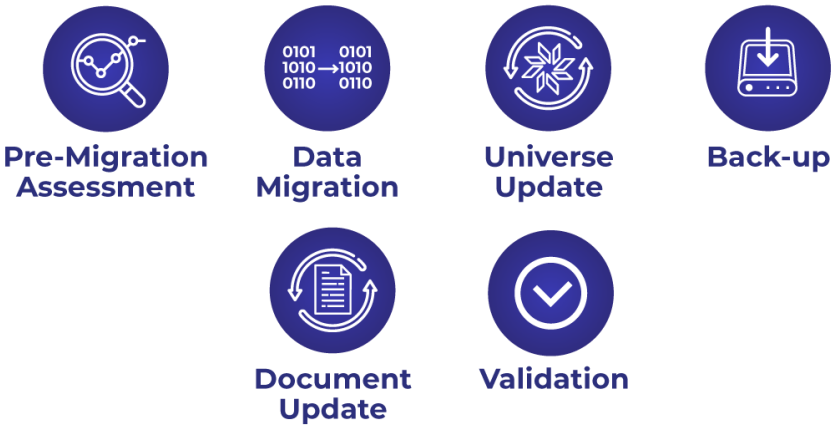
This guide will show you how to migrate[1] a relational database from Microsoft SQL Server 2019 for example, to Snowflake.

Then how to manually manage your SAP BusinessObjects content:

- Universe(s) and their connections to Snowflake

And if necessary, how to:

- Repoint your Web Intelligence document(s) to your Snowflake Universe(s)
- Repoint your Crystal Reports document(s) to your Snowflake ODBC



1. Identify which Universes, Reports, and Users will be impacted by repointing the database connectivity to Snowflake.

   This step can also be used to identify and document which tables and columns in your databases are used (and not used) by SAP BusinessObjects. This can help identify the data that needs to be migrated, in which order or not at all.

2. This is the migration step by the customer.

3. There are two scenarios:

[1] More definitions here: https://mssql.tosnowflake.com/

a. Simply repointing the Universe Connection to Snowflake works, the Universe passes all integrity checks, you can start the validation of the Webi reports (step 5).

   If you have Crystal Reports pointing directly at the database and not via a Universe, they will need to be updated with the new database connections (step 4).

b. Due to changes in the Snowflake Schema, Column Type or just vendor-specific SQL inside the Universe means that we need to make changes to the Universe.

   This step is also applicable if you are converting from a UNV to a UNX.

   Here we will work with a copy of the existing Universe and apply the necessary changes so they are fully operational. Depending on your strategy, you may need to later repoint all the documents that use the old Universe to the new one (step 4).

4. Back-up: before making any changes to your content, make sure that you have a reliable and performant back-up.

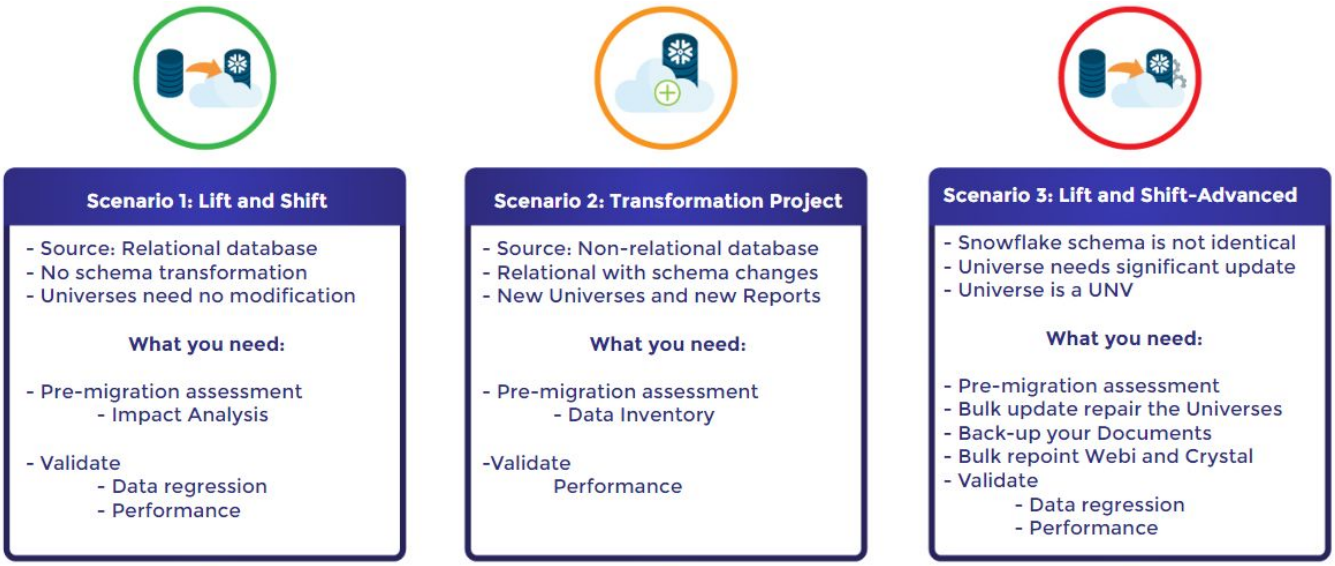
5. Repointing Webi and Crystal Documents (tutorial video here) to the new / updated universes.

6. Testing and Validation: ensure user satisfaction and meet regulatory needs by testing the data, its security, the layout of the documents, the network connectivity and performance of your documents.

Disclaimer

This blog demonstrates the concepts of a database migration and how to manage SAP BusinessObjects content affected by this project. Every case is different and the steps mentioned here may not be the same for you. Here we took the example of a migration from a Microsoft SQL Server to Snowflake, but this guide can apply to a multitude of different databases.

For completeness of this document we are simulating the workflow where migrating the database will require changes to the schema: database name, table names therefore involving changes in the Universe(s).



This guide will cover the Lift & Shift Advanced scenario showing all the steps involved when simply changing the Universe connection is not sufficient.

Readme.txt

Before any migration project, it is important to carry out a Pre-Migration Impact Analysis first in order to decide what needs to be migrated over. You should also analyze what will be impacted by this project to help avoid any risks during the whole process.

Examples:

- Universes:
  - List of Connections pointing to the database(s) to Migrate
  - List of Universes pointing to these Universe Connections
  - List of Universe Restrictions (aka overloads)
  - Document Universes Usage / Non-Usage
  - Document Universe Objects (dimensions, details, measure) Usage / Non-Usage

- Content:
  - List of Web Intelligence, Crystal Reports and other documents pointing to these Universes
  - List of Web Intelligence, Crystal Reports and other documents directly pointing to these database(s). E.g. Crystal Reports 2016
  - Document Web Intelligence formulas that might be affected
  - Document impacted content Usage / Non-Usage
  - Document Instances impacted by this migration

- Users:
  - Document users impacted (based on actions and ownership) - For better communication

- Data:
  - Document Database Tables to be migrated based on Impact Analysis and Usage / Non-Usage
  - Document Columns in Tables to be migrated based on Impact Analysis and Usage / Non-Usage

Pre-Requisites

This blog assumes you have SAP BusinessObjects 4.2 SP08 or higher as it is the earliest release officially supporting Snowflake.

It also assumes you have Universe(s), Web Intelligence and Crystal Reports documents pointing to a Microsoft SQL Server Database. This scenario can similarly be applied to any relational database.

Also, you need to have Snowflake ODBC and/or JDBC connectivity configured for SAP BusinessObjects. See this blog for more details:
https://blogs.sap.com/2020/03/12/snowflake-for-sap-businessobjects-4.2-sp08/

Microsoft SQL Server

Version: Microsoft SQL Server 2019

Database to Migrate: AdventureWorks2017 (https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15)

Database Size: 336 MB (71 Tables for over 760k rows)

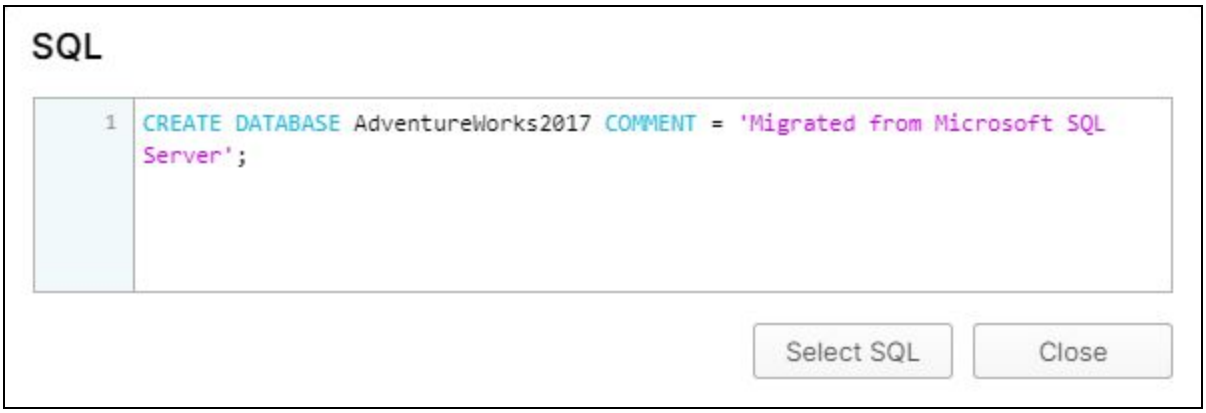
[redacted]\SQLEXPRESS (SQL Server 15.0.2000 - [redacted]\Administrator)\Databases\AdventureWorks2017\Tables

Name	Schema	Create Date
Address	Person	27/10/2017 14:33
AddressType	Person	27/10/2017 14:33
AWBuildVersion	dbo	27/10/2017 14:33
BillOfMaterials	Production	27/10/2017 14:33
BusinessEntity	Person	27/10/2017 14:33
BusinessEntityAddress	Person	27/10/2017 14:33
BusinessEntityContact	Person	27/10/2017 14:33
ContactType	Person	27/10/2017 14:33
CountryRegion	Person	27/10/2017 14:33
CountryRegionCurrency	Sales	27/10/2017 14:33
CreditCard	Sales	27/10/2017 14:33
Culture	Production	27/10/2017 14:33
Currency	Sales	27/10/2017 14:33
CurrencyRate	Sales	27/10/2017 14:33
Customer	Sales	27/10/2017 14:33
DatabaseLog	dbo	27/10/2017 14:33
Department	HumanResources	27/10/2017 14:33
Document	Production	27/10/2017 14:33
EmailAddress	Person	27/10/2017 14:33
Employee	HumanResources	27/10/2017 14:33
EmployeeDepartmentHistory	HumanResources	27/10/2017 14:33

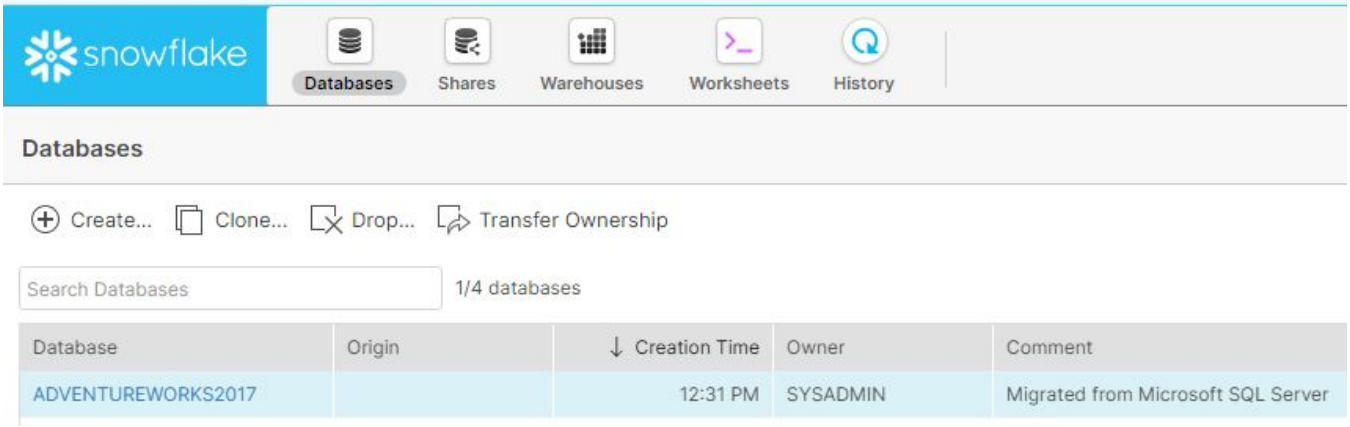
Snowflake

Create an empty database in Snowflake

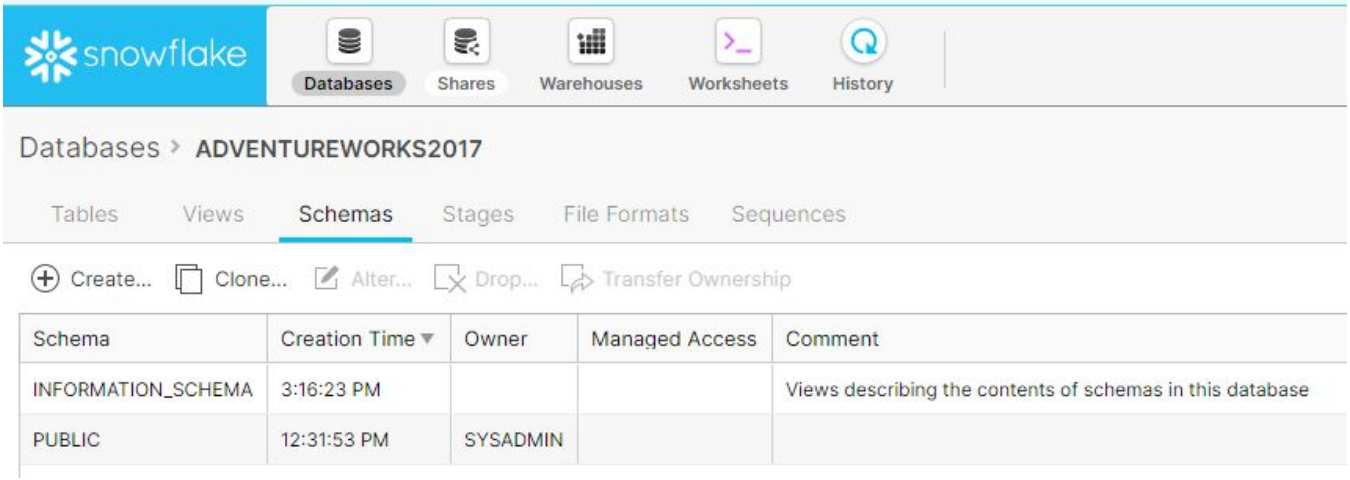
Database Name: AdventureWorks2017



Note: Unless you create tables and columns using double-quotes (therefore case sensitive) these identifiers will be displayed in uppercase but are case-insensitive.
Suggested Reading: Identifiers in Snowflake



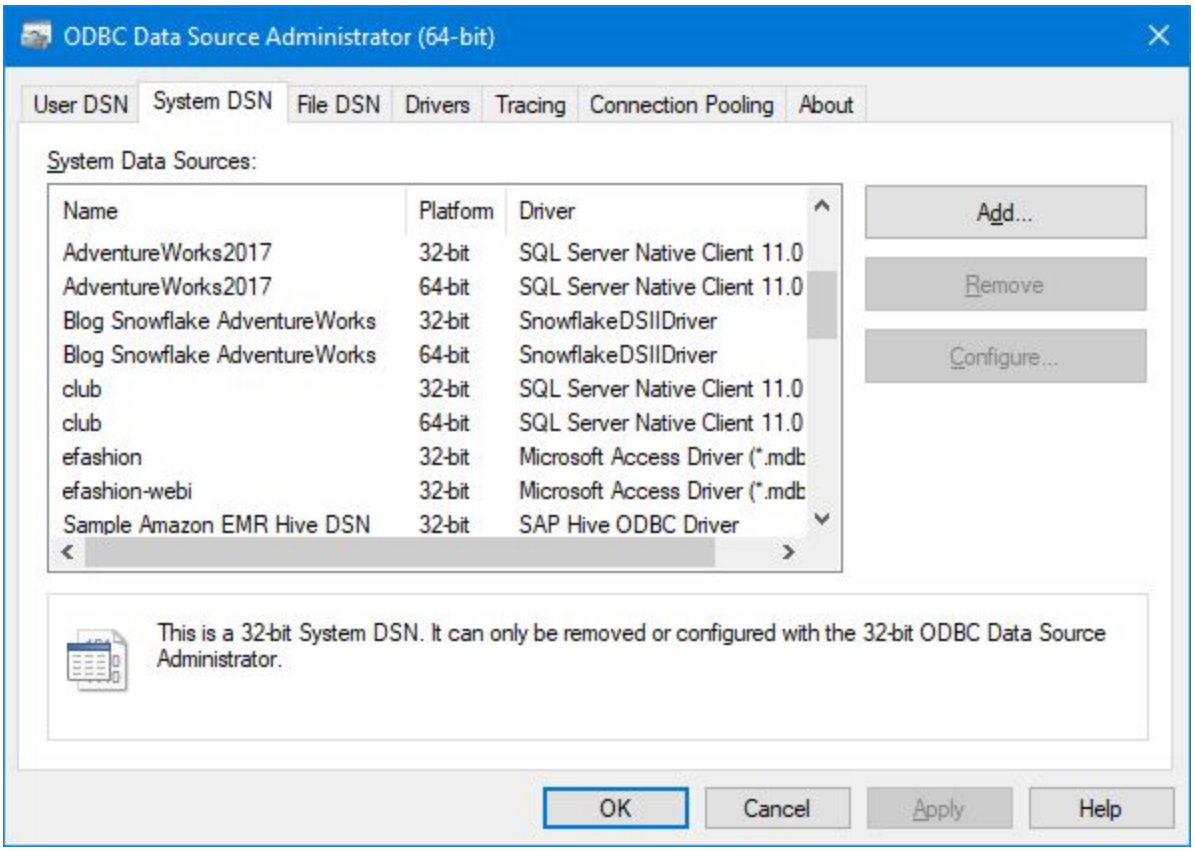
Schema (PUBLIC) available:



Schema	Creation Time	Owner	Managed Access	Comment
INFORMATION_SCHEMA	3:16:23 PM			Views describing the contents of schemas in this database
PUBLIC	12:31:53 PM	SYSADMIN		

Create identical 32-bit and 64-bit ODBC connections to Snowflake

Note: See this blog for more details:
https://blogs.sap.com/2020/03/12/snowflake-for-sap-businessobjects-4.2-sp08/



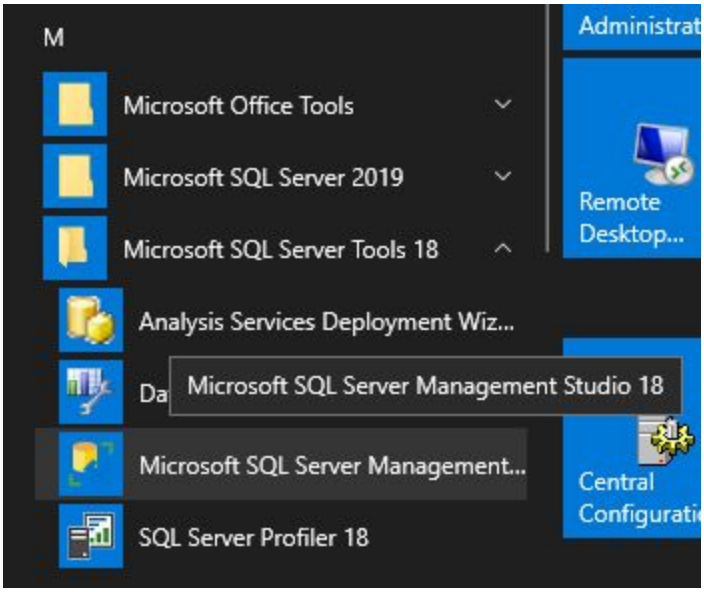
Note: You can use JDBC connections if you prefer. Please refer to the blog above.

Migrating the Database

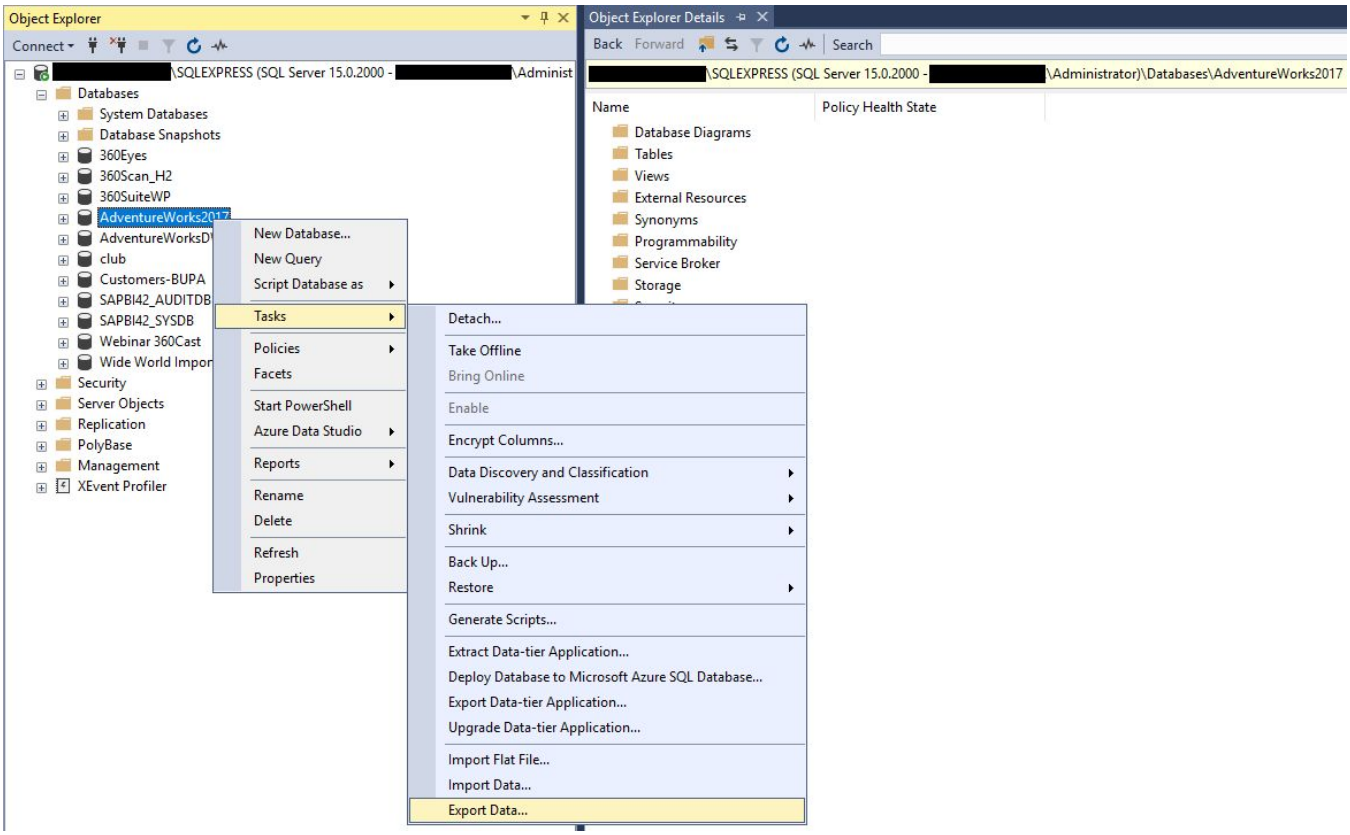
Move the database from Microsoft SQL Server to Snowflake

There are many strategies to run this task. In this blog, we'll use the *SQL Server Import and Export Wizard* via SQL Server Integration Services (SSIS) to generate Comma Separated Values (CSV) file and manually import them into Snowflake.

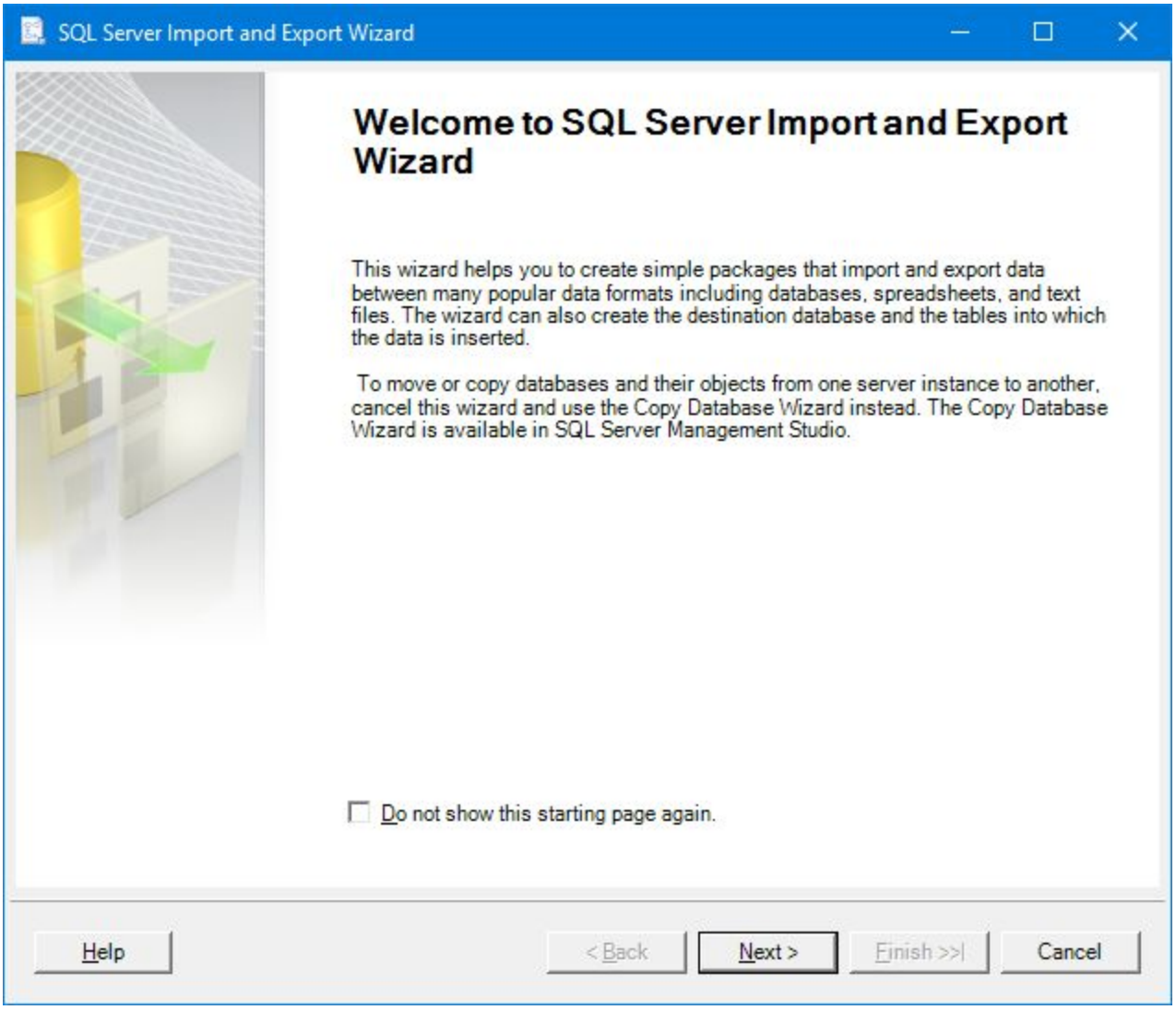
Open Microsoft SQL Server Management Studio



Select the database to migrate (e.g.: AdventureWorks2017)

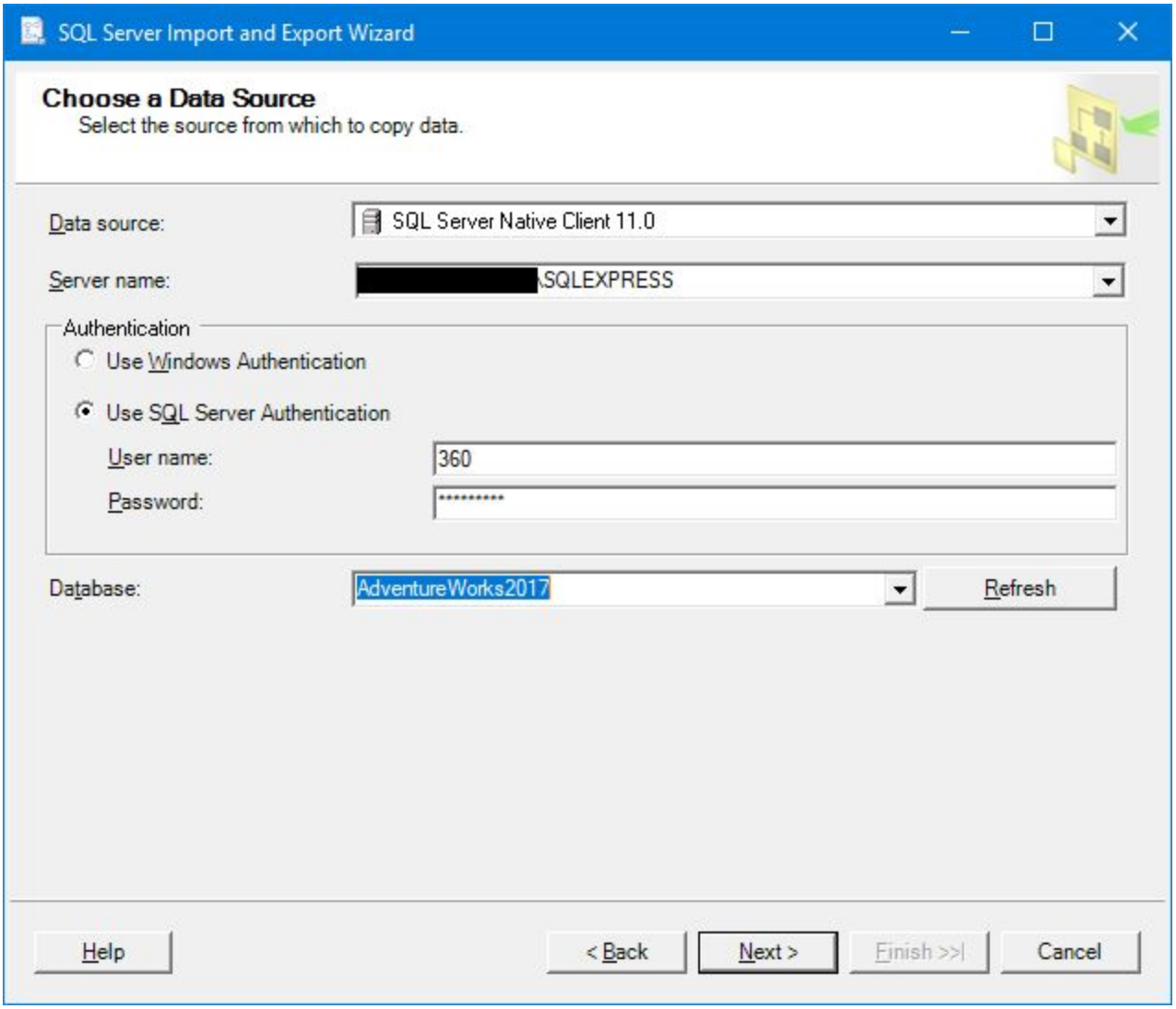


Right-Click > Tasks > Export Data...


SQL Server Import and Export Wizard
Welcome to SQL Server Import and Export Wizard
This wizard helps you to create simple packages that import and export data between many popular data formats including databases, spreadsheets, and text files. The wizard can also create the destination database and the tables into which the data is inserted.
To move or copy databases and their objects from one server instance to another, cancel this wizard and use the Copy Database Wizard instead. The Copy Database Wizard is available in SQL Server Management Studio.
Do not show this starting page again.
Help
< Back
Next >
Finish >>|
Cancel

Click: Next

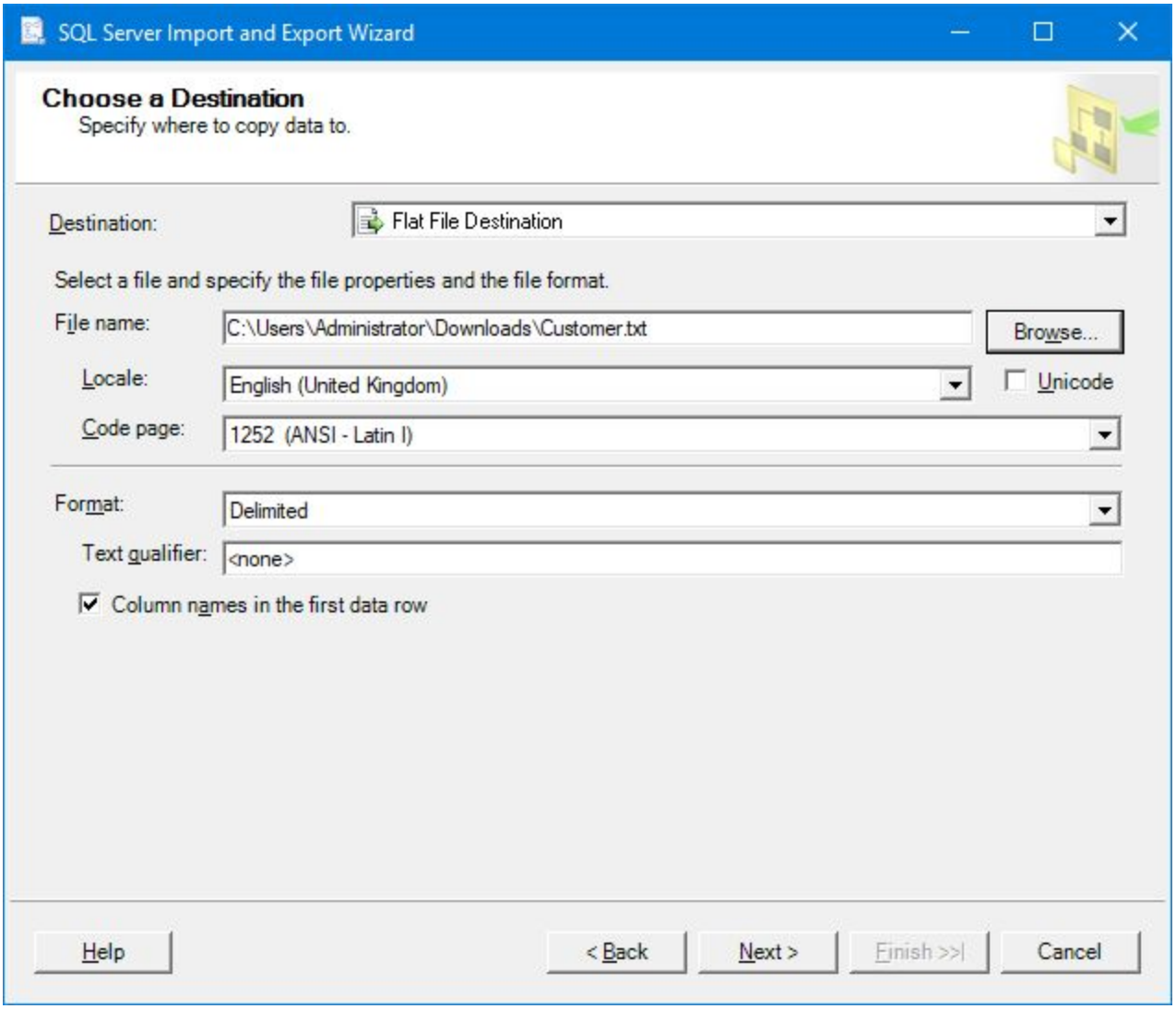
Choose a Data Source



Data source: SQL Server Native Client 11.0
Server name: [ENTER YOUR SERVER NAME / INSTANCE]
Authentication: [ENTER YOUR CREDENTIALS]
Database: [ENTER YOUR DATABASE] (E.g.: AdventureWorks2017)

Click: Next

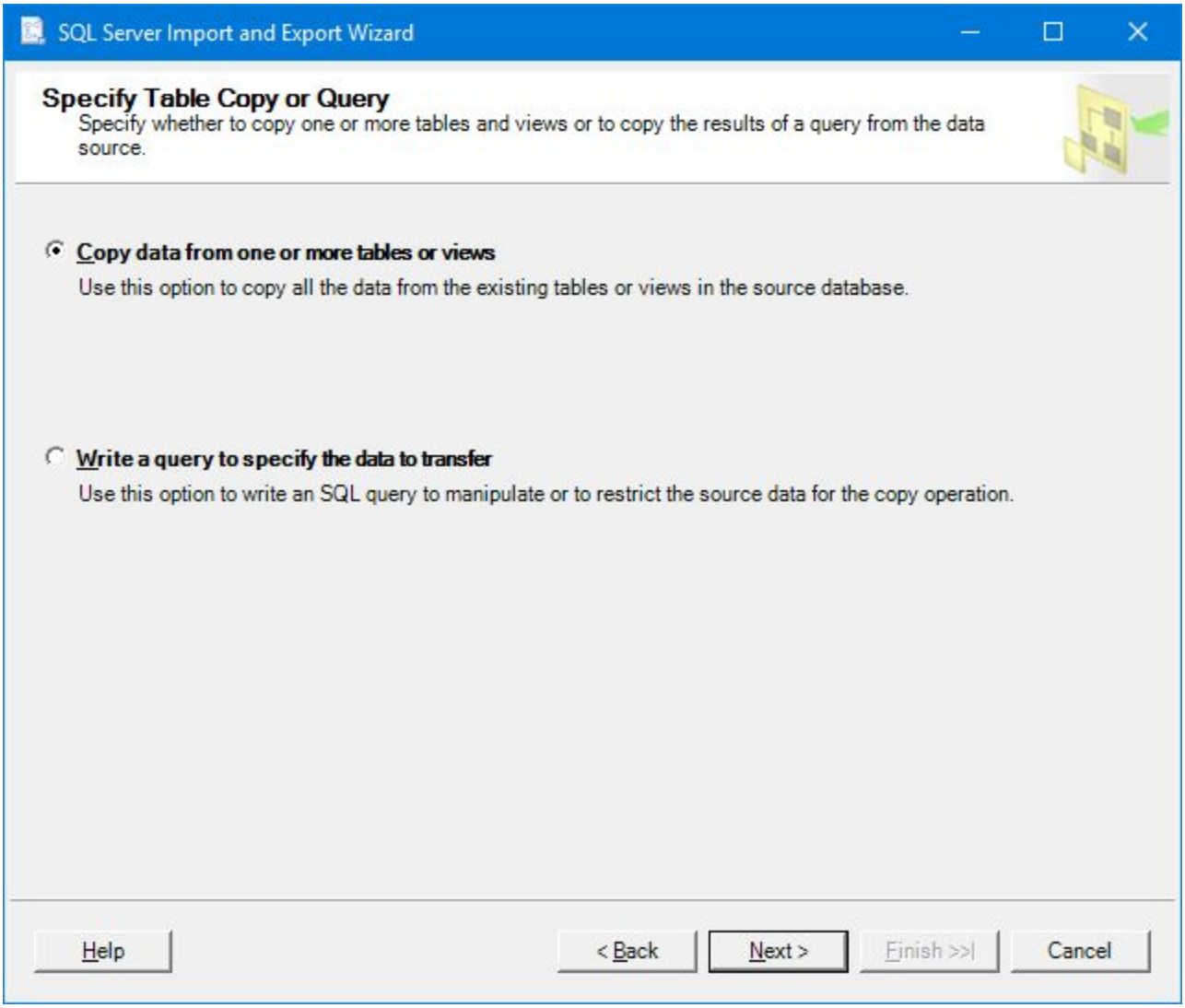
Choose a Destination



Destination: Flat File Destination
File name: [BROWSE TO PATH AND ENTER A FILE NAME]

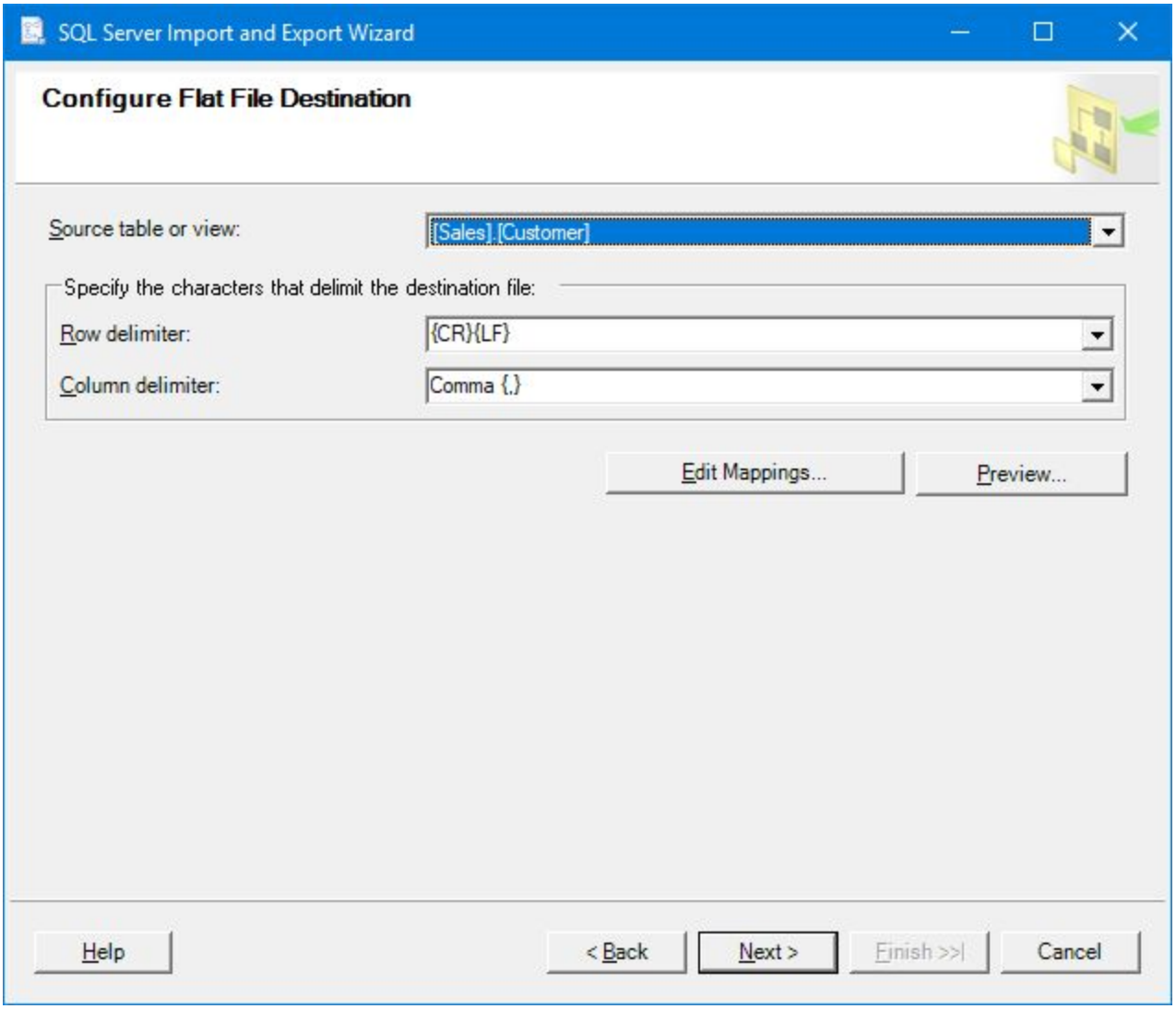
Click: Next

Specify Table Copy or Query



Click: Next

Configure Flat File Destination

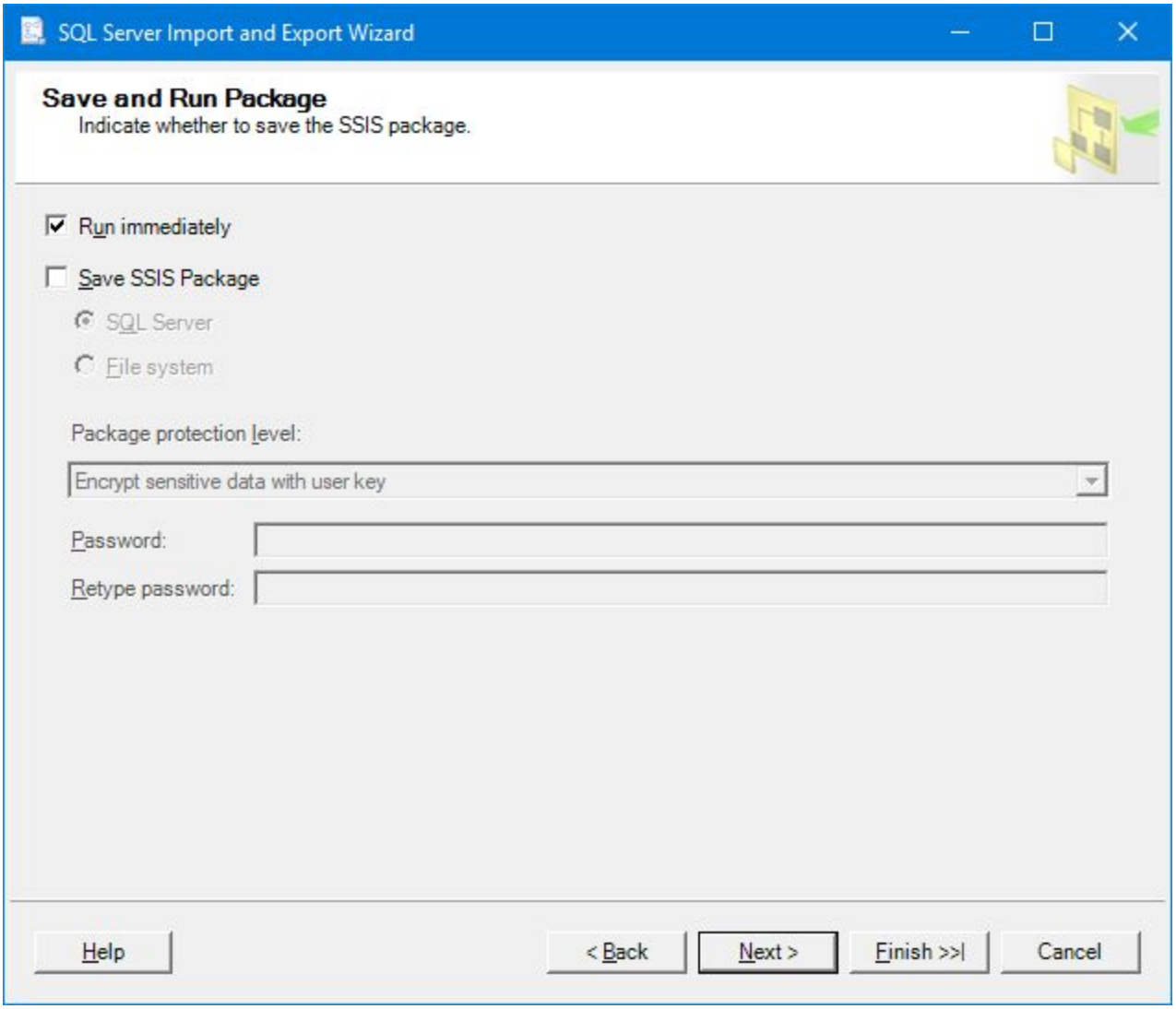


Select: [Sales].[Customer]

Note: These are the tables used in our SAP BusinessObjects Universe.

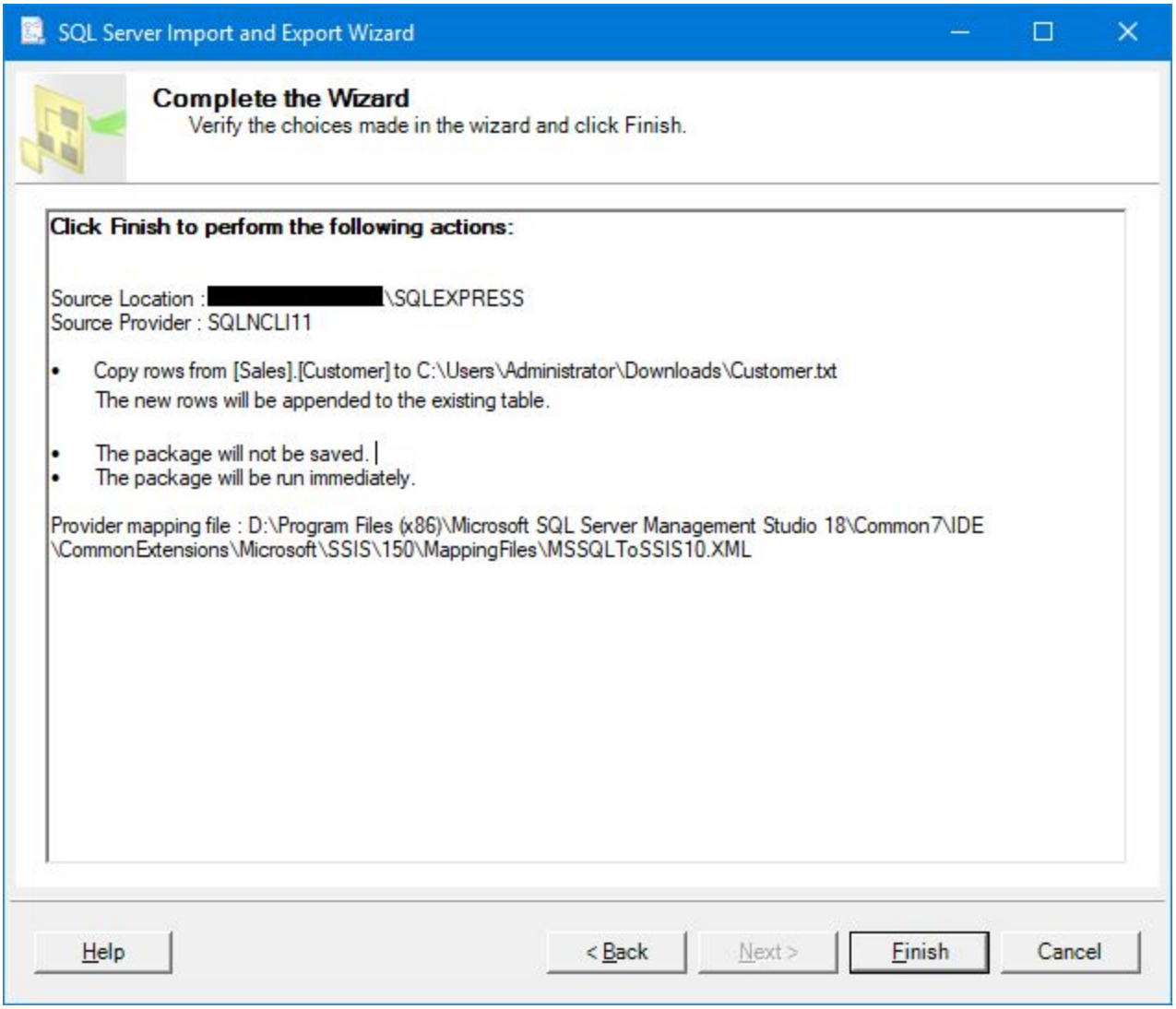
Click: Next

Save and Run Package



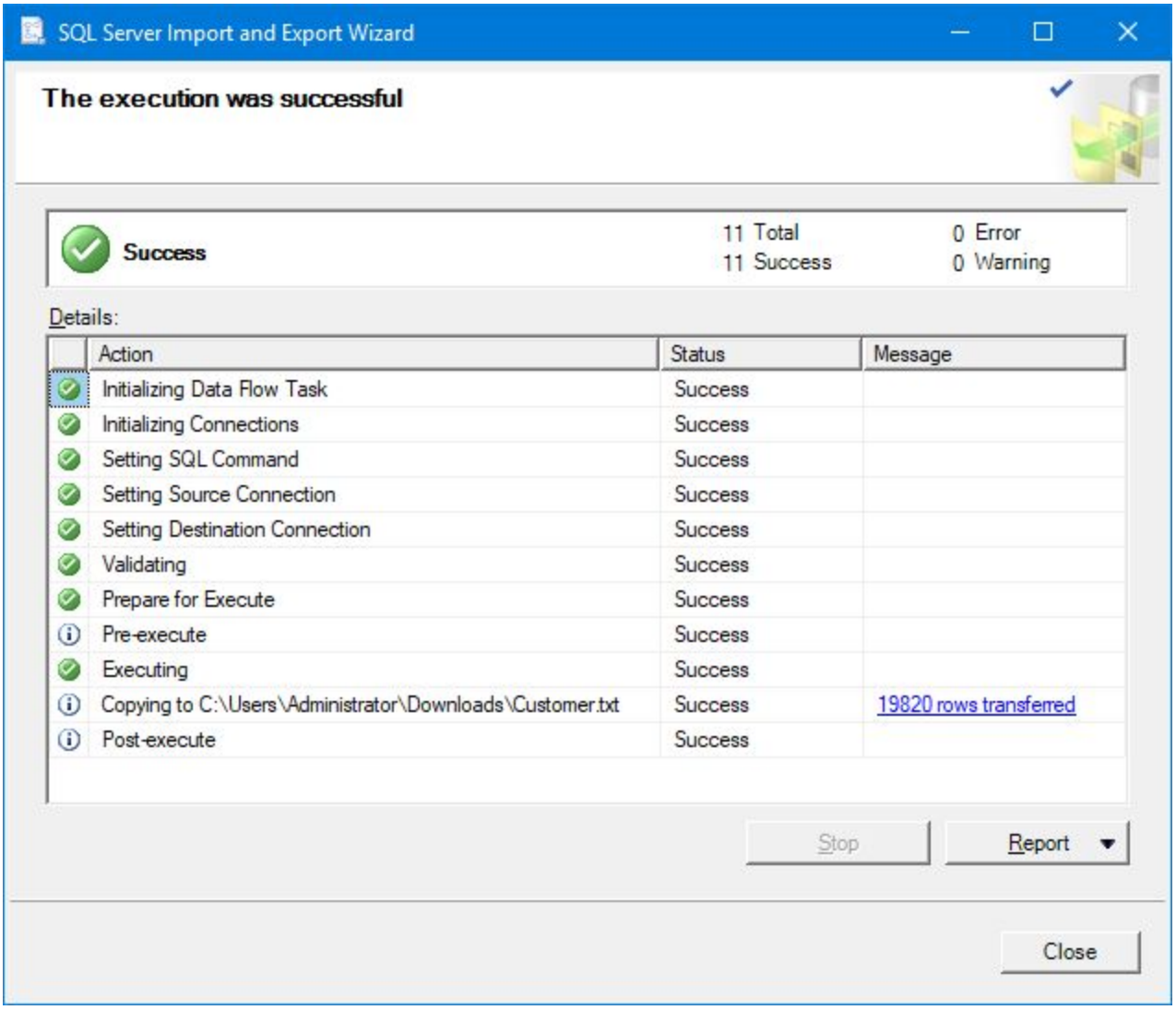
Click: Next

Complete the Wizard



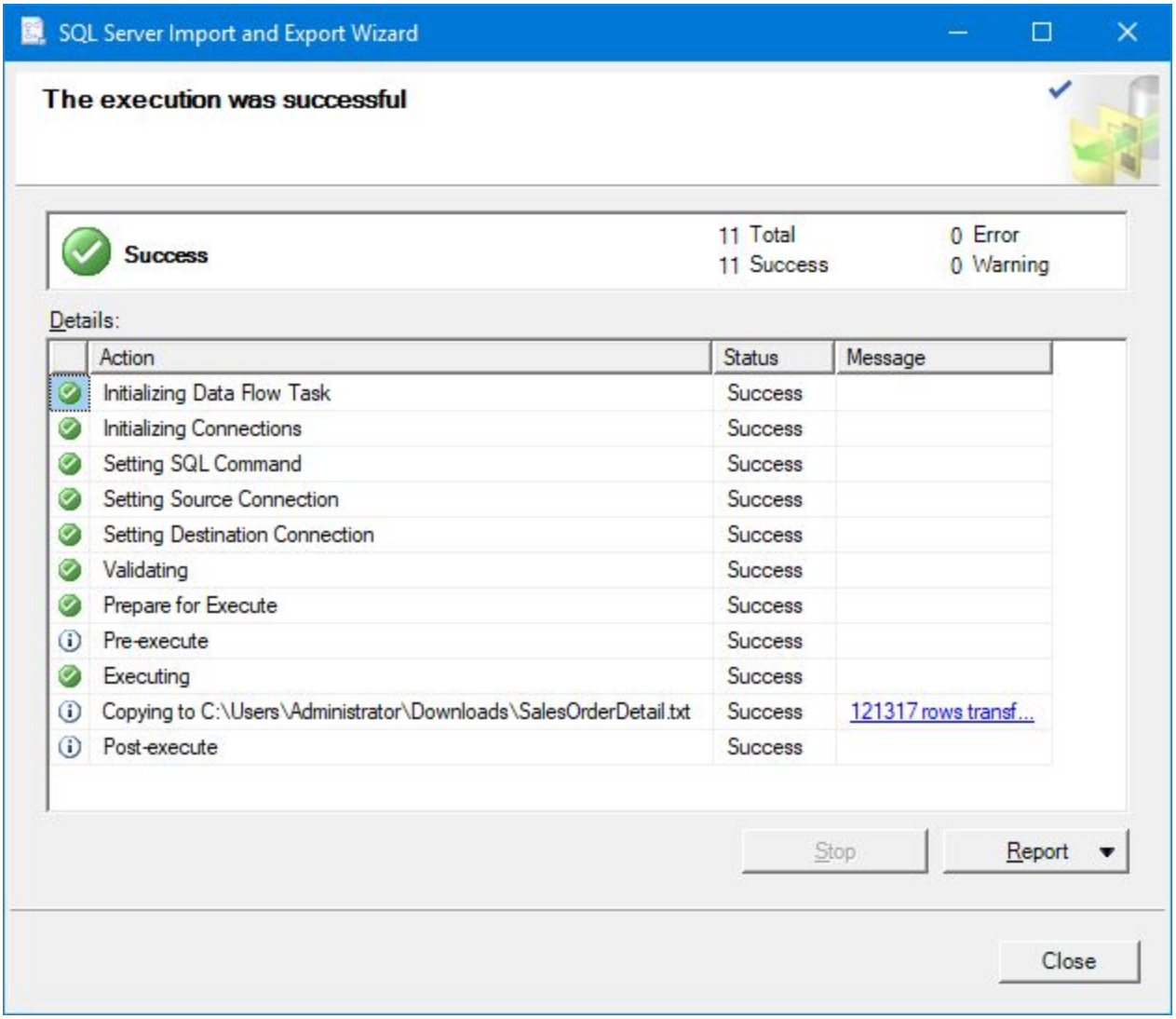
Click: Finish

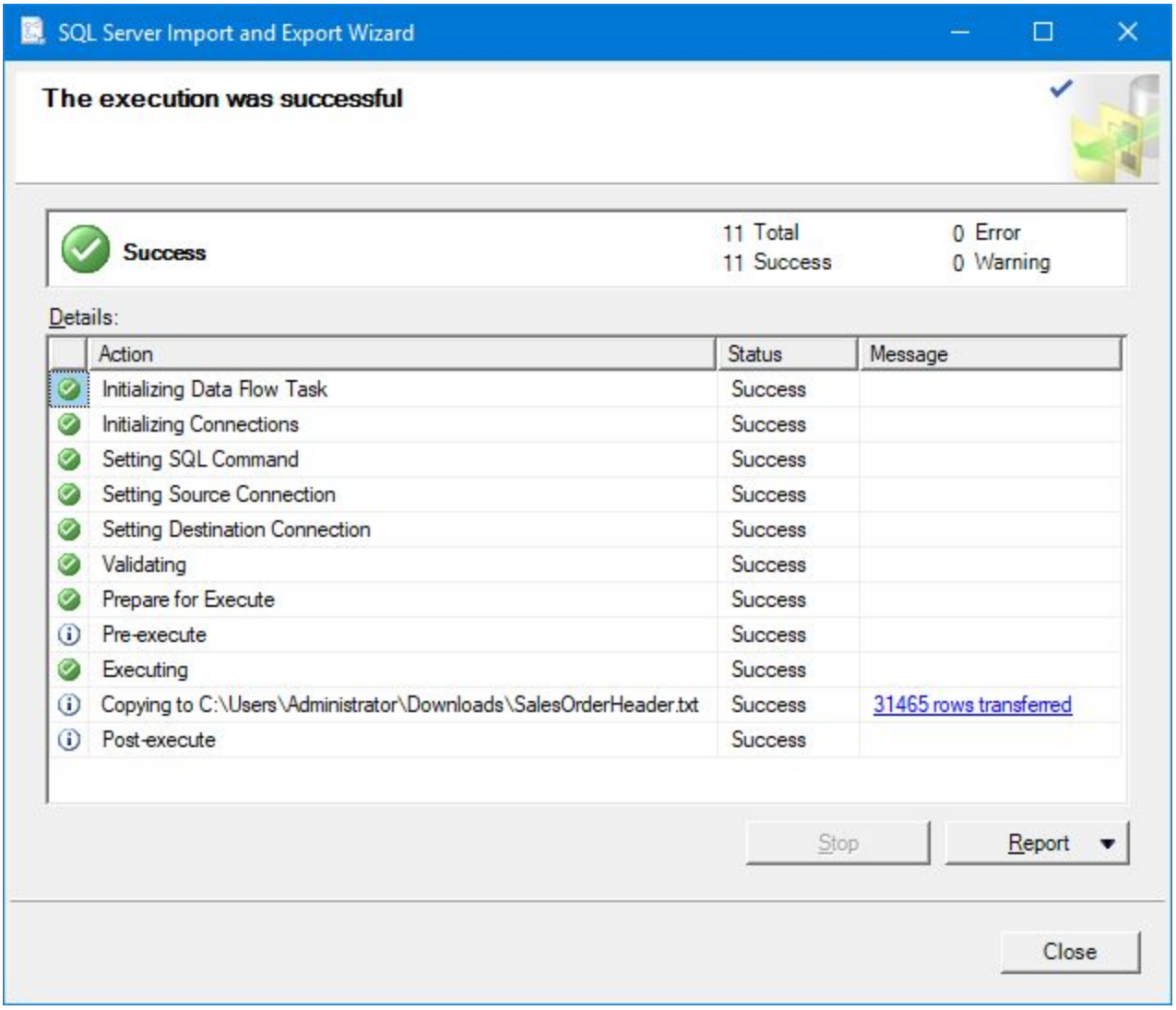
The execution was successful



Click: Close

Repeat for SalesOrderDetail and SalesOrderHeader

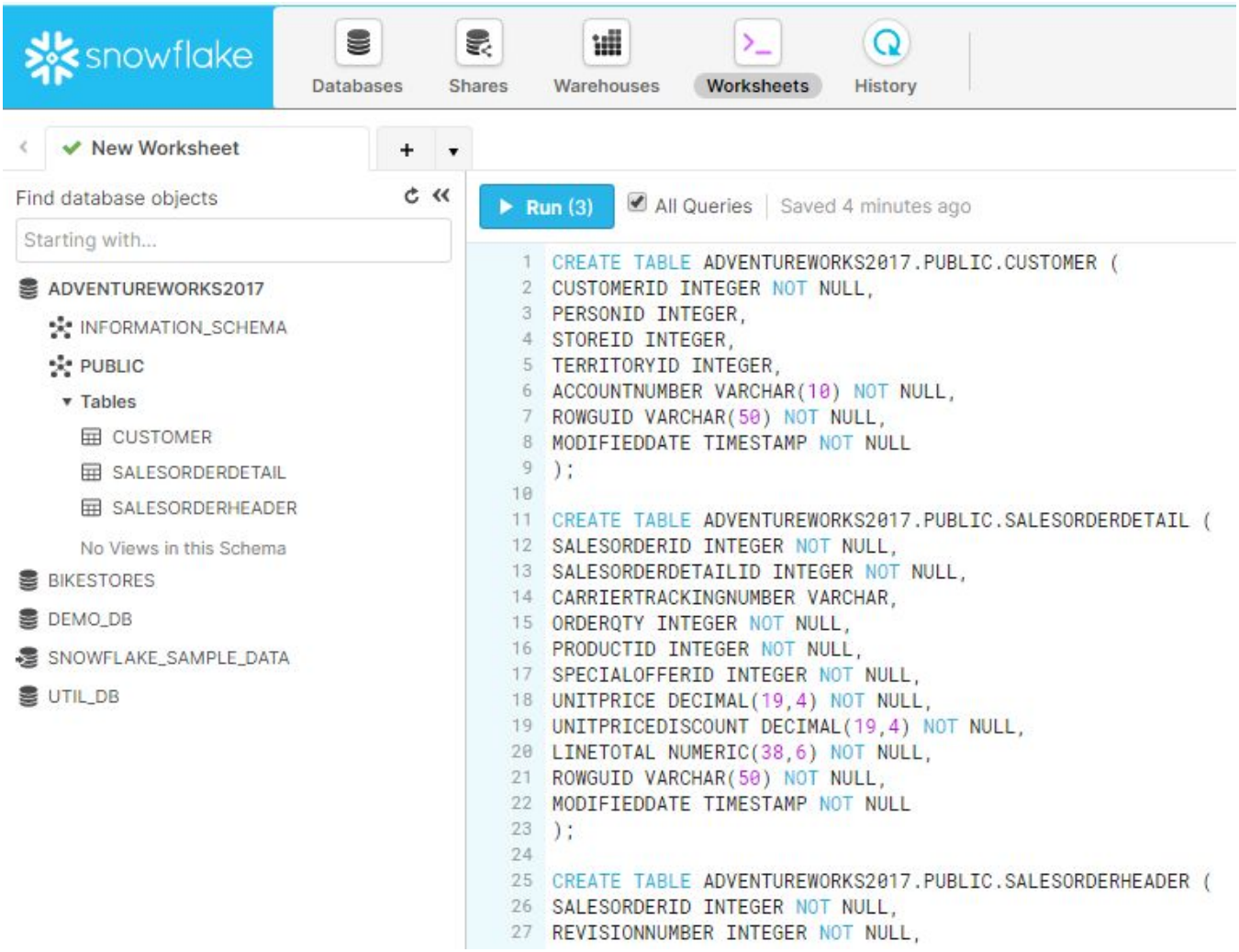



SQL Server Import and Export Wizard
The execution was successful
Success
11 Total
11 Success
0 Error
0 Warning
Details:
Action
Status
Message
Initializing Data Flow Task
Success
Initializing Connections
Success
Setting SQL Command
Success
Setting Source Connection
Success
Setting Destination Connection
Success
Validating
Success
Prepare for Execute
Success
Pre-execute
Success
Executing
Success
Copying to C:\Users\Administrator\Downloads\SalesOrderHeader.txt
Success
31465 rows transferred
Post-execute
Success
Stop
Report
Close


Customer.txt 30/03/2020 13:35 Text Document 1,852 KB
SalesOrderDetail.txt 30/03/2020 11:17 Text Document 12,648 KB
SalesOrderHeader.txt 30/03/2020 11:19 Text Document 7,323 KB

Create Tables in Snowflake

Logon to Snowflake
Click: Worksheets



Copy / Paste this SQL Query to create the new tables:

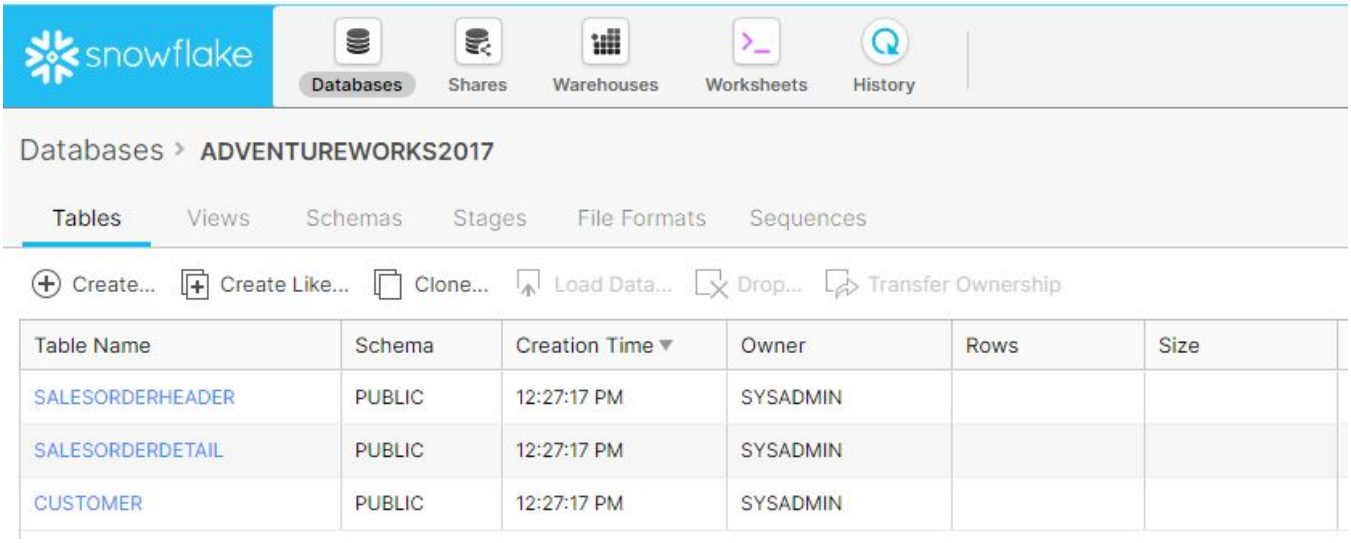
```
CREATE TABLE ADVENTUREWORKS2017.PUBLIC.CUSTOMER (
CUSTOMERID INTEGER NOT NULL,
PERSONID INTEGER,
STOREID INTEGER,
TERRITORYID INTEGER,
ACCOUNTNUMBER VARCHAR(10) NOT NULL,
ROWGUID VARCHAR(50) NOT NULL,
MODIFIEDDATE TIMESTAMP NOT NULL
);

CREATE TABLE ADVENTUREWORKS2017.PUBLIC.SALESORDERDETAIL (
SALESORDERID INTEGER NOT NULL,
SALESORDERDETAILID INTEGER NOT NULL,
CARRIERTRACKINGNUMBER VARCHAR,
ORDERQTY INTEGER NOT NULL,
PRODUCTID INTEGER NOT NULL,
SPECIALOFFERID INTEGER NOT NULL,
UNITPRICE DECIMAL(19,4) NOT NULL,
UNITPRICEDISCOUNT DECIMAL(19,4) NOT NULL,
LINETOTAL NUMERIC(38,6) NOT NULL,
ROWGUID VARCHAR(50) NOT NULL,
MODIFIEDDATE TIMESTAMP NOT NULL
);

CREATE TABLE ADVENTUREWORKS2017.PUBLIC.SALESORDERHEADER (
SALESORDERID INTEGER NOT NULL,
REVISIONNUMBER INTEGER NOT NULL,
ORDERDATE TIMESTAMP NOT NULL,
DUEDATE TIMESTAMP NOT NULL,
SHIPDATE TIMESTAMP,
STATUS INTEGER NOT NULL,
ONLINEORDERFLAG BOOLEAN NOT NULL,
SALESORDERNUMBER VARCHAR NOT NULL,
PURCHASEORDERNUMBER VARCHAR,
ACCOUNTNUMBER VARCHAR,
CUSTOMERID INTEGER NOT NULL,
```

```
SALESPERSONID INTEGER,
TERRITORYID INTEGER,
BILLTOADDRESSID INTEGER NOT NULL,
SHIPTOADDRESSID INTEGER NOT NULL,
SHIPMETHODID INTEGER NOT NULL,
CREDITCARDID INTEGER,
CREDITCARDAPPROVALCODE VARCHAR(15),
CURRENCYRATEID INTEGER,
SUBTOTAL DECIMAL(19,4) NOT NULL,
TAXAMT DECIMAL(19,4) NOT NULL,
FREIGHT DECIMAL(19,4) NOT NULL,
TOTALDUE DECIMAL(19,4) NOT NULL,
COMMENT VARCHAR,
ROWGUID VARCHAR(50) NOT NULL,
MODIFIEDDATE TIMESTAMP NOT NULL
);
```

Click: Run

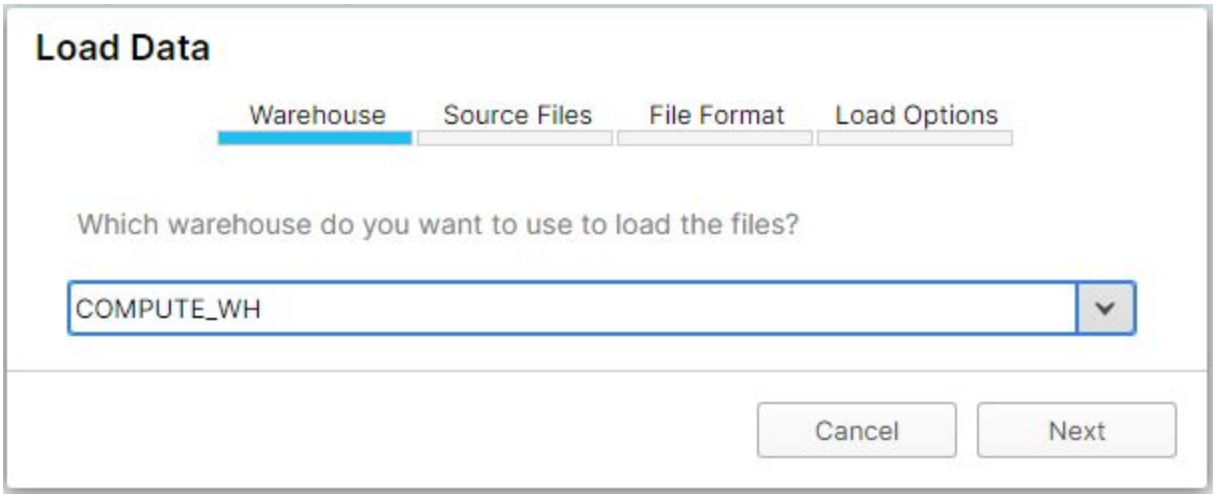
Import Data via CSV into Snowflake

Logon to Snowflake
Click: Databases
Click: ADVENTUREWORKS2017



Select: CUSTOMER
Click: Load Data...

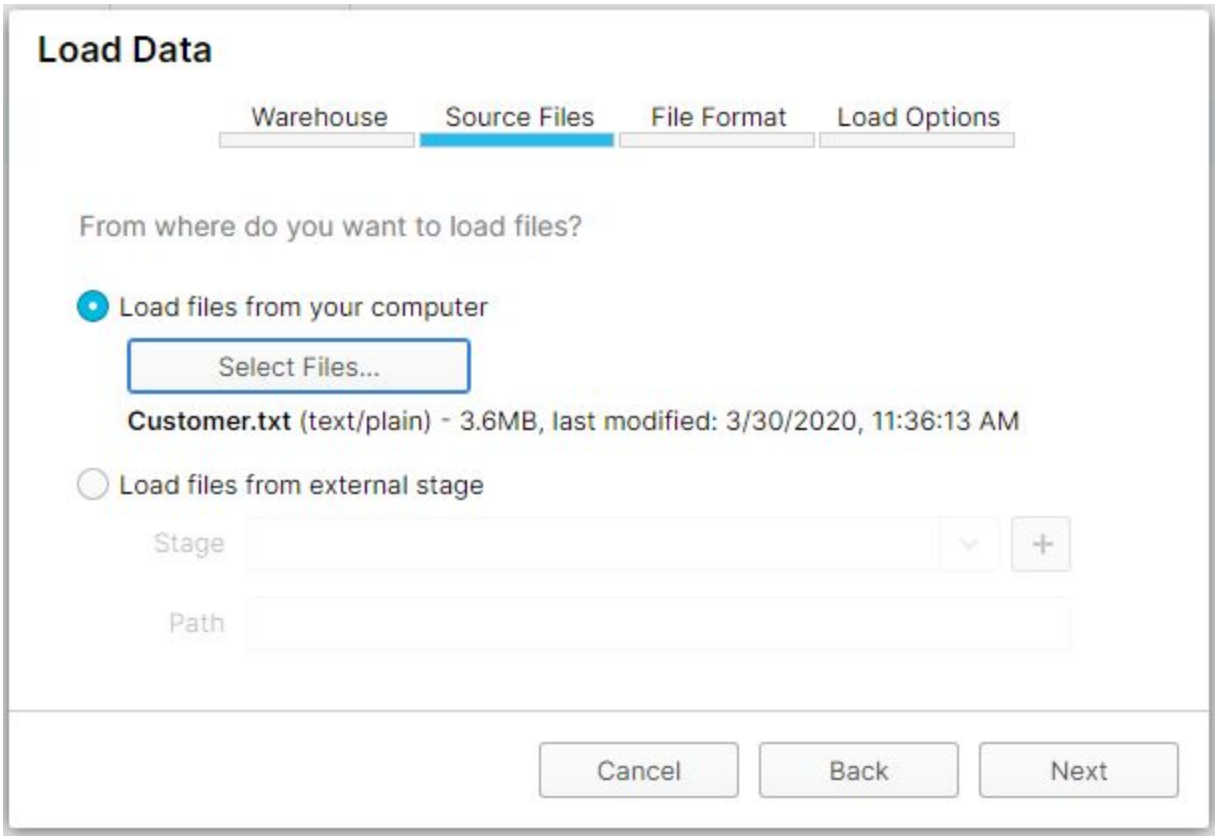
Load Data - Warehouse



Click: Next

Load Data - Source Files

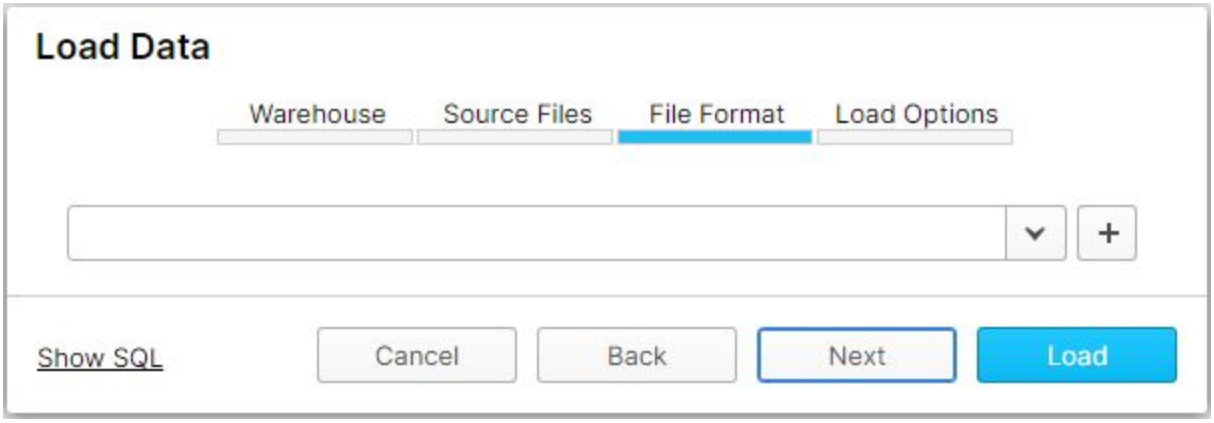
Click: Select Files...
Browse: customer.txt



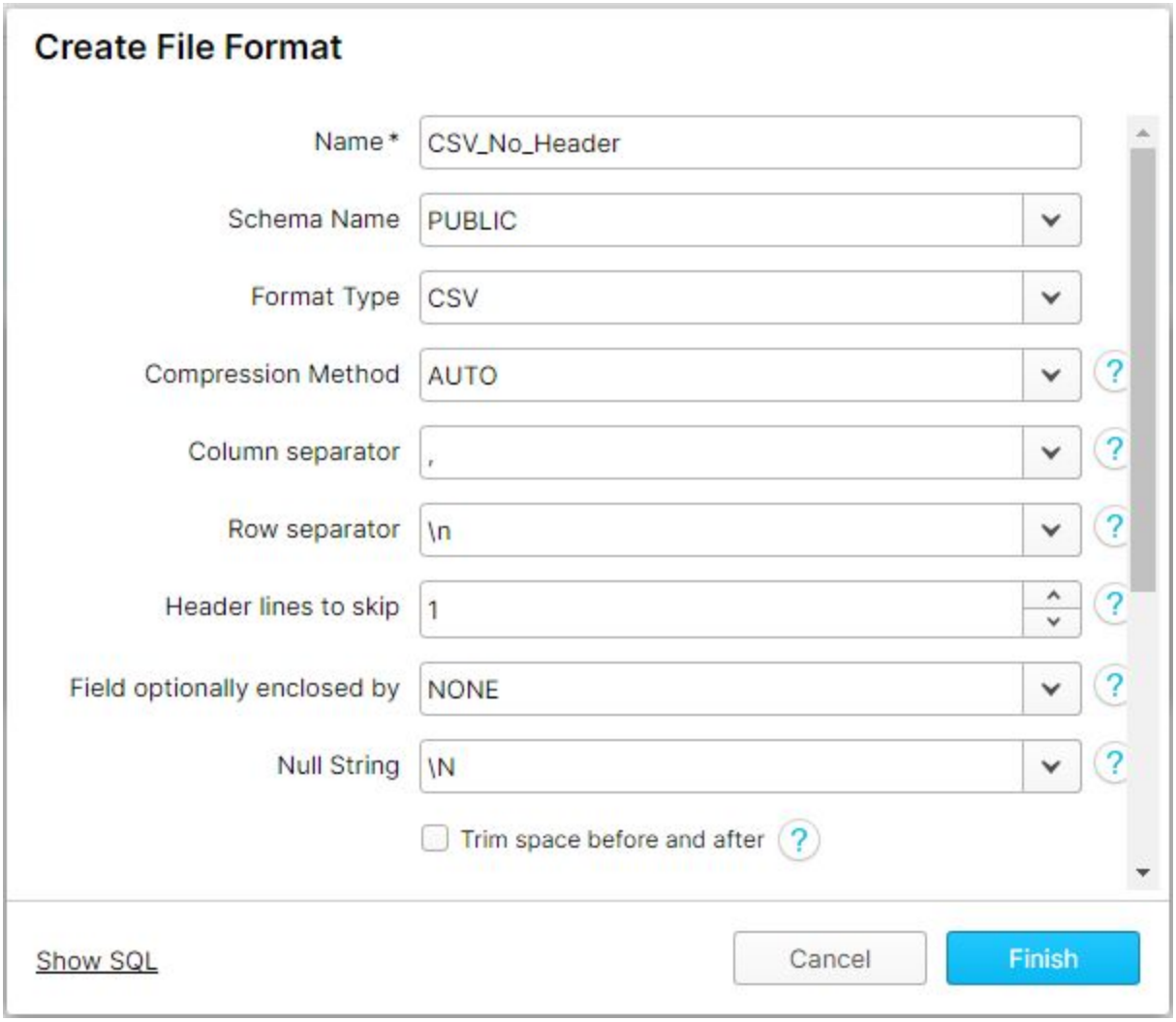
Click: Next

Load Data - File Format

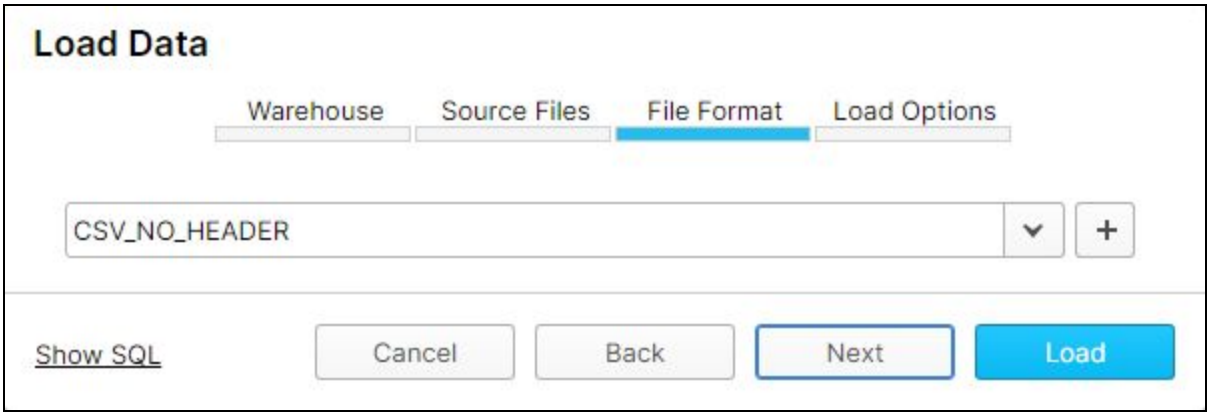
Click: +



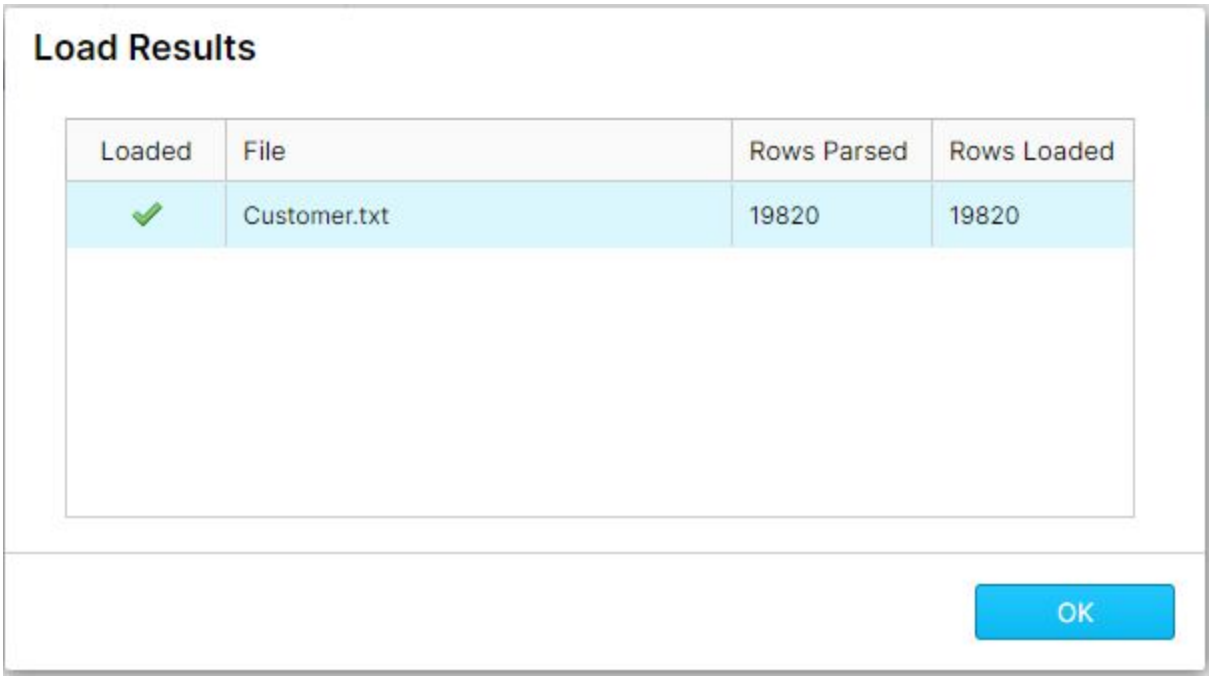
Name: [ENTER A NAME]
Header lines to skip: Change 0 to 1



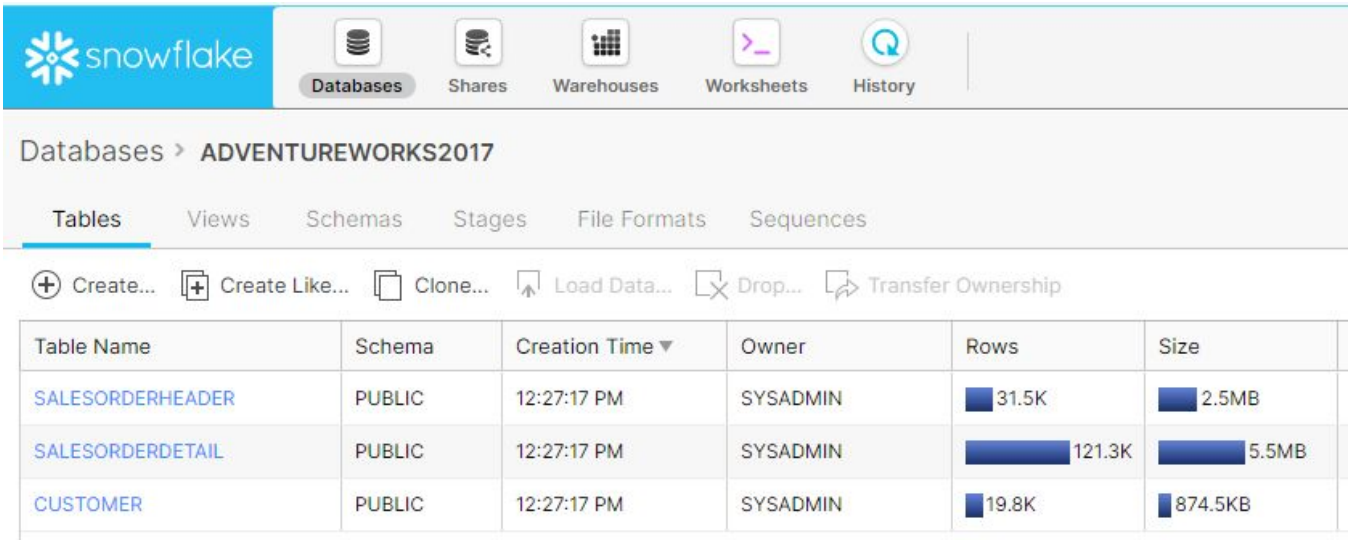
Click: Finish


Load Data
Warehouse
Source Files
File Format
Load Options
CSV_NO_HEADER
Show SQL
Cancel
Back
Next
Load

Click: Load


Load Results
Loaded
File
Rows Parsed
Rows Loaded
Customer.txt
19820
19820
OK

Repeat for SALESORDERDETAIL and SALESORDERHEADER



What's Next?

Now that we have our data into Snowflake, we need to work with SAP BusinessObjects to make its content e.g.: Universes and Connections, Web Intelligence and Crystal Reports point to the new data source.

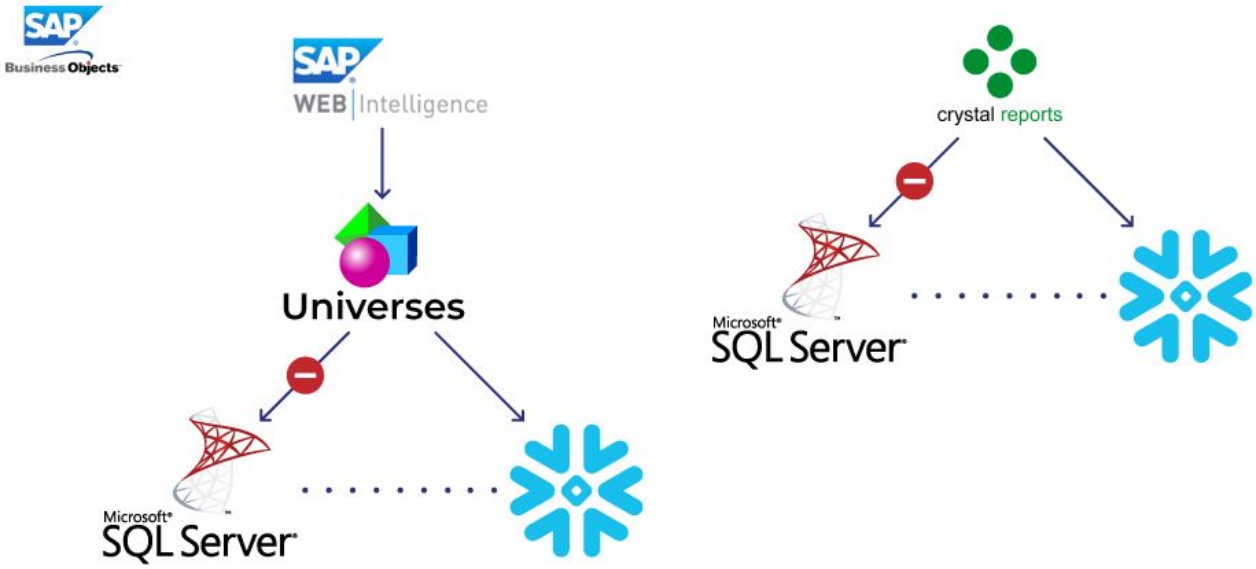
As mentioned in the introduction, some of you will only be required to repoint the Universe Connection to Snowflake. This may be true if there are no changes to the owners, qualifiers, schema or presence of vendor specific SQL in the Universes.

Others for reasons mentioned above will need first to make a copy of the existing Universe to not affect Production before repointing it to Snowflake. Next is to repair the Universes where required. Finally and based on your scenario, you may have to repoint all content to this new Universe.

The next sections will demonstrate this use case. Not all steps may be applicable to your project.

These steps can be done manually and/or via automated solutions by 360Suite to reduce time, cost, and risks.

Finally, it is important to perform enough functional, data, and performance testing to ensure the project is successful.



Updating SAP BusinessObjects

Copying Universes

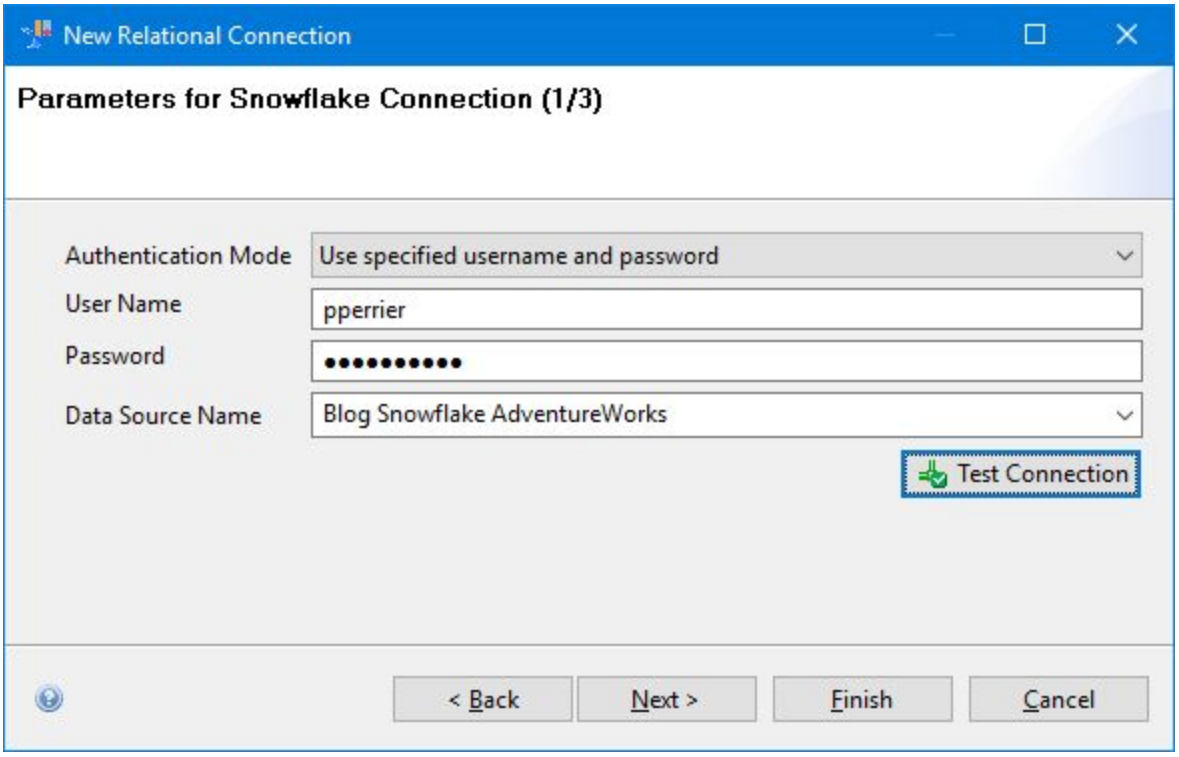
The first step is to create a copy of the existing *AdventureWorks2017* currently pointing to Microsoft SQL Server and make it use the new Snowflake database.

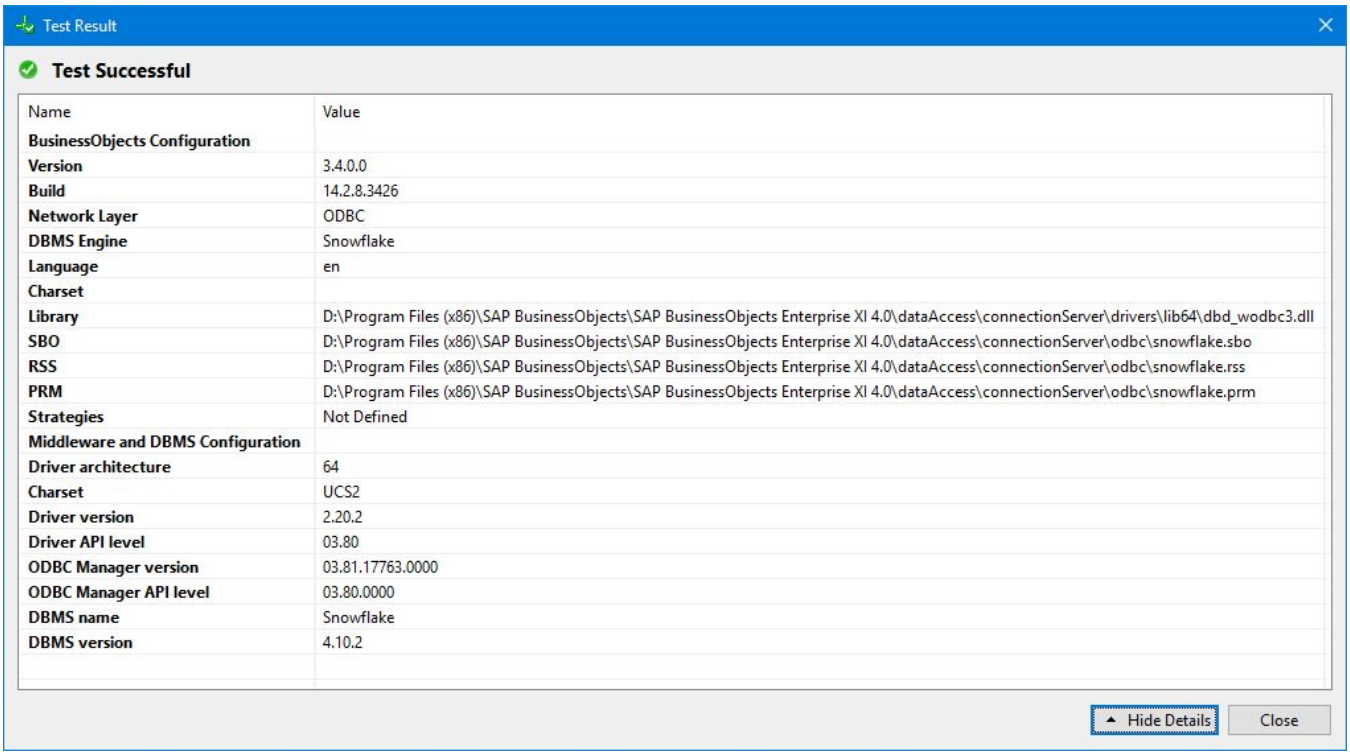
By the end of this section, before you repoint your documents (e.g.: Web Intelligence, Crystal Reports) you want to make sure the Universe is working correctly by performing a *Check Integrity*.

It may highlight vendor specific SQL syntax that won't work with Snowflake. Issues with data type, etc. In case there are a lot of objects to repair, we suggest doing this in bulk to save time and avoid mistakes using 360Univ.

Create a Universe Connection to Snowflake

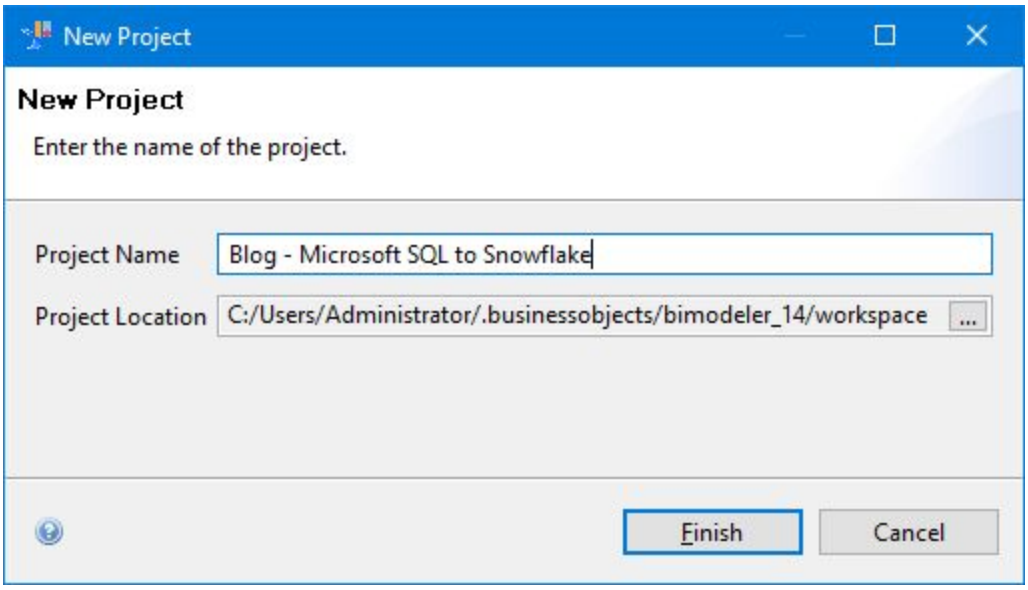
This step can be done using the 32-bit ODBC or JDBC connection you did earlier in this document.





Note: See this blog for more details:
https://blogs.sap.com/2020/03/12/snowflake-for-sap-businessobjects-4.2-sp08/

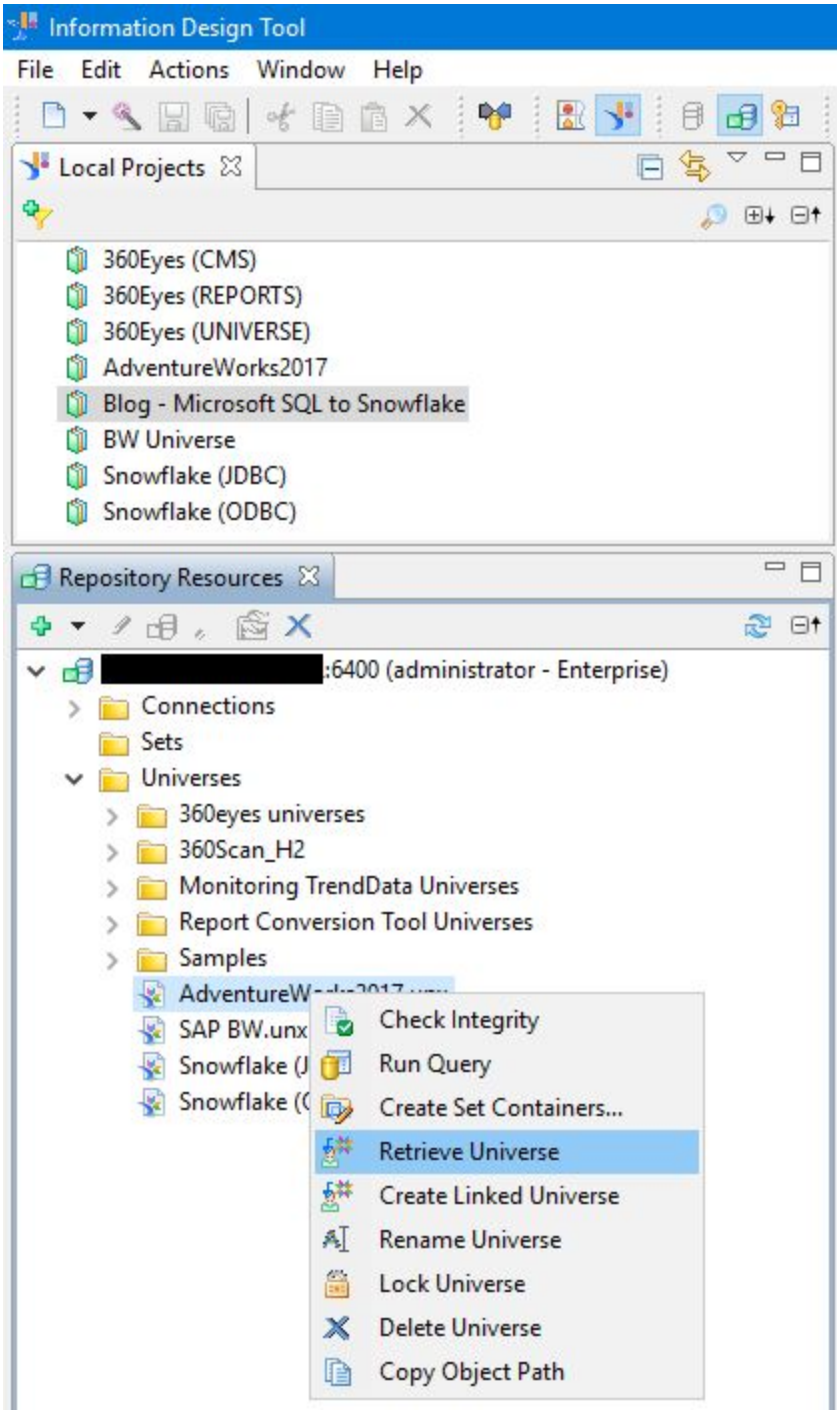
Create a New Local Project



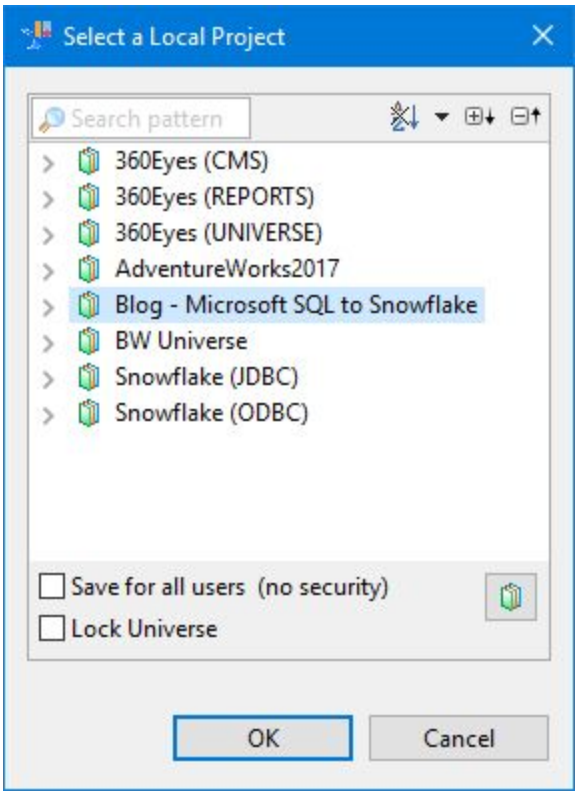
Project Name: Blog - Microsoft SQL to Snowflake

Click: Finish

Retrieve the Universe

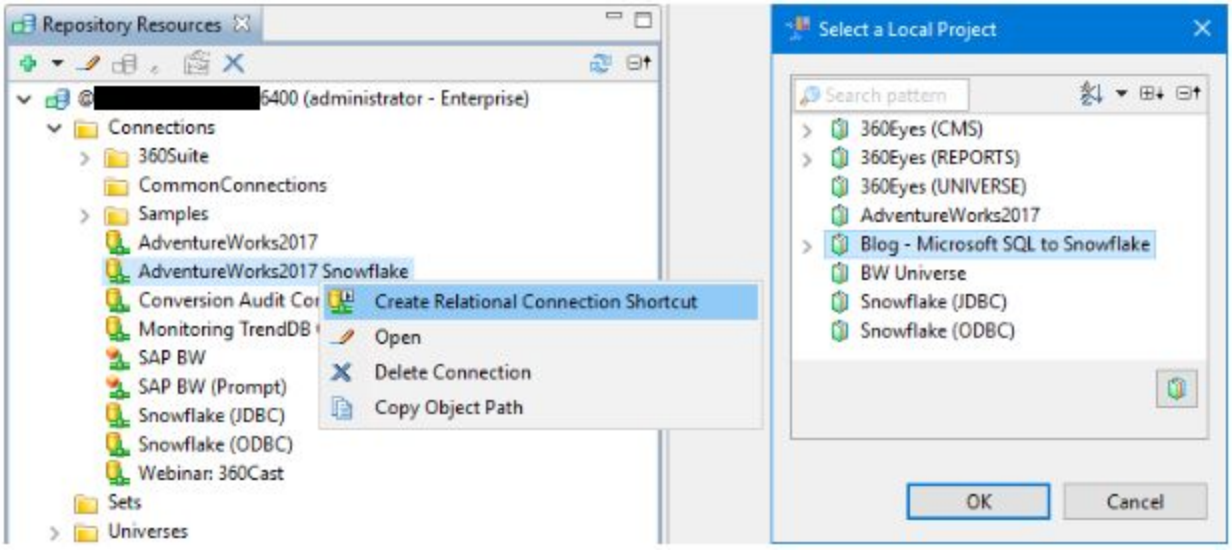


Select a Local Project



Click: OK

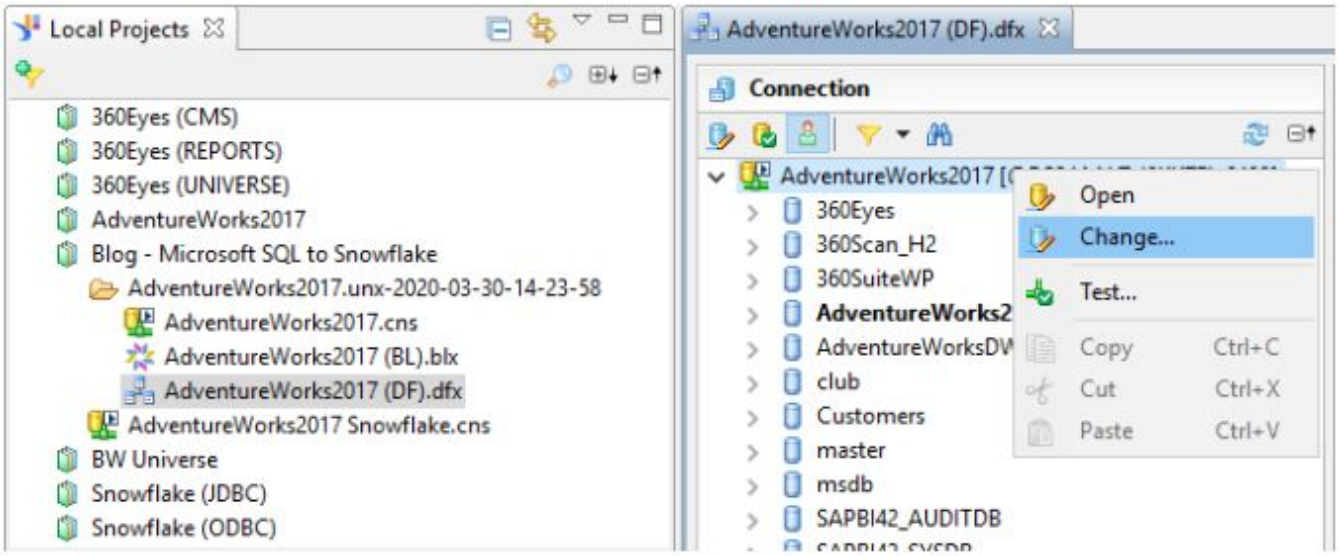
Create a Relational Connection Shortcut



1. Right-Click AdventureWorks2017 Snowflake
2. Click: Create Relational Connection Shortcut
3. Select a Local Project: Blog - Microsoft SQL to Snowflake
4. Click: OK

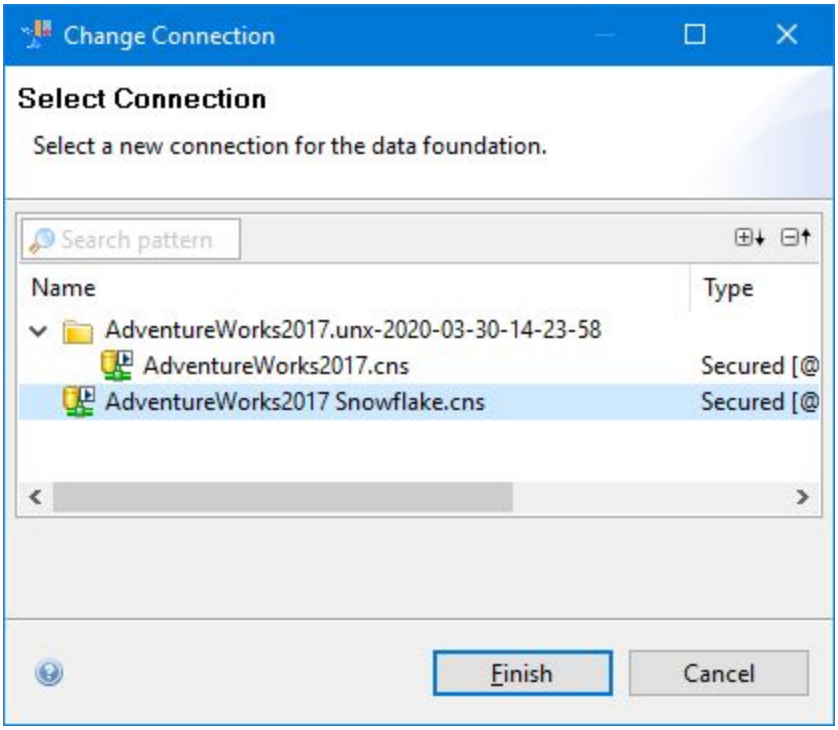
Repoint the Data Foundation to the Snowflake Connection

1. Open AdventureWorks2017 (DF).dfx



Under Connection > Right-Click AdventureWorks2017
Click: Change...

2. Select: AdventureWorks2017 Snowflake.cns

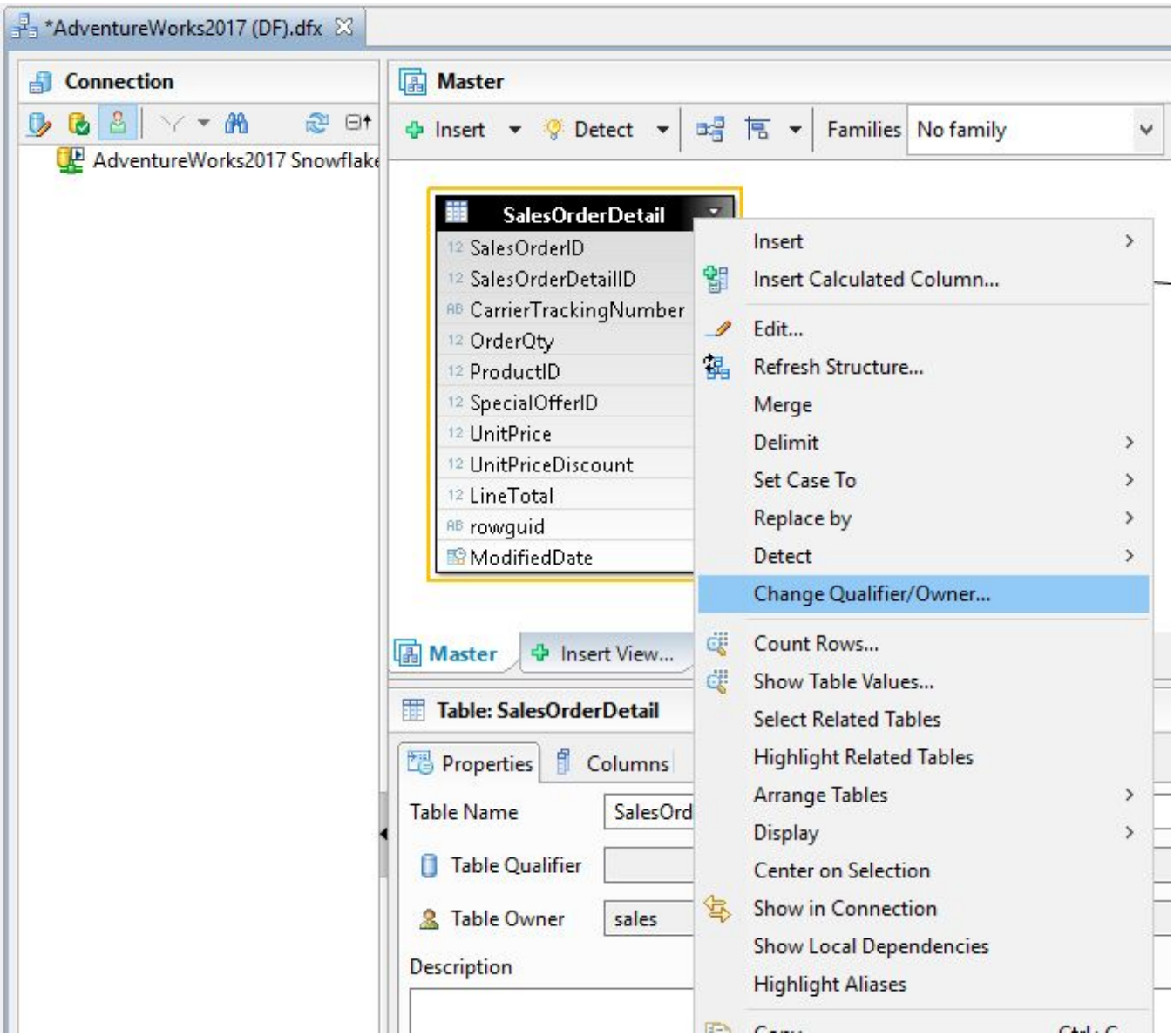


3. Click: Finish

4. Save the Data Foundation

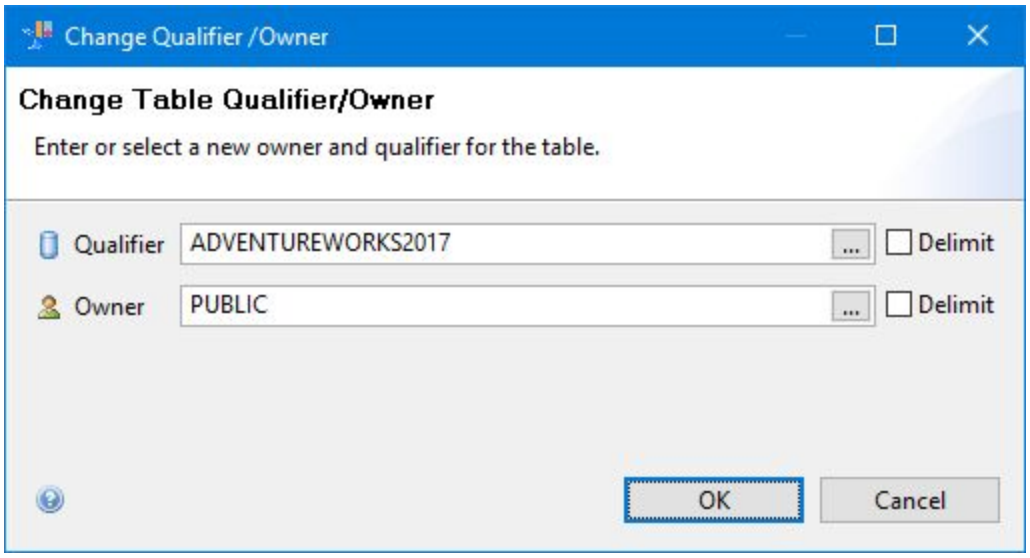
Change the Qualifier/Owner of the Tables in the Data Foundation

1. Open AdventureWorks2017 (DF).dfx



Under Master > Right-Click: SalesOrderDetail table
Select Change Qualifier/Owner...

2. Change Table Qualifier/Owner



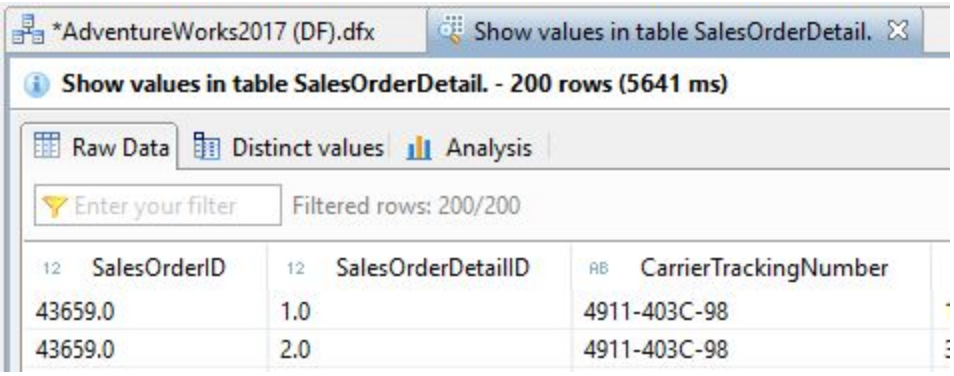
Enter Qualifier: ADVENTUREWORKS2017
Enter Owner: PUBLIC
Click: OK

3. Repeat for tables: SalesOrderHeader and Customer

   Note: You can multiple select tables and change qualifiers in bulk.

4. Save the Data Foundation

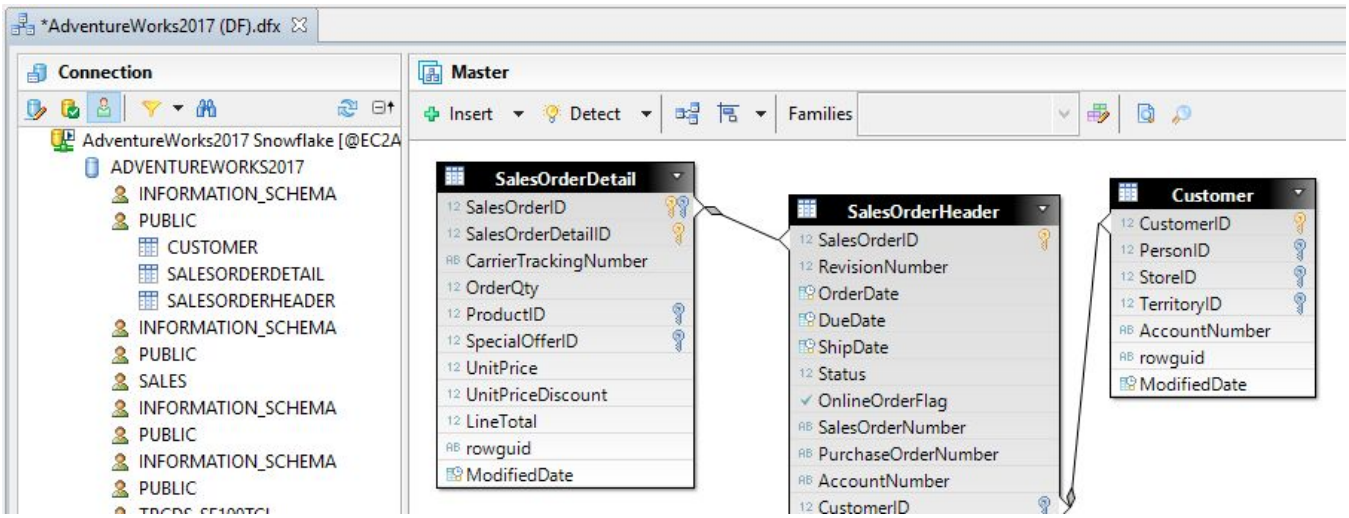
Note: At this stage you are able to preview data from the tables in the Data Foundation using “Show Table Values”.


*AdventureWorks2017 (DF).dfx
Show values in table SalesOrderDetail.
Show values in table SalesOrderDetail. - 200 rows (5641 ms)
Raw Data
Distinct values
Analysis
Enter your filter
Filtered rows: 200/200
SalesOrderID
SalesOrderDetailID
CarrierTrackingNumber
43659.0
1.0
4911-403C-98
43659.0
2.0
4911-403C-98

Set the Tables and Columns Case

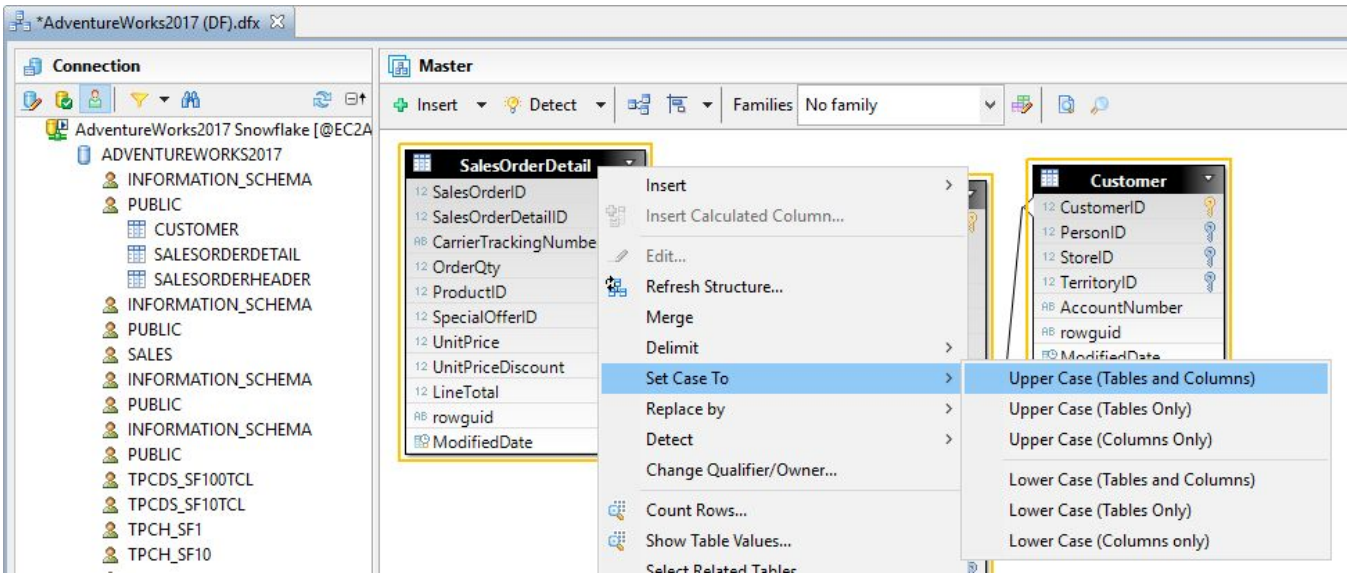
Although you can now preview data, the Information Design Tool doesn't correctly identify the tables in uppercase under Connection (left) with the tables in mixed case under Master (right).

As mentioned in the introduction, this is because the identifiers in Snowflake are case-insensitive but displayed in uppercase.



The following step could be seen as optional as at this stage, the Universe is functional. But if you check the integrity of the Universe it will fail because of this.

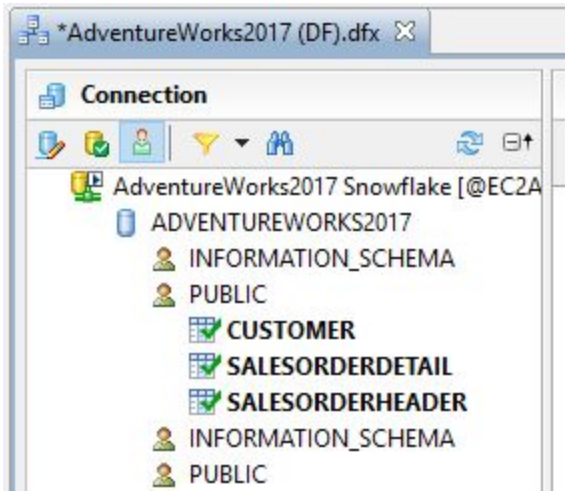
5. Open AdventureWorks2017 (DF).dfx



Under Master > Right-Click: SalesOrderDetail table
Click: Set Cast To > Upper Case (Tables and Columns)

6. Repeat for tables: SalesOrderHeader and Customer

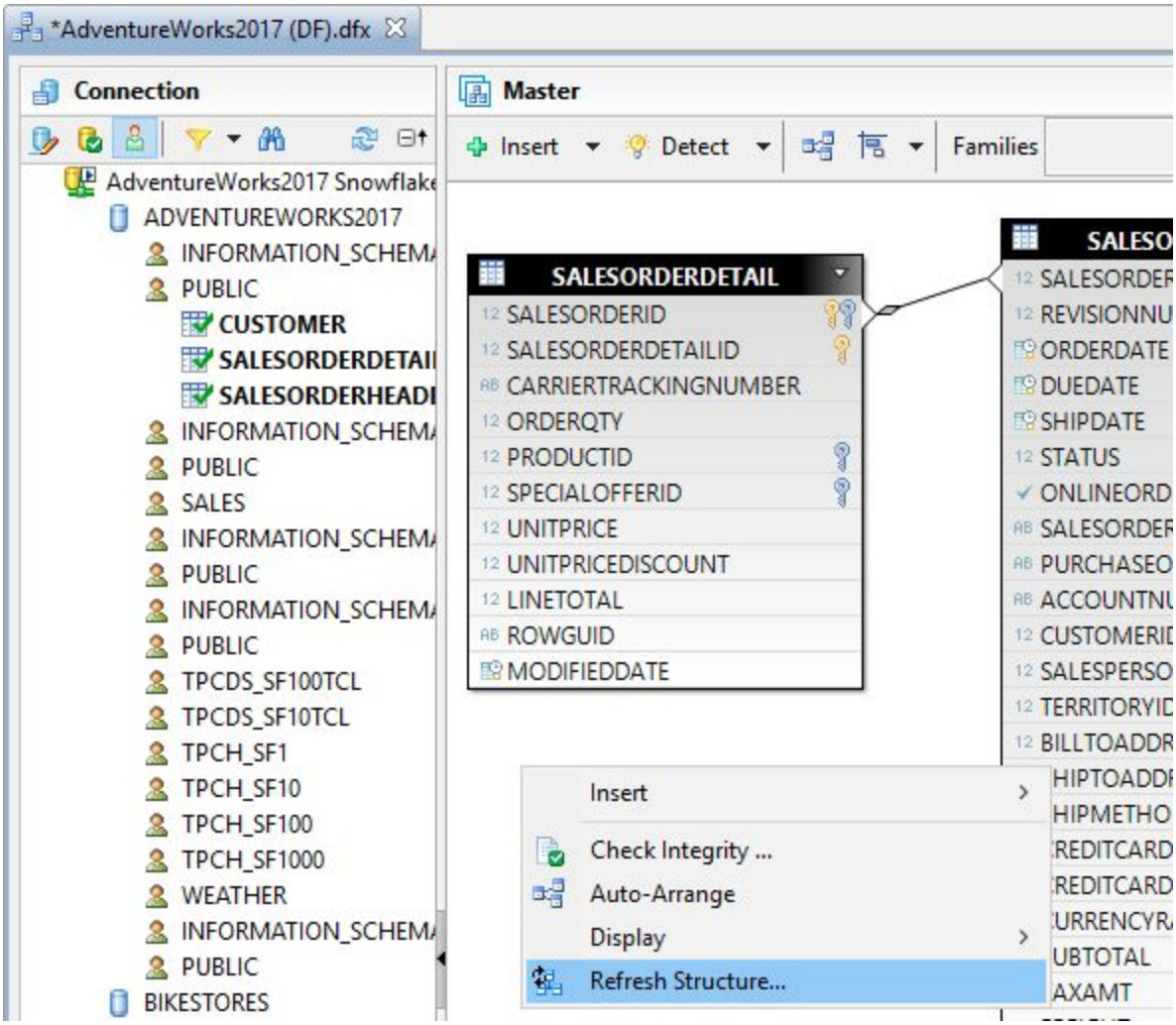
   Note: You can multiple select tables and change qualifiers in bulk.



7. Save the Data Foundation

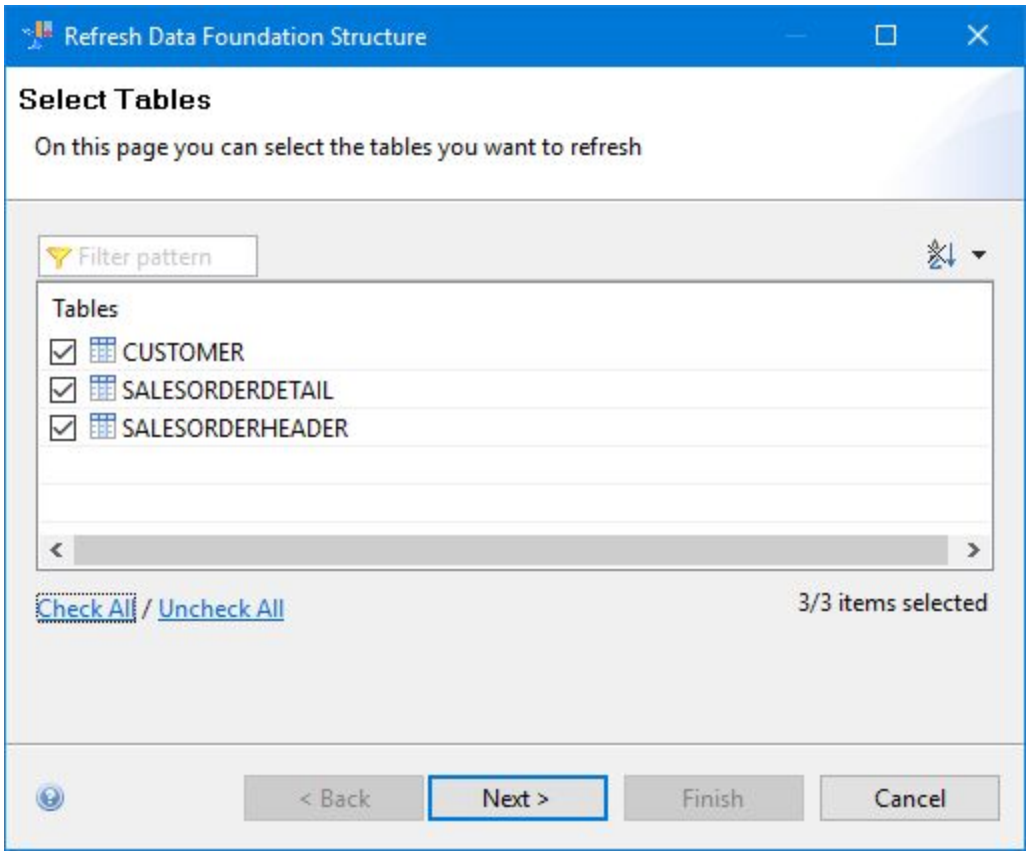
Refresh Universe Foundation Structure

1. Open AdventureWorks2017 (DF).dfx



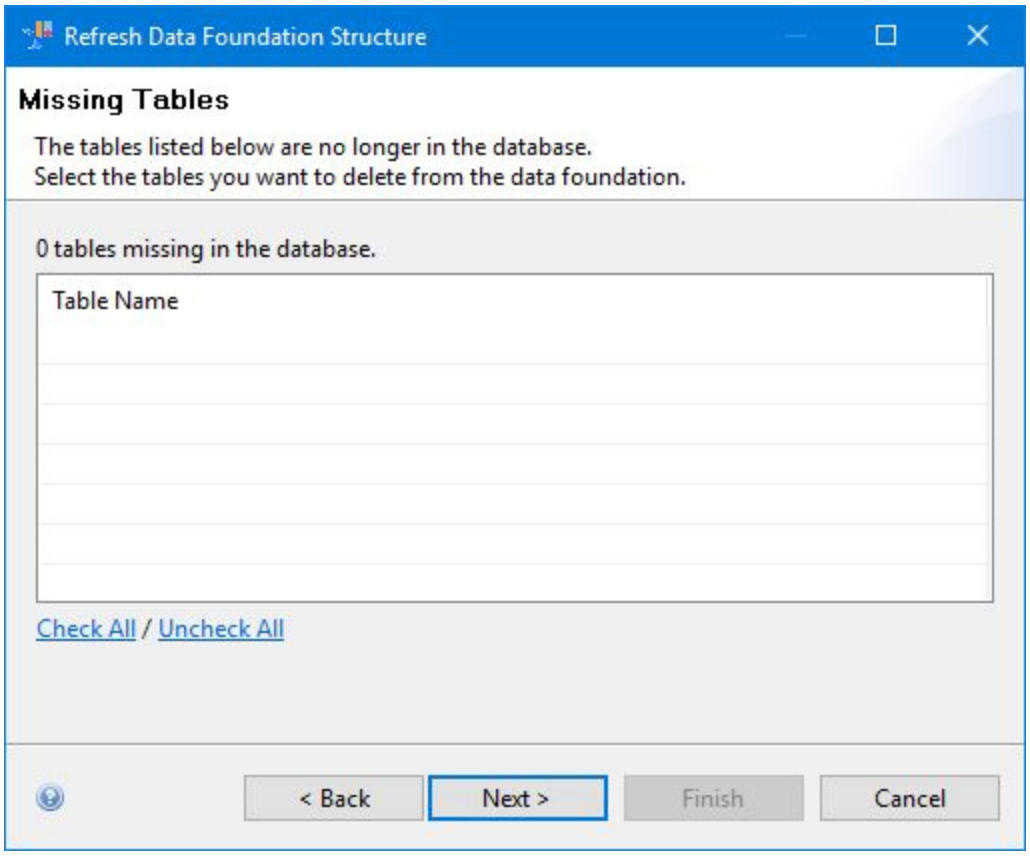
Under Master > Right-Click in the white area
Click: Refresh Structure...

2. Select Tables



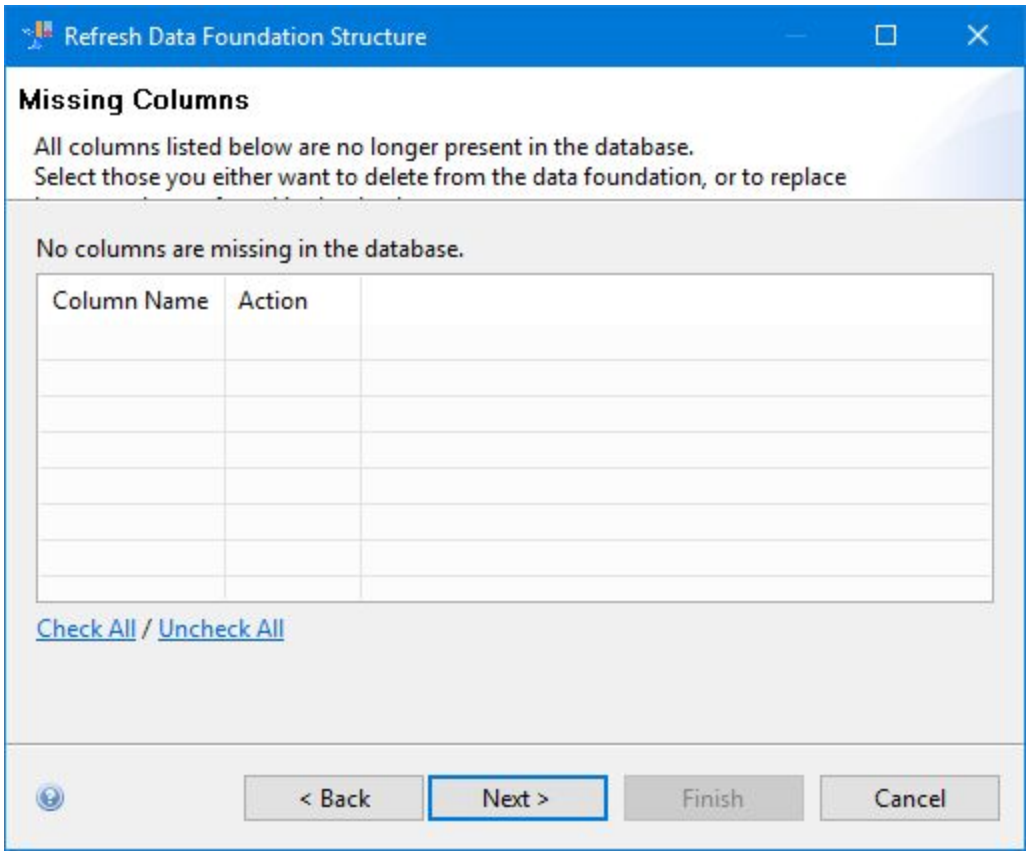
Click: Next

3. Missing Tables



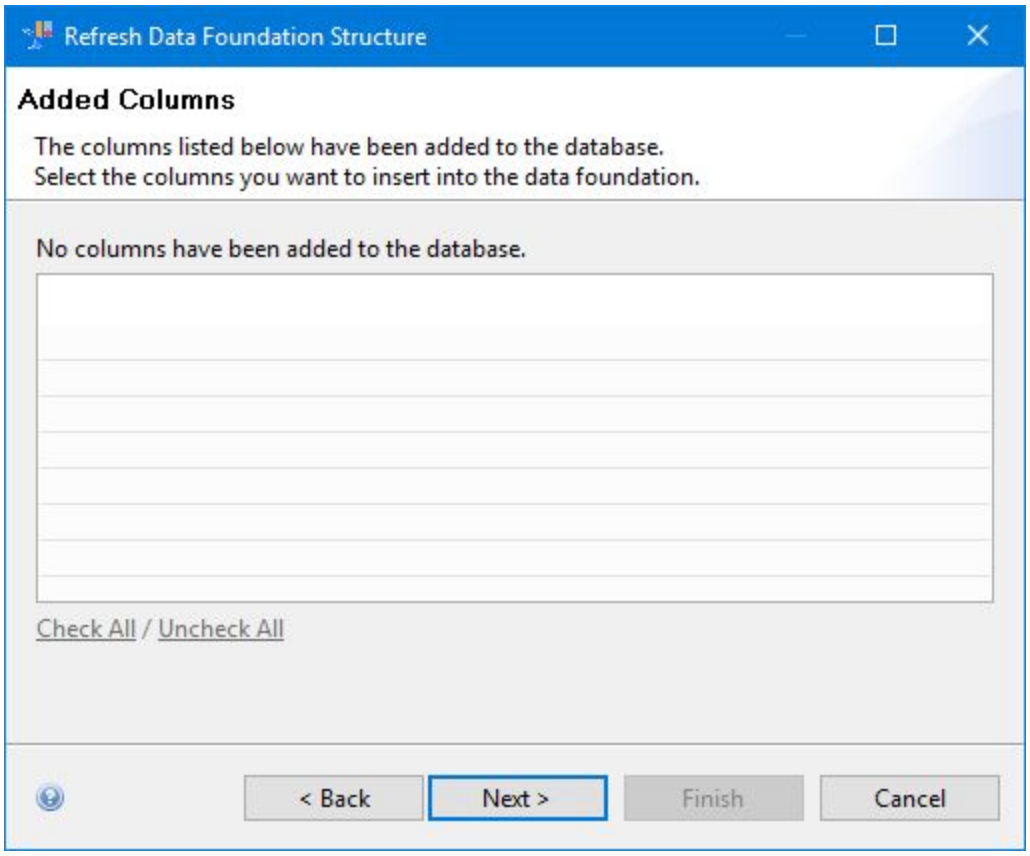
Click: Next

4. Missing Columns



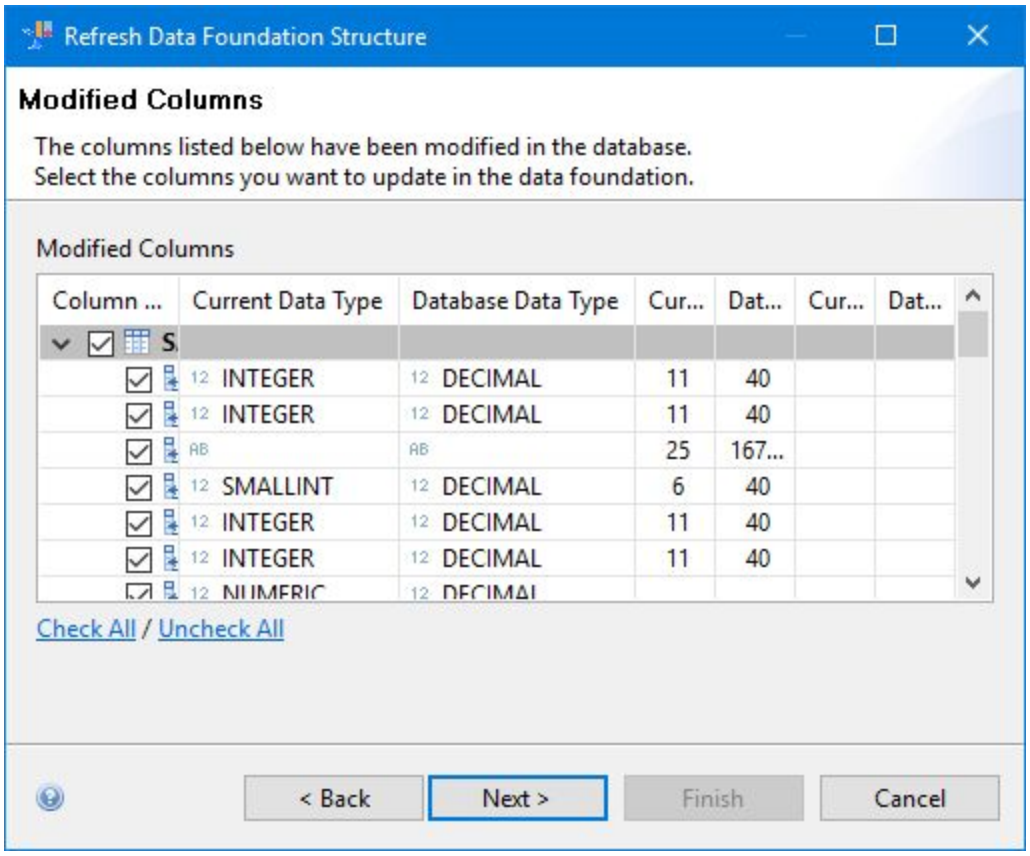
Click: Next

5. Added Columns



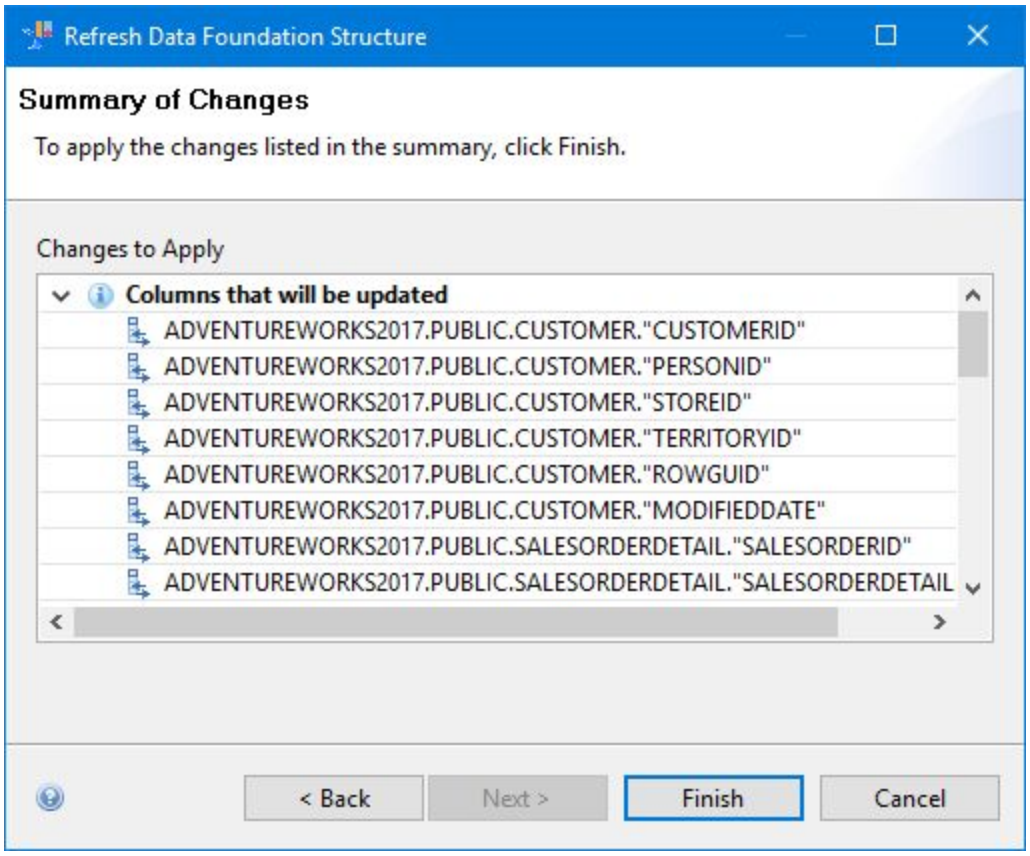
Click: Next

6. Modified Columns



Click: Next

7. Summary of Changes

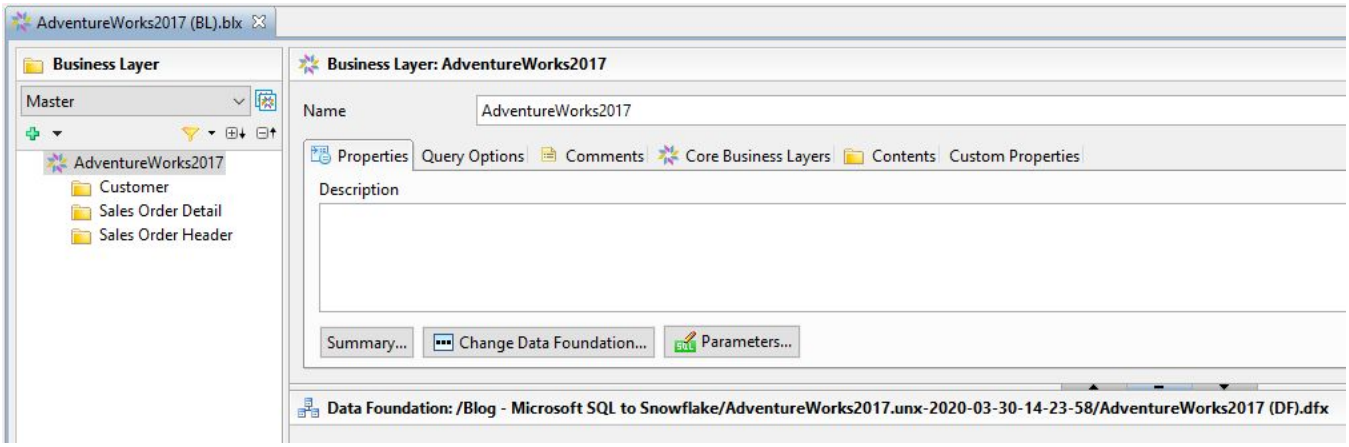


Click: Finish

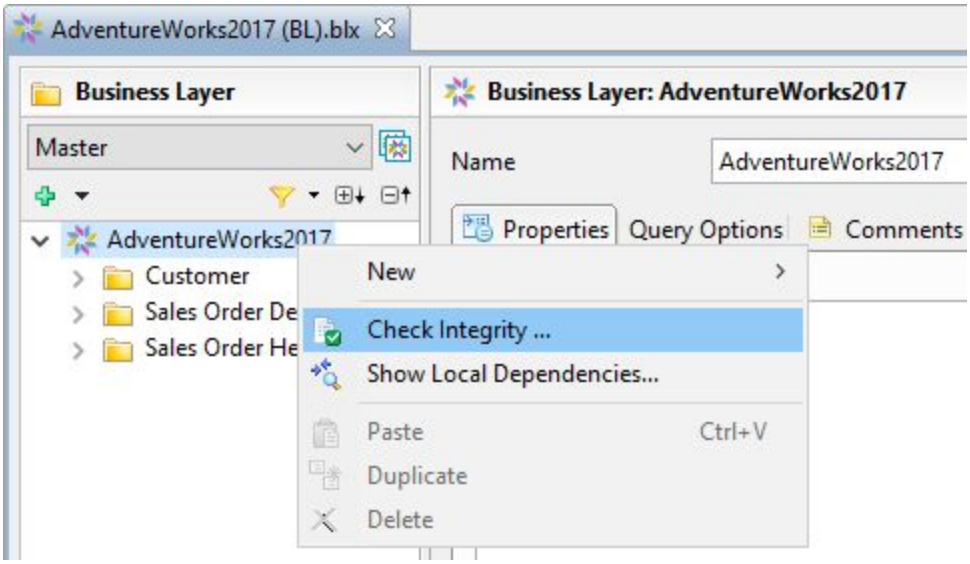
8. Save the Data Foundation

Validate the Business Layer with the Snowflake Data Foundation

1. Open AdventureWorks2017 (BL).blx

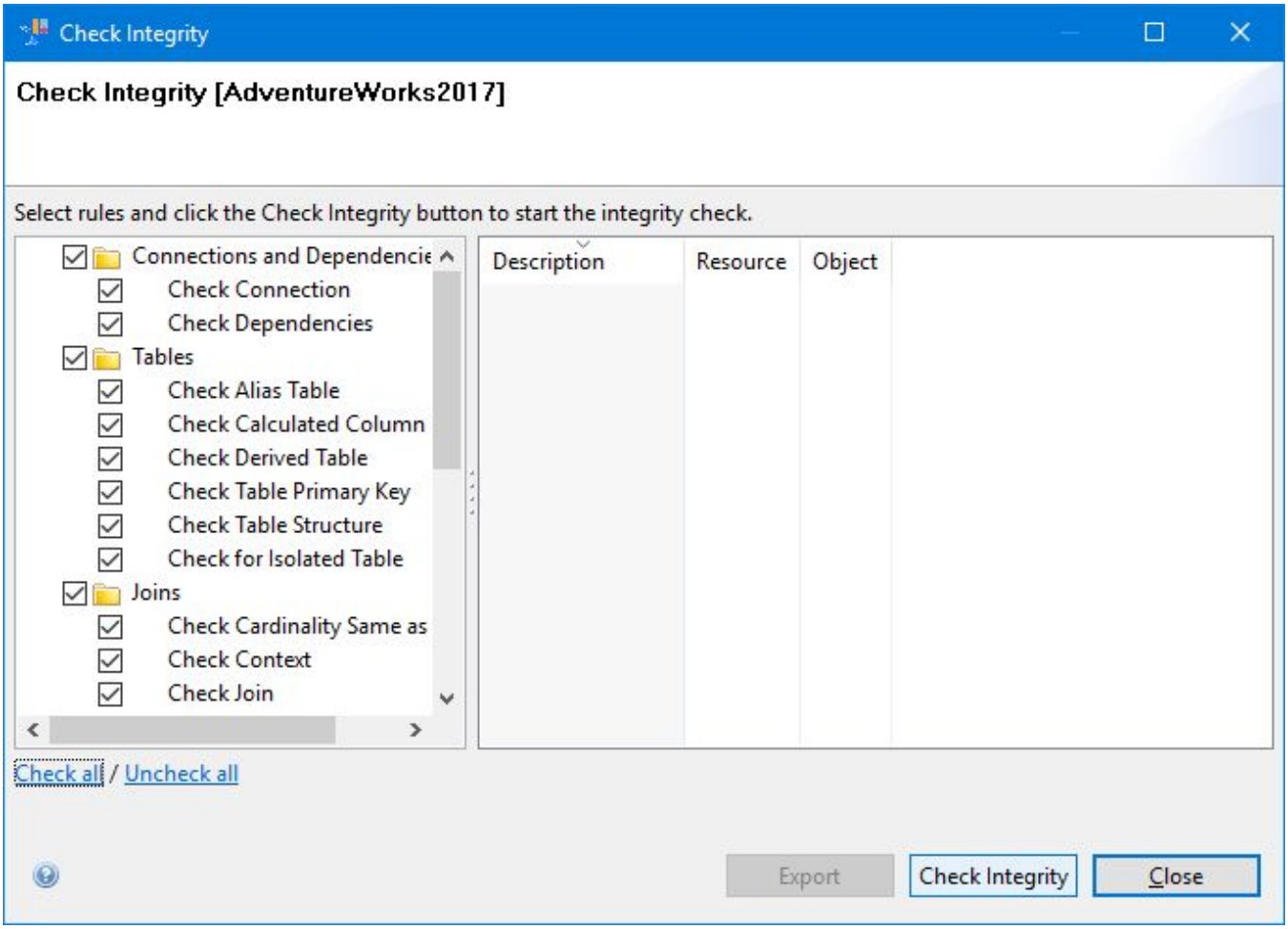


2. Check Integrity...



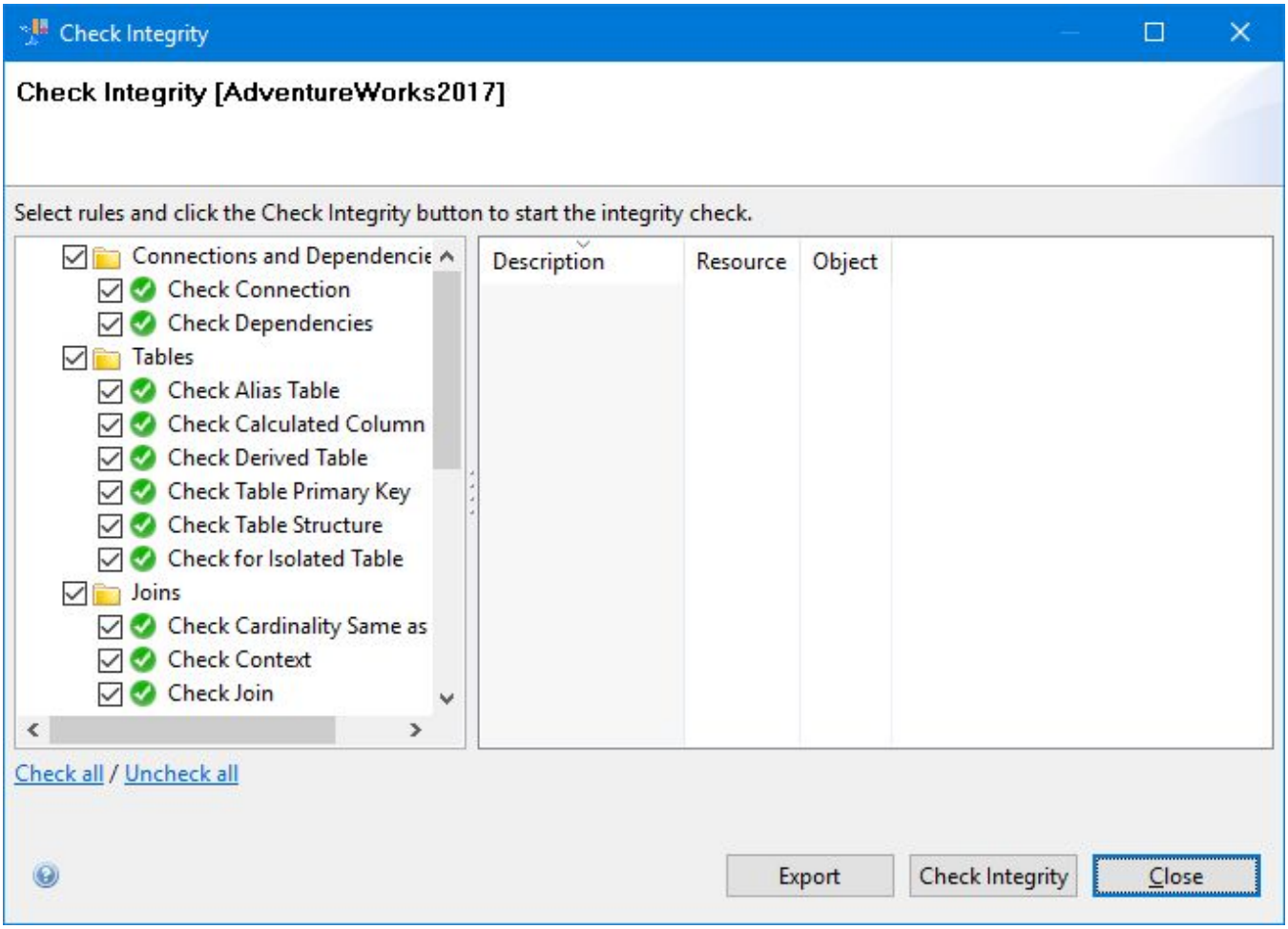
Under Business Layer > Right-Click Adventureworks2017
Click: Check Integrity...

3. Check Integrity



Click: Check all
Click: Check Integrity

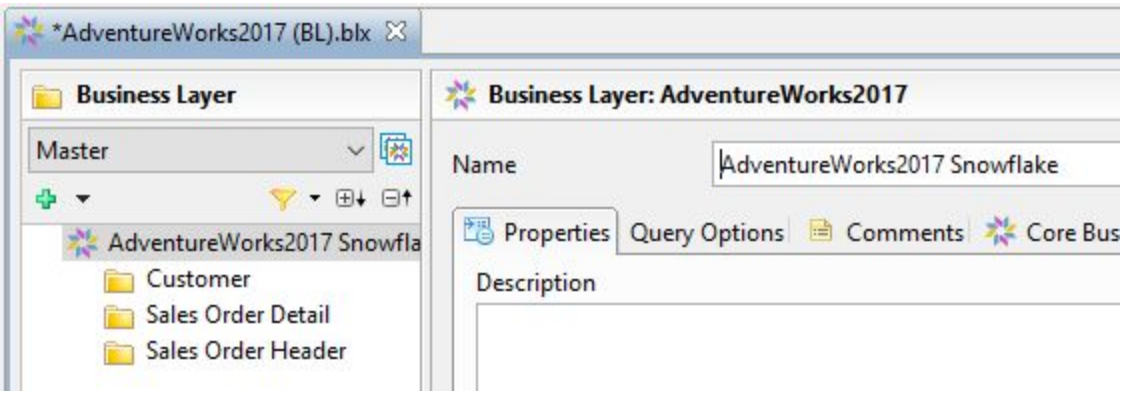
4. Confirm Results



Click: Close

Rename the Universe

1. Open AdventureWorks2017 (BL).blx

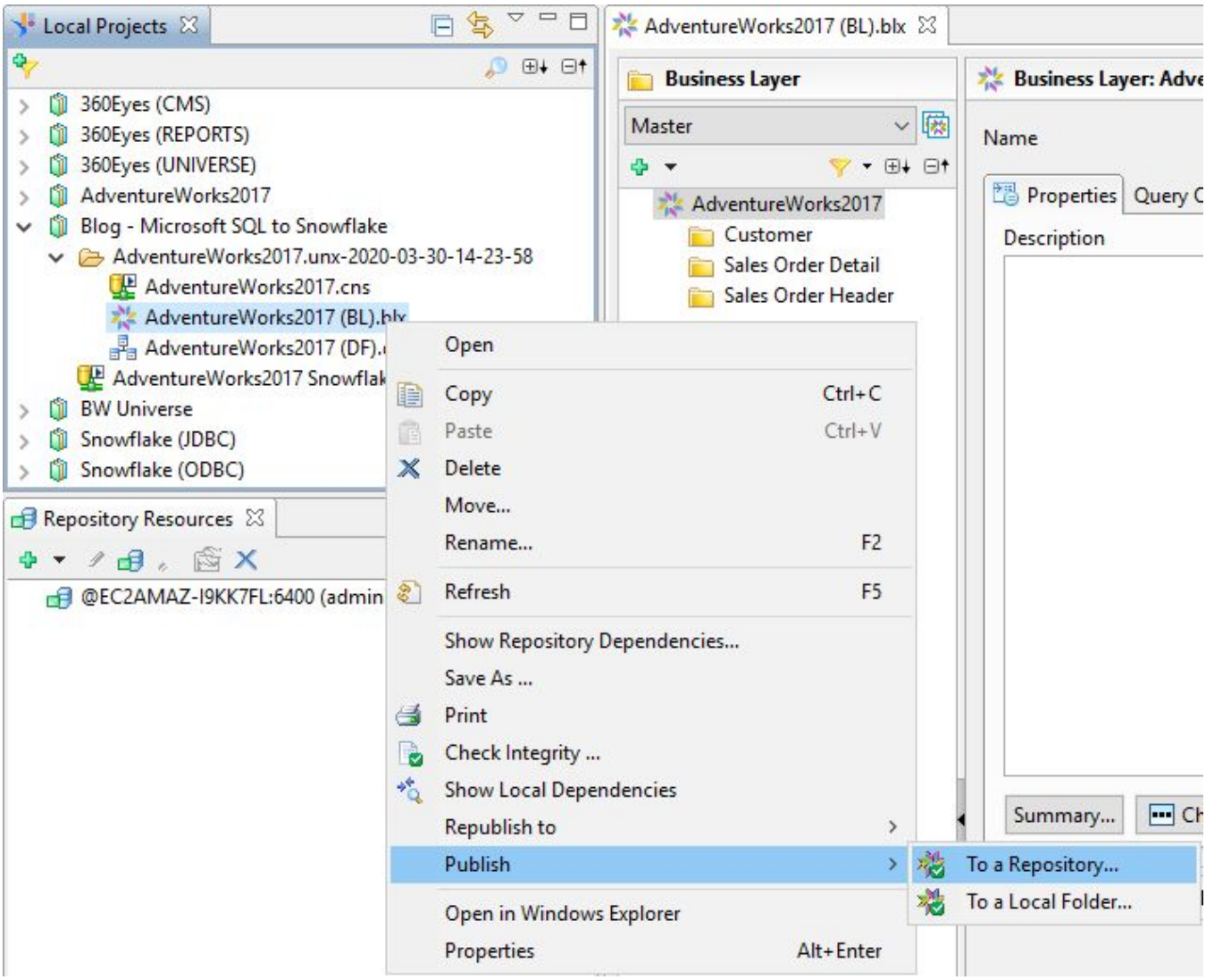


Name: [ENTER NEW NAME] E.g. AdventureWorks2017 Snowflake

2. Save the Business Layer

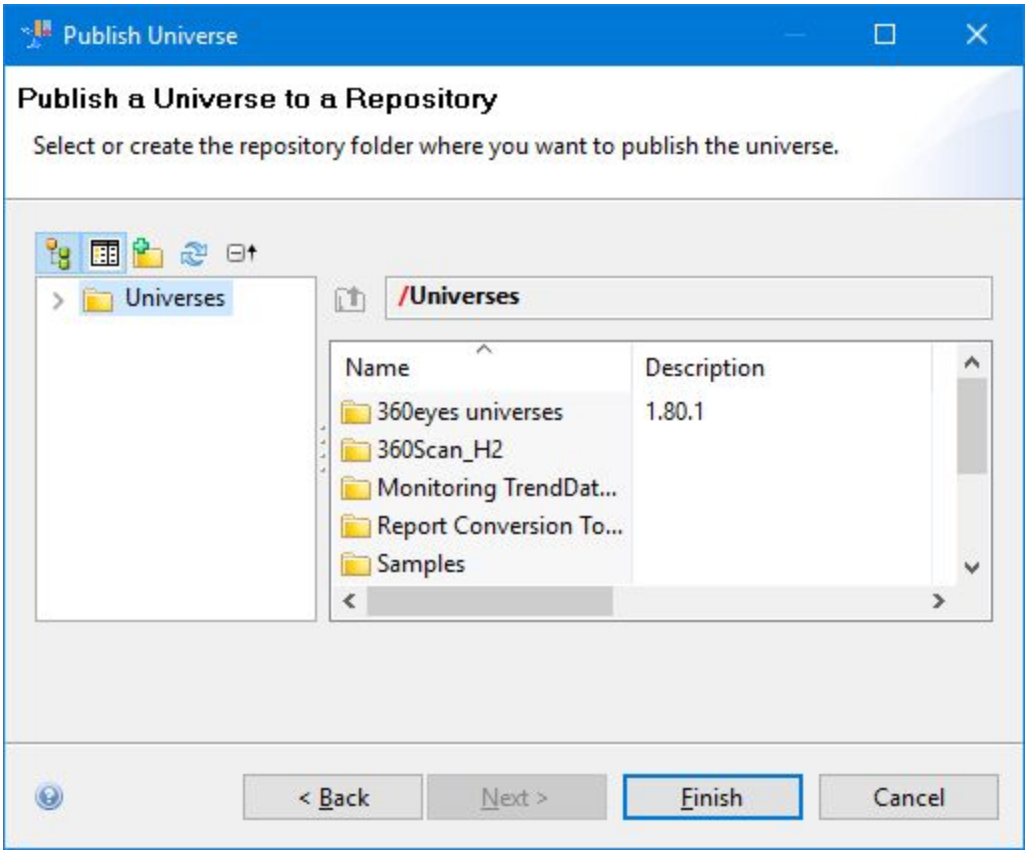
Publish the Business Layer with the Snowflake Data Foundation

1. Open AdventureWorks2017 (BL).blx

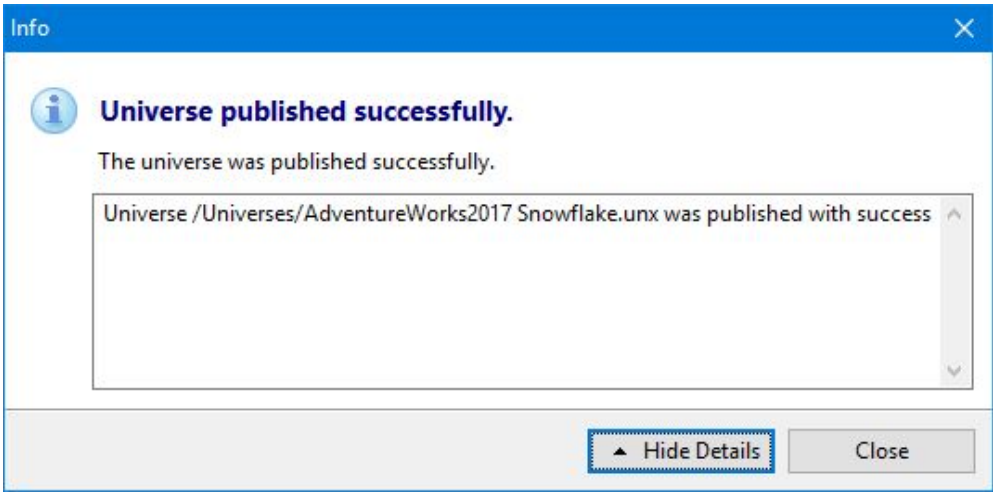


Under Local Projects > Right-Click Adventureworks2017 (BL).blx
Click: Publish > To a Repository
Click: Next

Select where you want to save the Universe



Click: Finish



Click: Close

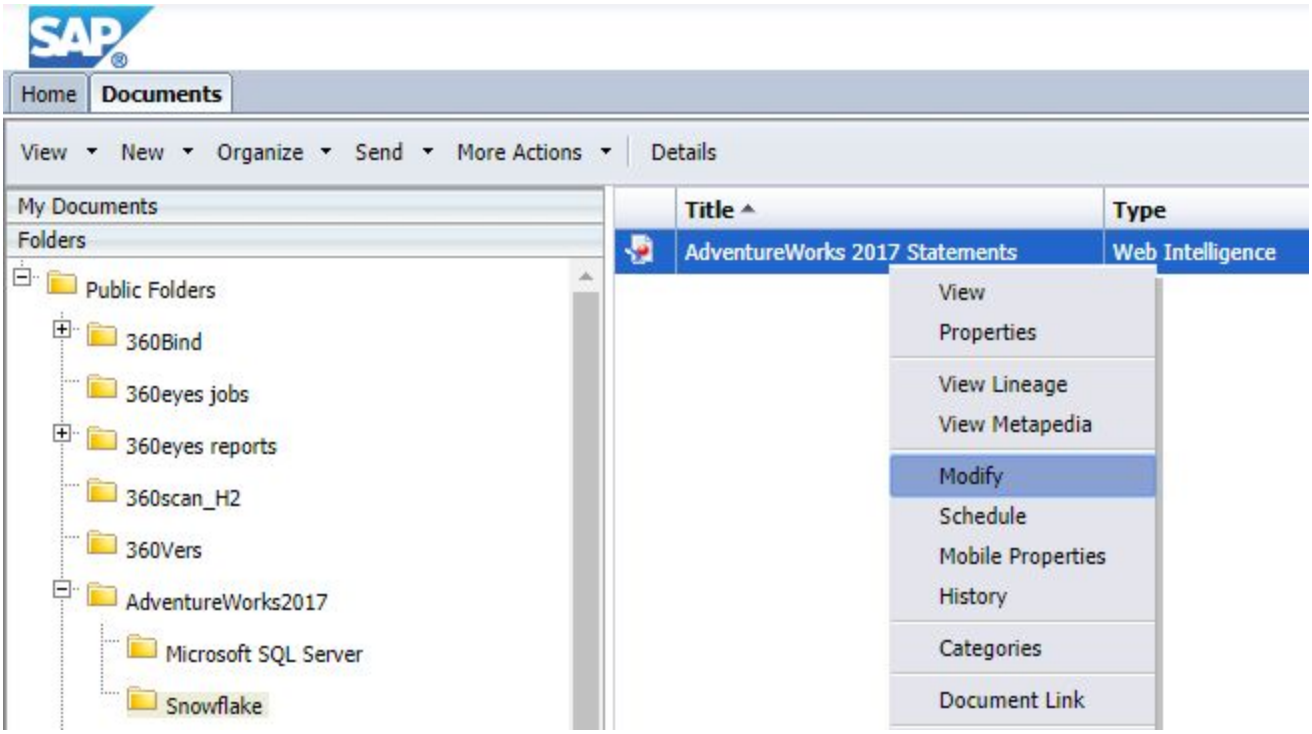
Updating Web Intelligence

These steps are to update your Web Intelligence documents to point to the new Snowflake Universe.

You can do this either within your current report or as we will do here, make a copy (backup) first and then modify the new one.

Note: These steps are to be repeated for every document.

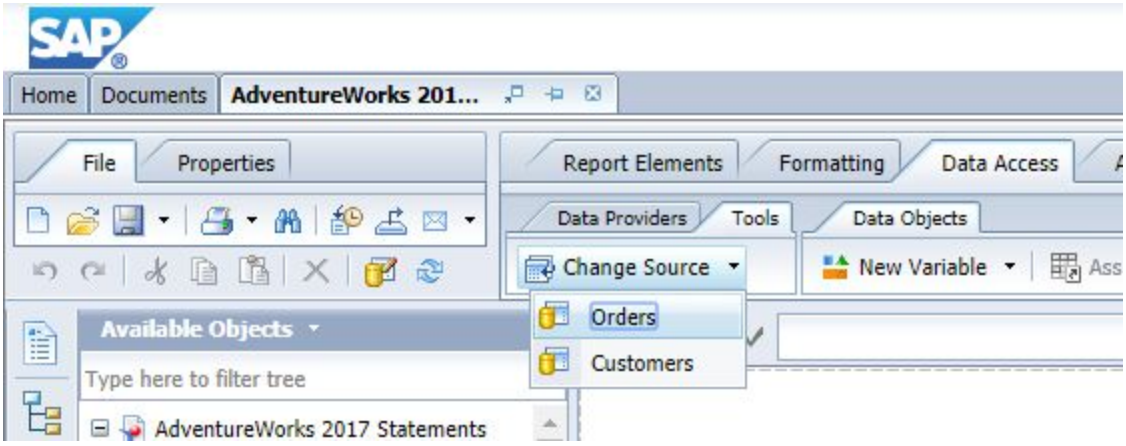
Modify your Web Intelligence



Right-Click the Web Intelligence
Click: Modify

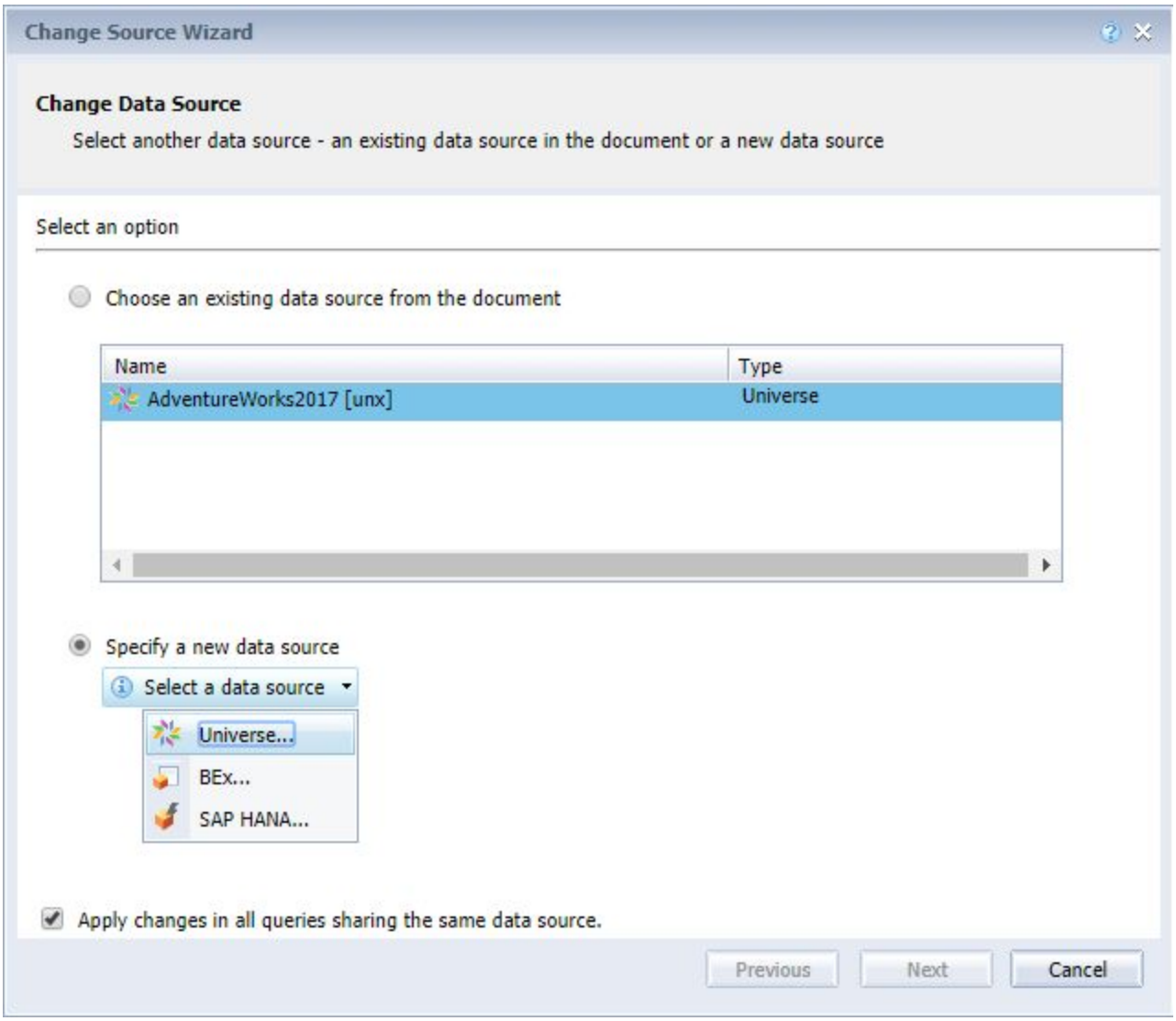
Change Data Source

1. Change Source



Click Data Access tab > Tools tab > Change Source
Click: Orders (this is the query to modify)

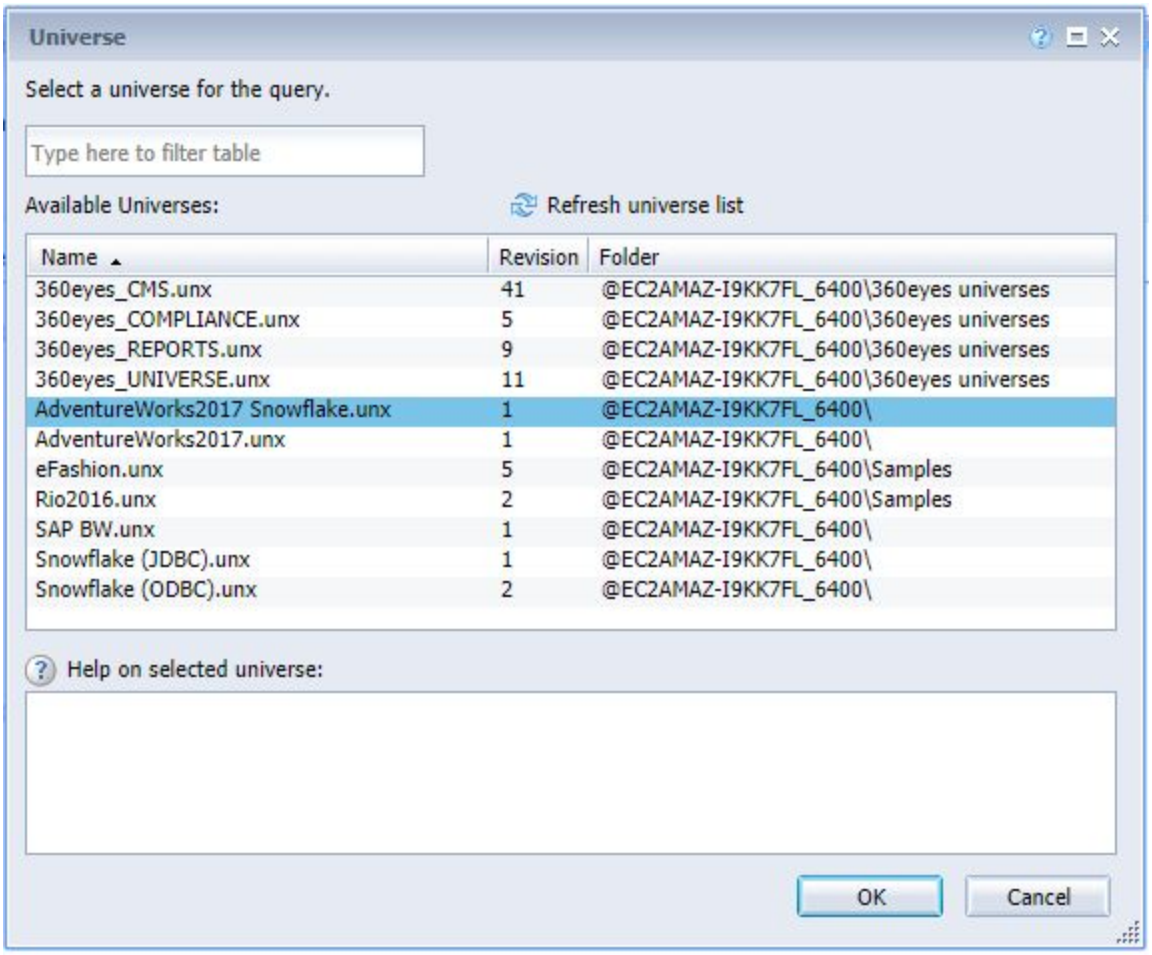
2. Change Source Wizard



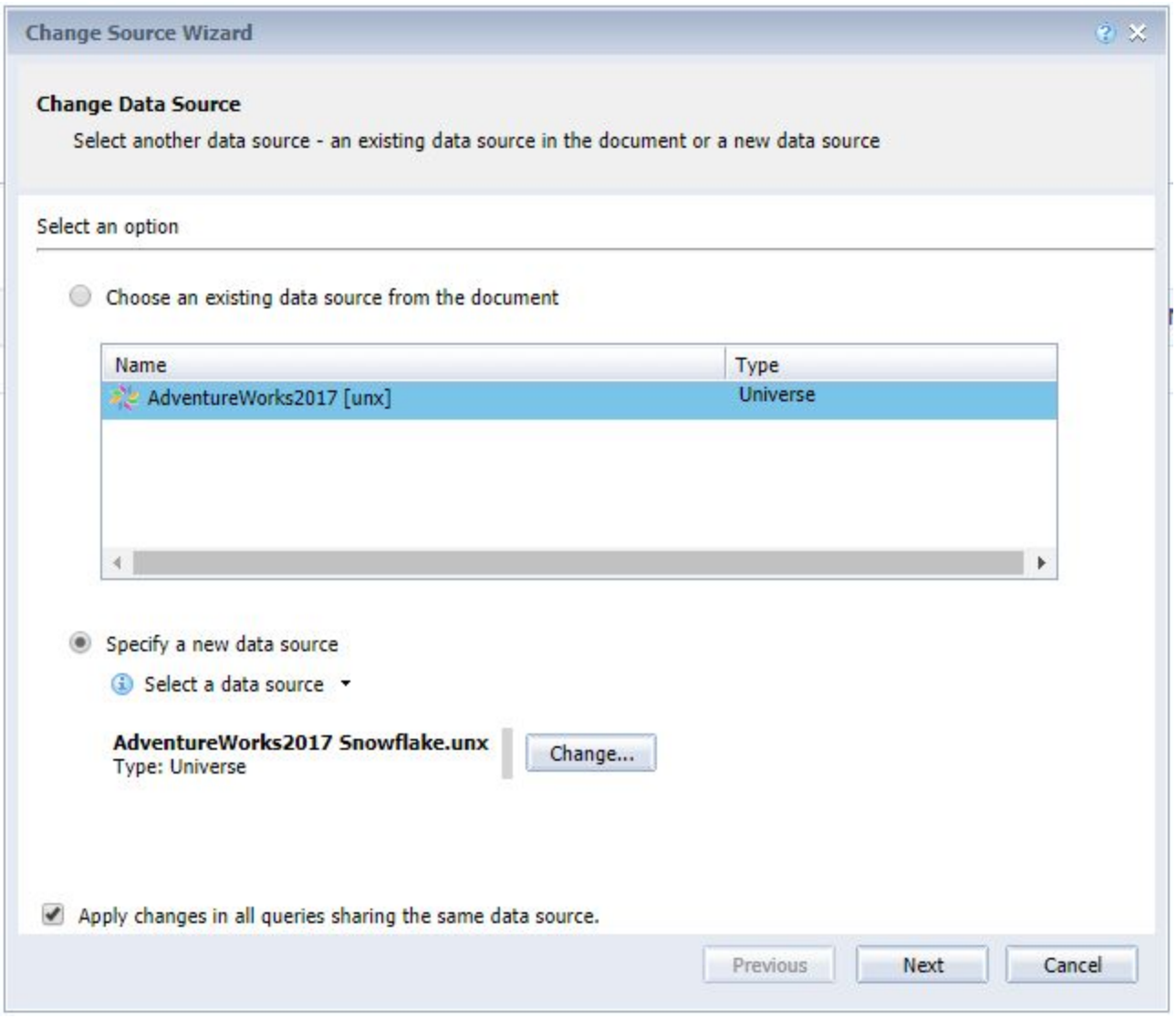
Click: Specify a new data source
Check: Apply changes in all queries sharing the same data source.
Click: Universe...

Note: You will need to repeat these steps for queries not sharing the same data source.

3. Select a universe for the query

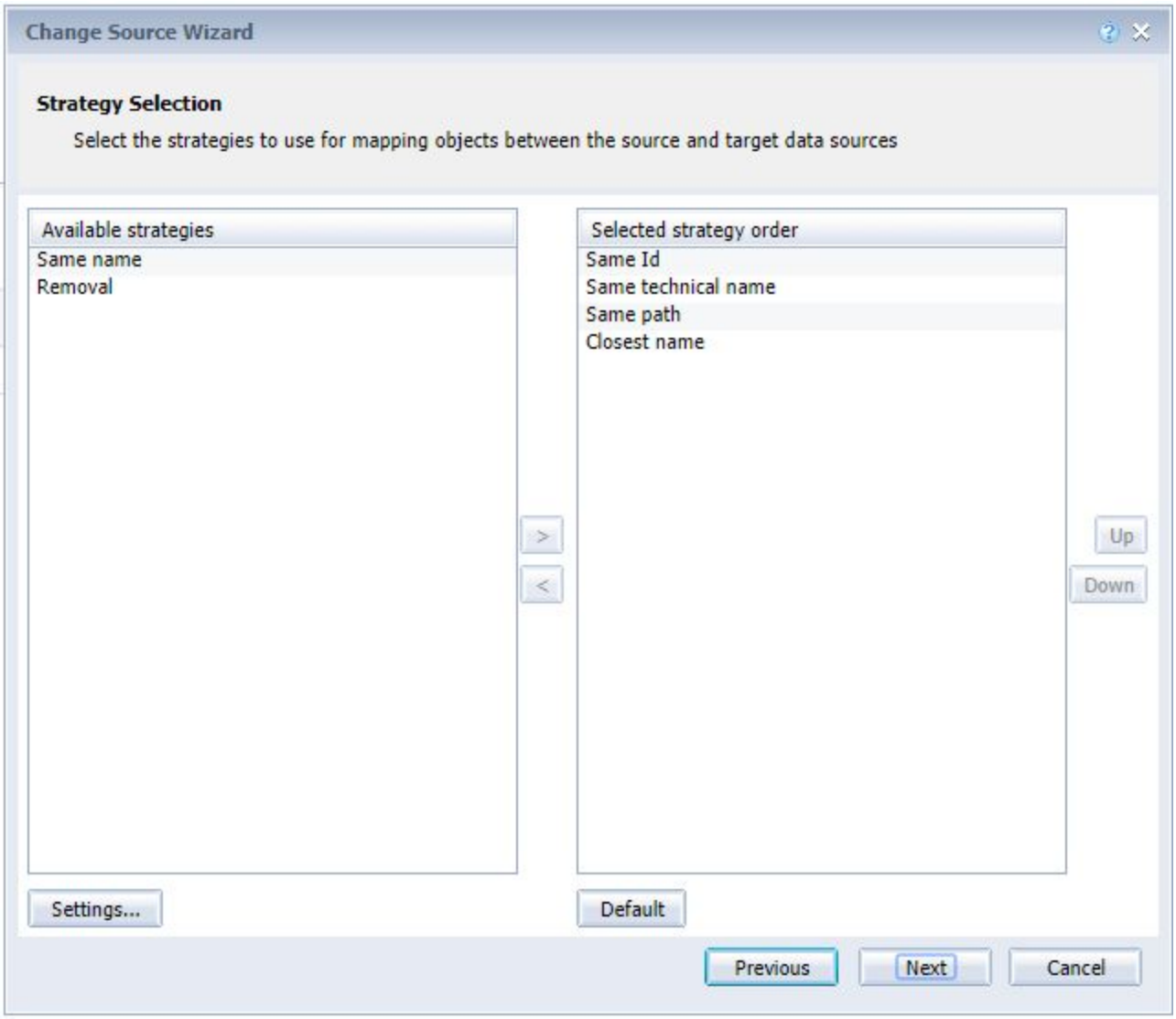


Click: AdventureWorks2017 Snowflake.unx
Click: OK


Change Source Wizard
Change Data Source
Select another data source - an existing data source in the document or a new data source
Select an option
Choose an existing data source from the document
Name
Type
AdventureWorks2017 [unx]
Universe
Specify a new data source
Select a data source
AdventureWorks2017 Snowflake.unx
Type: Universe
Change...
Apply changes in all queries sharing the same data source.
Previous
Next
Cancel

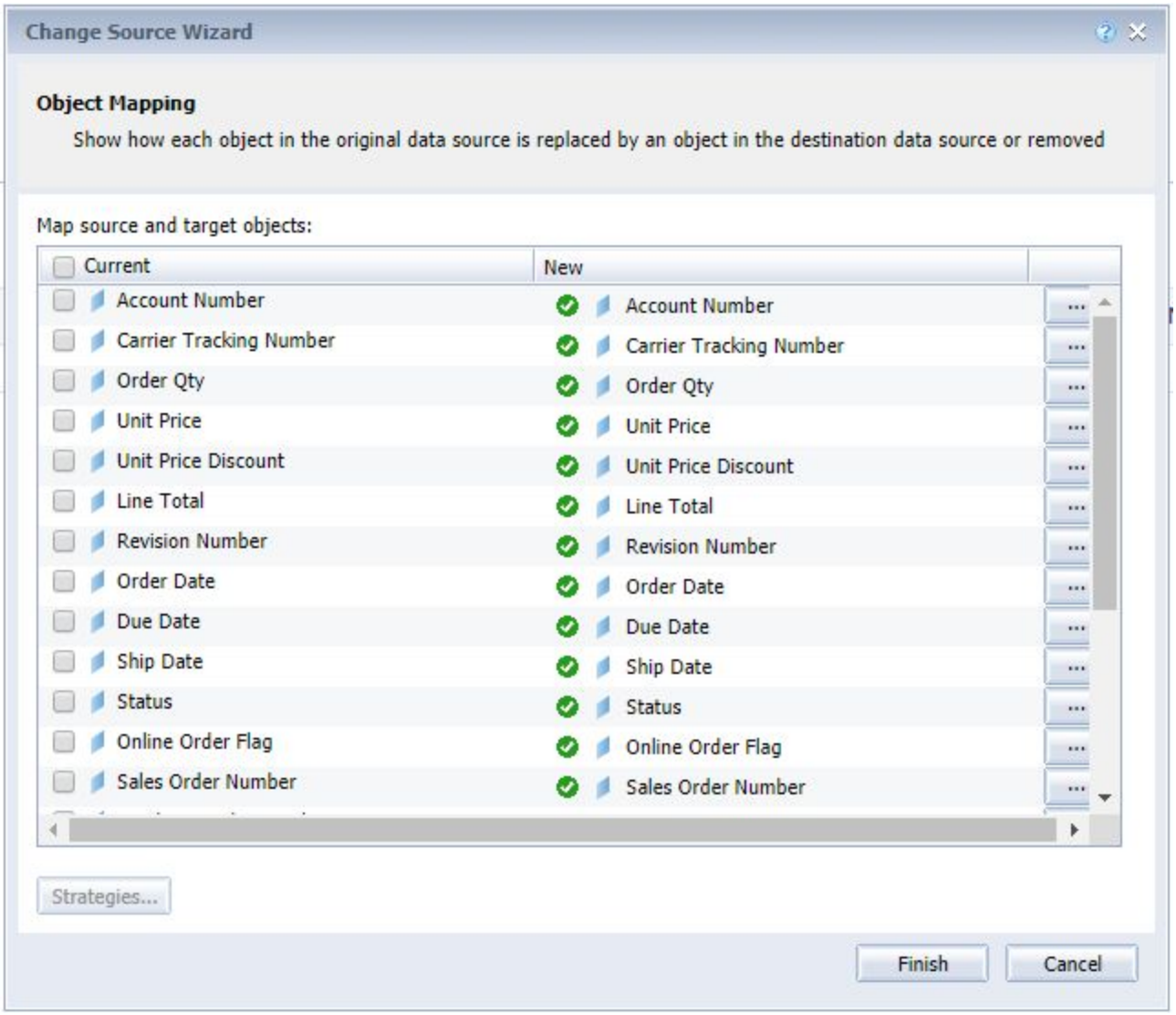
Click: Next

4. Strategy Selection



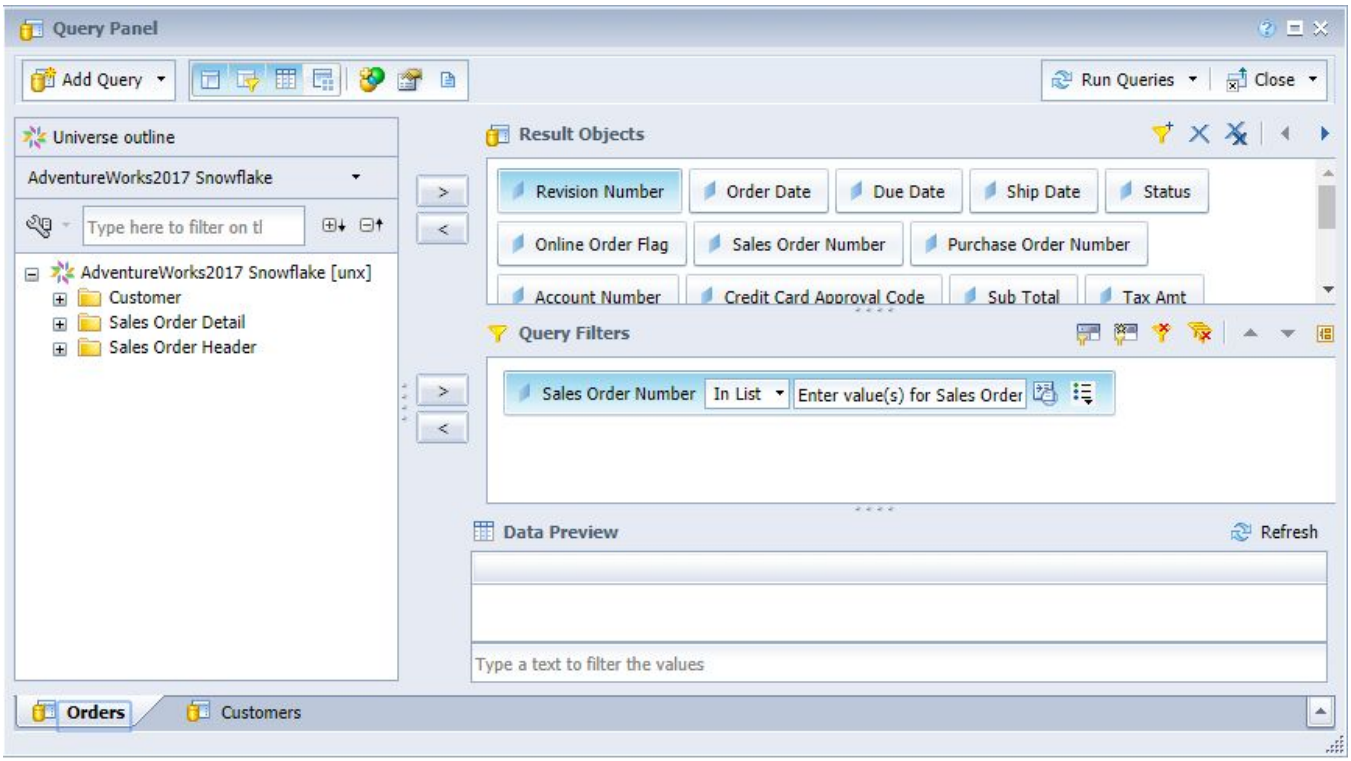
Click: Next

5. Object Mapping



Click: Finish

6. Query Panel



Click: Run Queries

7. Save the Web Intelligence document

Updating Crystal Reports

These steps are to update your Crystal Reports documents to update the database connection string(s) within the reports.

You can do this either within your current report or as we will do here, make a copy (backup) first and then modify the new one.

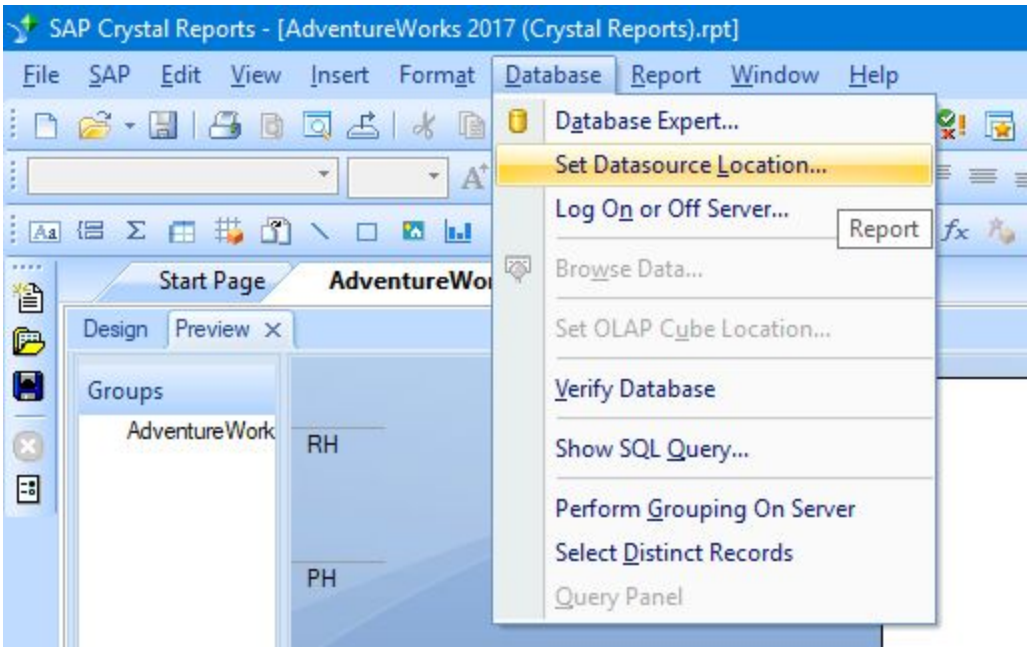
There are two methods:

1. In Crystal Reports “Desktop”
2. In the Central Management Console (CMC)

Note: These steps are to be repeated for every document.

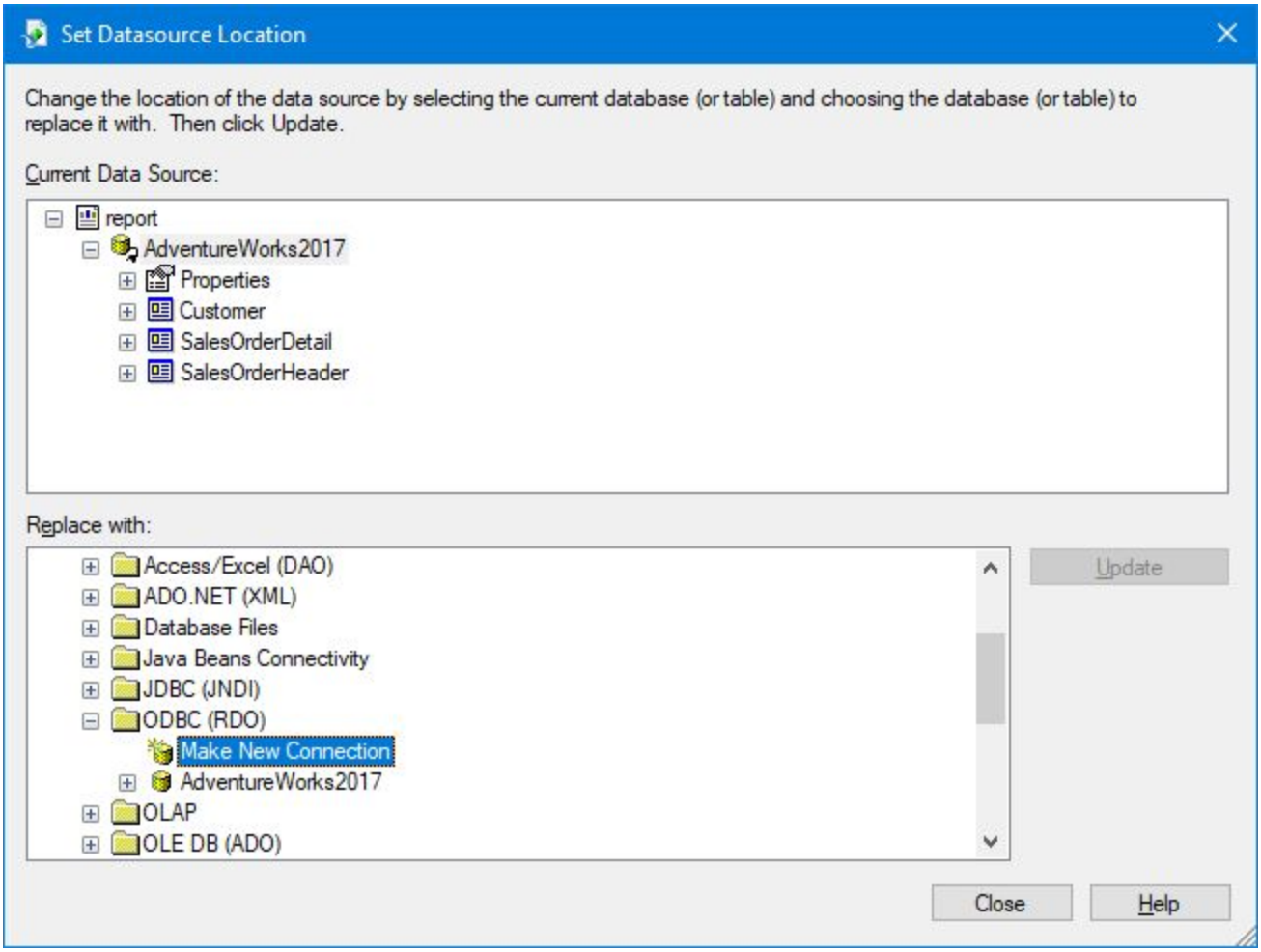
Crystal Reports “Desktop”

1. Open your Crystal Reports

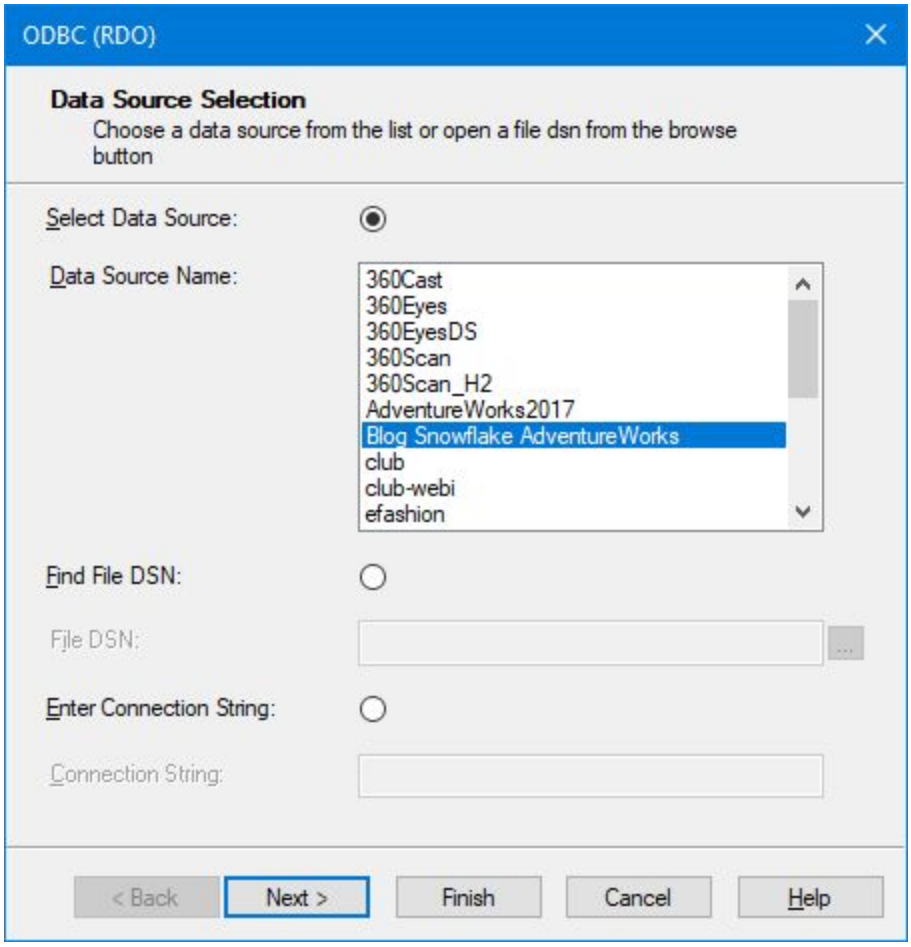


Under Database
Select: Set Datasource Location...

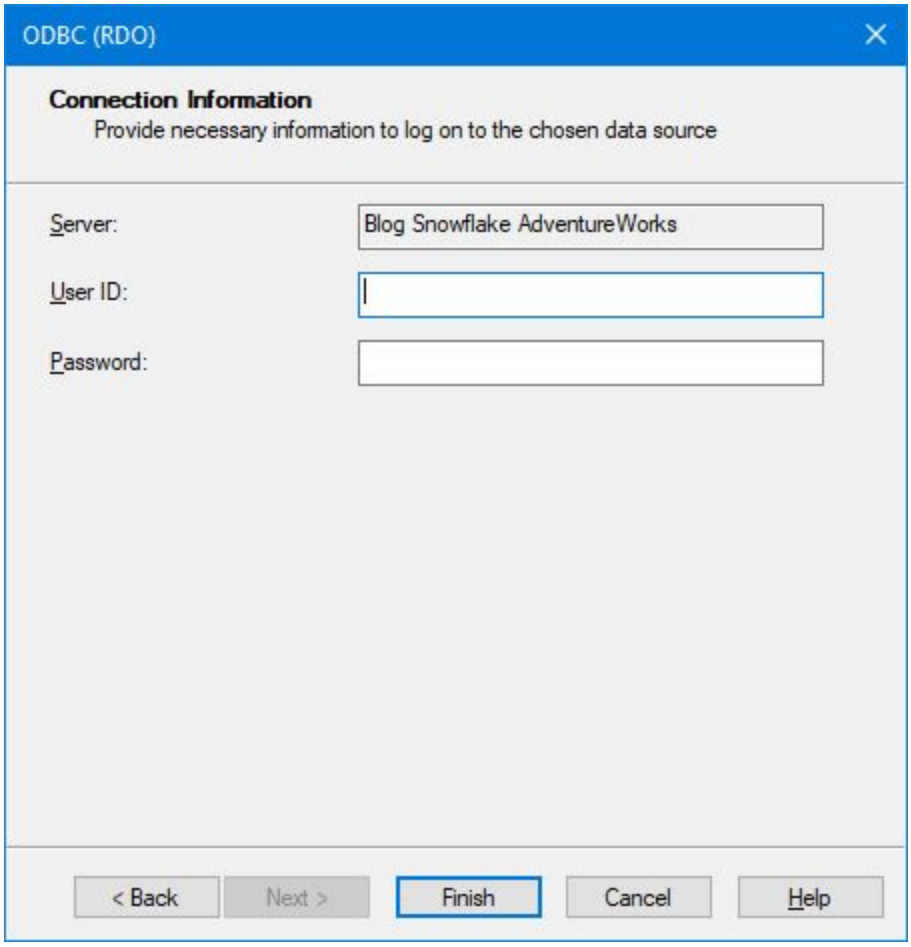
2. Set Datasource Location



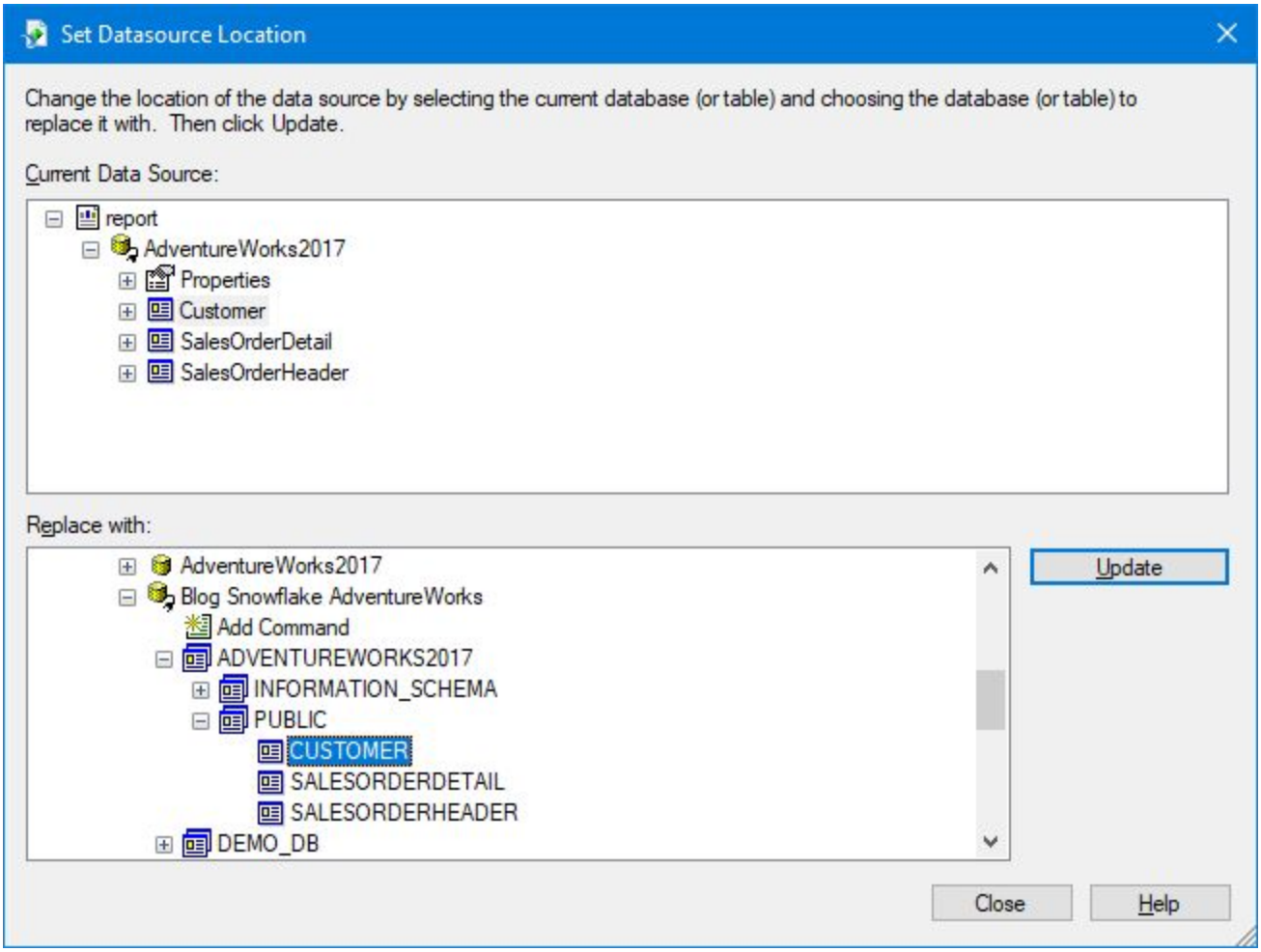
Under ODBC (RDO)
Select: Make a New Connection



Select: Blog Snowflake AdventureWorks
Click: Next



Enter: User ID
Enter: Password
Click: Finish



If your table names are identical you can simply map the database name.

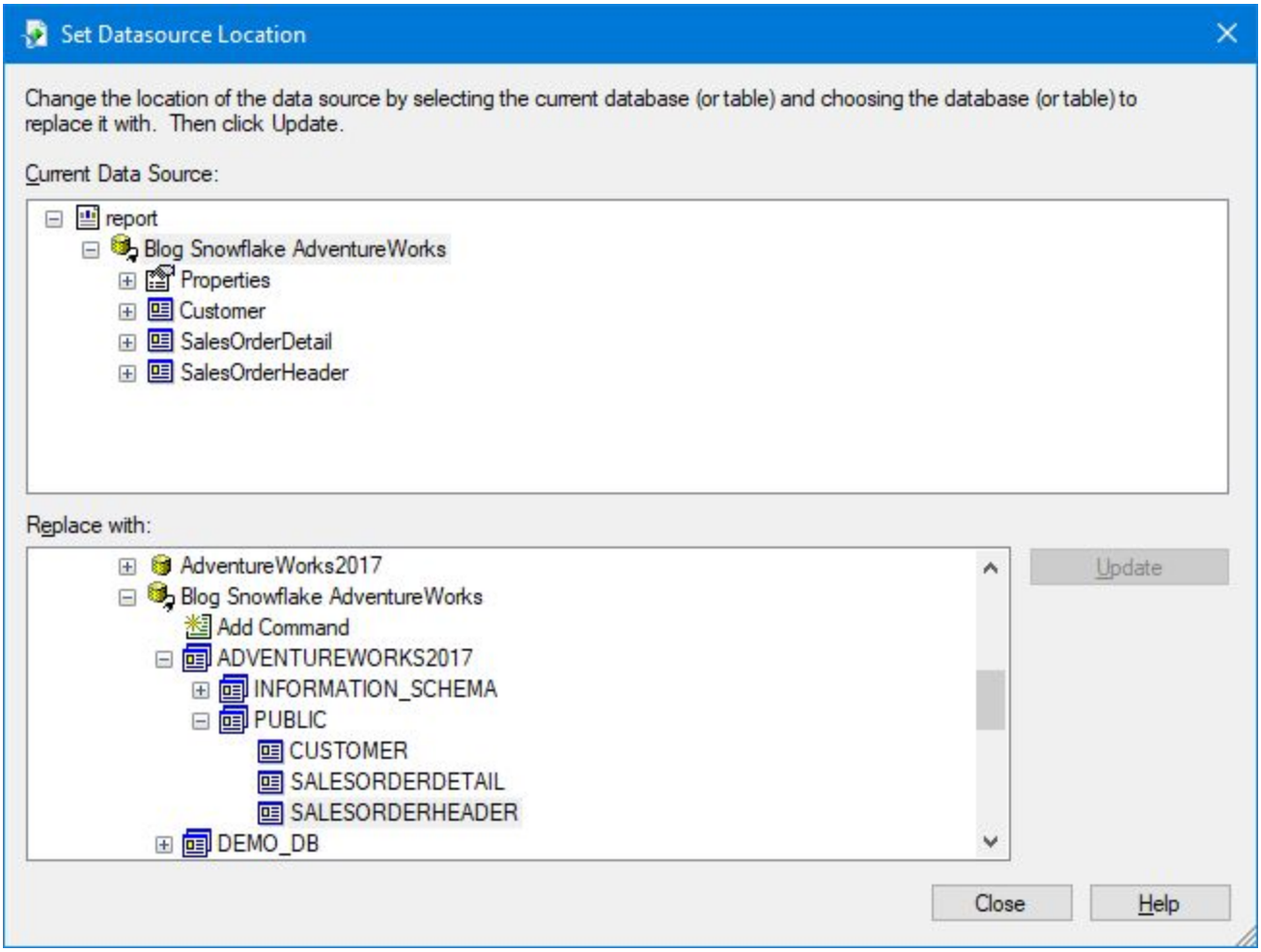
In this case, the tables are in uppercase so we need to map the tables individually:

Under Current Data Source
Select: Customer

Under Replace with
Select CUSTOMER

Click Update

Repeat for tables SalesOrderDetail and SalesOrderHeader.

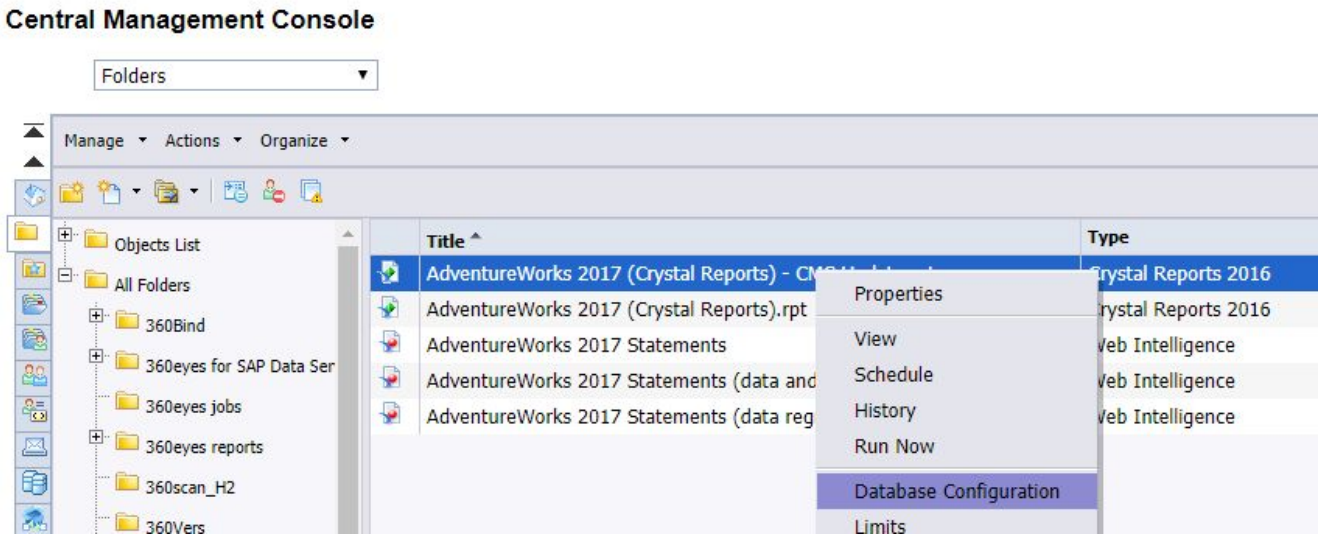


Tables are now remapped to Blog Snowflake Adventure Works
Click: Close

3. Save the Crystal Reports

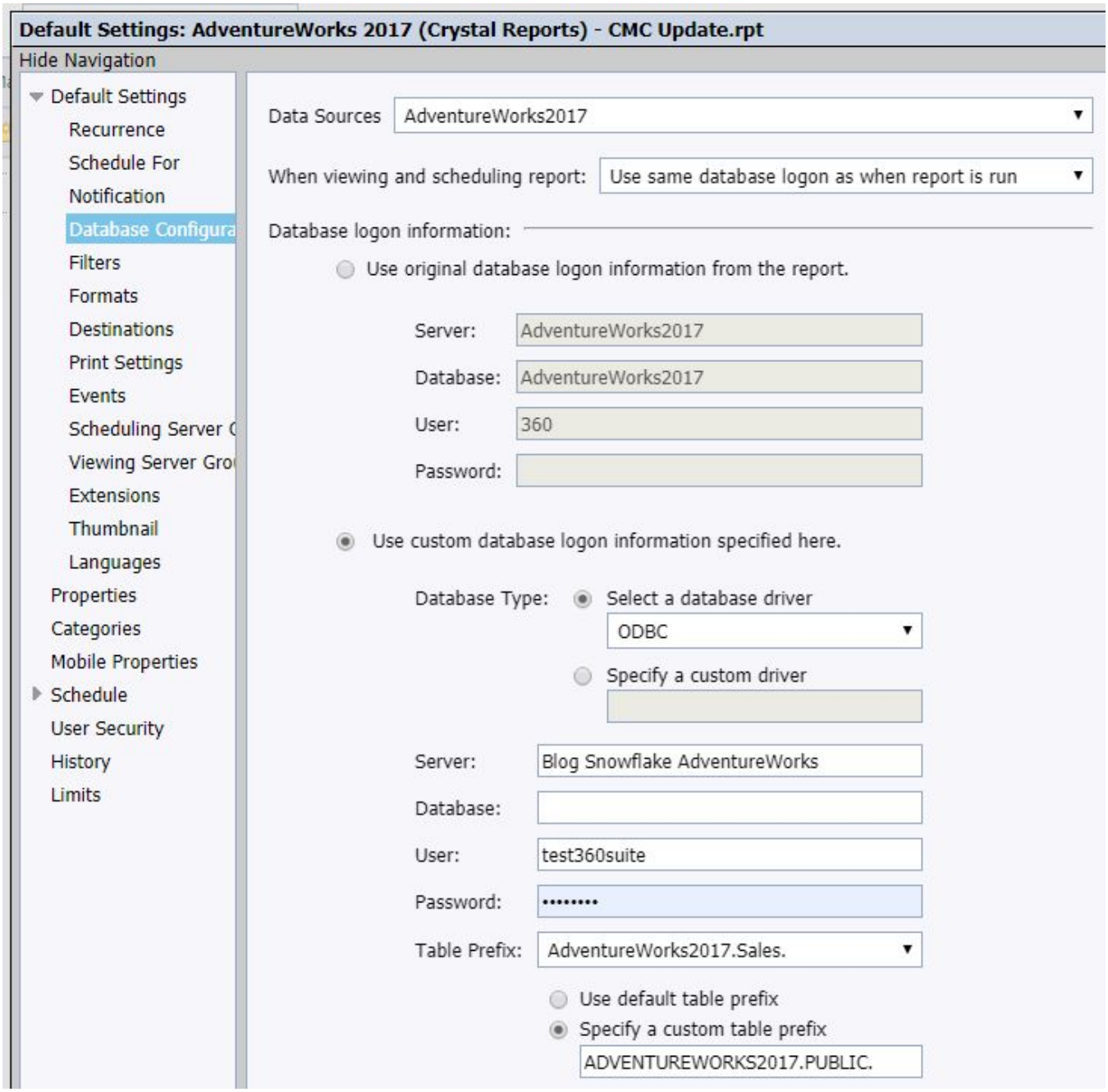
Central Management Console (CMC)

1. Logon to the Central Management Console (CMC)



Find your Crystal Reports
Right-Click > Database Configuration

3. Database Configuration



Select: Use custom database logon information specified here.
Enter: Server: Blog Snowflake AdventureWorks
Enter: Table Prefix > Specify a custom table prefix
ADVENTUREWORKS.PUBLIC.
Click: Save

Note: If the name (and case) of your tables are not exactly the same, you cannot update your Crystal Reports this way and you will need to use the method described previously, in Crystal Reports “Desktop”.

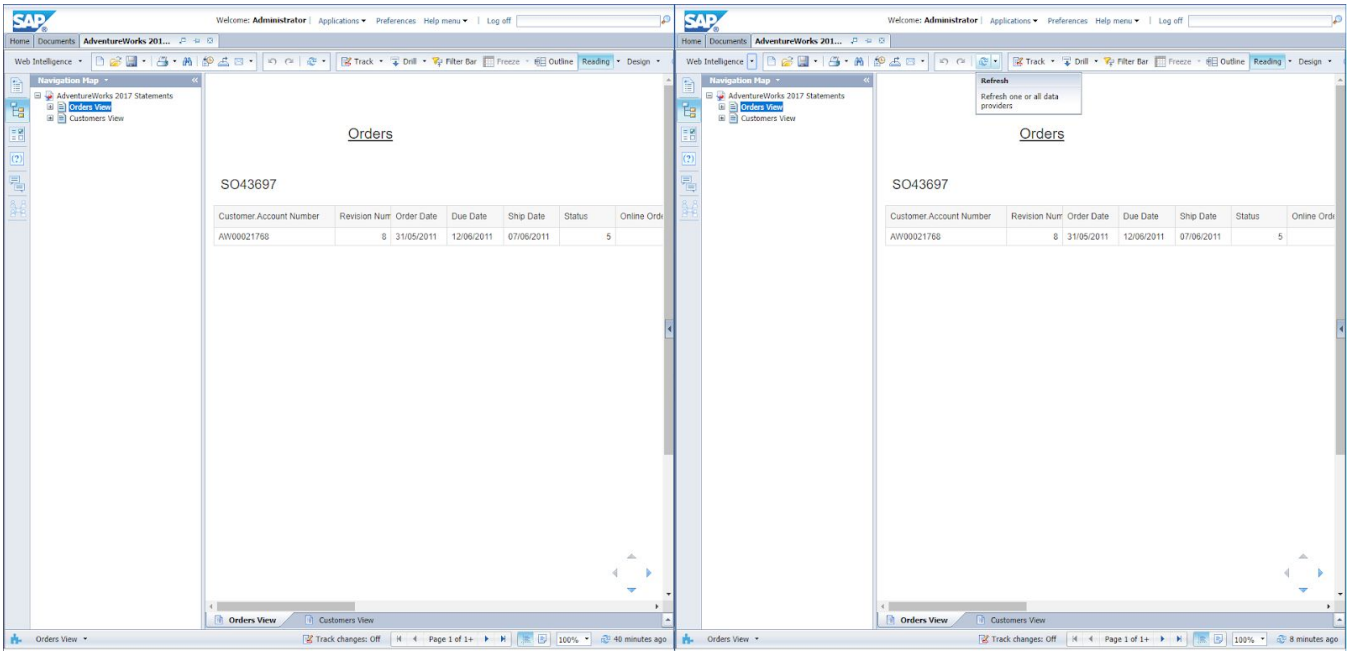
Testing Content

If you have made a copy of your Web Intelligence documents, you can do side by side comparisons.

In this section, we will validate that the documents appear to be the same and compare the refresh time between Microsoft SQL Server and Snowflake.

Comparing Data

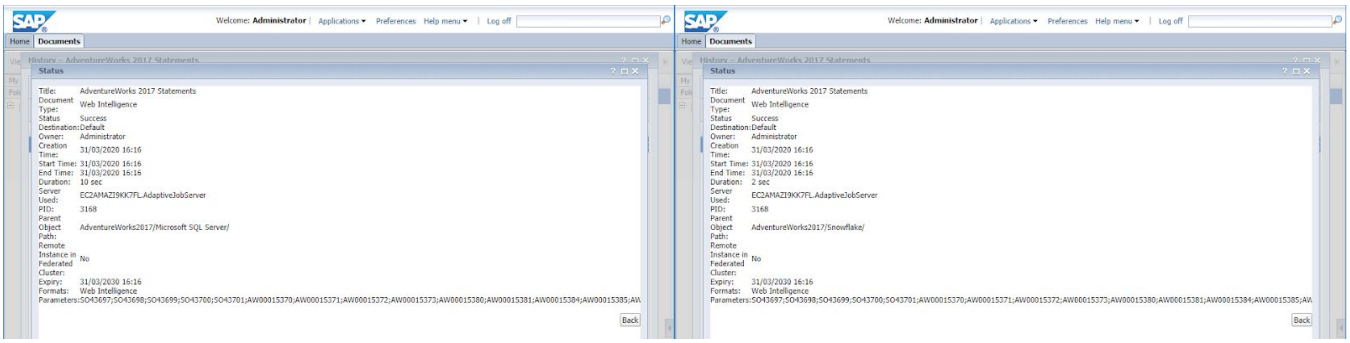
Unfortunately, using SAP BusinessObjects, this manual task involves opening both Web Intelligence documents and comparing values.



Note: As this task is manual (i.e. not automated), it will be time-consuming to execute therefore you will likely only test a subset of your reports. Mistakes are very likely as documents will have a lot of tables, columns, rows over many pages. It will be difficult to document evidence of testing and accuracy of these tests.

Comparing Performance

Unfortunately, once again using SAP BusinessObjects, this manual task is to schedule both Web Intelligence documents and compare running times.



Note: As this task is manual (i.e. not automated), it will be time-consuming to execute therefore you will likely only test a subset of your reports. It will be difficult to document.

With 360Suite Automation: Reducing time, cost and risks

Pre-Migration Assessment

Identify what will be impacted by repointing the database connectivity to Snowflake. This helps you prevent hidden side effects and helps you define the scope of the migration.

Universe Update

Required in most migration projects, you'll need to apply the necessary changes to the universes (tables, columns, measures, SELECT, WHERE).

Back-up

Make sure to have a reliable back-up before making any changes.

Document Update

Bulk repoint your Webi and Crystal reports to the new, updated universes.

Validation

Automate your testing and identify the regressions (layout, data, performance, connectivity) in order to fix your documents and universes. Avoid any risk and validate the migration.

Schedule Your Pre-Migration Assessment With Us

Request Your trial

Author:

Patrick Perrier is Analytics CoE Director and is passionate about helping companies maximize their SAP BusinessObjects investment with complex issues such as regulations, migration, and administration. He has over 20 years of experience in Business Intelligence, starting back when he worked at Crystal Decisions and then Business Objects. More recently, he held roles such as Head of Technical Architecture, BI, and Training

.