

# InfoSphere DataStage Configuration Steps

## **Data Migration**

**With Samples from Netezza to Azure Synapse Analytics**

*Prepared by*

**Data SQL Ninja Engineering Team ([datasqlninja@microsoft.com](mailto:datasqlninja@microsoft.com))**

## Disclaimer

The High-Level Architecture, Migration Dispositions and guidelines in this document is developed in consultation and collaboration with Microsoft Corporation technical architects. Because Microsoft must respond to changing market conditions, this document should not be interpreted as an invitation to contract or a commitment on the part of Microsoft.

Microsoft has provided generic high-level guidance in this document with the understanding that MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THE INFORMATION CONTAINED HEREIN.

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet Web site references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2020 Microsoft. All rights reserved.

**Note:** The detail provided in this document has been harvested as part of a customer engagement sponsored through the [Data SQL Ninja Engineering](#).

# Table of Contents

1	Introduction .....	5
2	InfoSphere DataStage .....	6
2.1	Utilizing InfoSphere DataStage .....	6
2.2	InfoSphere DataStage Initial Screen .....	7
2.3	Select Job from the Project .....	7
2.4	Working with the InfoSphere DataStage Palette .....	8
2.5	Sample Netezza Job – Update and Insert .....	8
3	Connectivity with InfoSphere DataStage .....	10
3.1	Selection for Connectivity of Available Source and Targets .....	10
3.1.1	Overview of Connectivity.....	10
3.1.2	Configuration .....	10
3.1.3	Settings for the DRS Connector stage.....	10
3.1.4	Defining a DRS Connector stage connection to the database .....	10
3.2	Singleton Bulk Inserts.....	11
3.3	DRS Connection.....	12
3.4	Selection of Choices for Processing.....	12
4	Ingestion, Processing and Insertion via InfoSphere DataStage .....	14
4.1	Selection of File Options .....	14
4.2	Setup Database Tables for Use.....	14
4.3	Import Table Selections.....	15
4.4	Import each table used.....	15
4.5	InfoSphere DataStage ODBC.ini driver parameters.....	16
4.6	Converted Sample Job Update & Insert .....	17
4.7	Utilize same method of pulling information .....	19
4.8	For updating data, first, ingest into staging table using Bulk Insert. ....	19
4.9	Detail example of the Update Statement.....	20
4.10	Detail example of the Insert statement.....	21

4.11	Changes to Transformers.....	22
4.12	Change Transformer Processor .....	23
4.13	Executing Stored Proc on Azure Synapse Analytics .....	23
4.14	Stored Procedure to Build Key and Insert.....	23
4.15	InfoSphere DataStage Job to Execute a Stored Procedure .....	24
4.16	Stored Procedure General Setup .....	24
4.17	Set Up Syntax for Stored Procedure Execution.....	25
4.18	Set up "Copy" process Input Properties .....	26
4.19	Set up "Copy" Process Output Properties for Stored Proc.....	28
4.20	Set up "Copy" process for Output Mapping .....	29
4.21	Set up "Copy" process Output Columns .....	30
5	Large Dataset Ingestion .....	31
5.1	Large Data Ingestion in Azure Synapse Strategy .....	31
5.2	Writing Data into Azure Storage with InfoSphere DataStage.....	31
5.3	Bulk Loading into Azure Synapse with InfoSphere DataStage .....	32
5.4	Bulk Loading with COPY INTO statement .....	32
6	Samples Scripts.....	34
6.1	Ingestion into Azure Synapse Analytics SQL Pool .....	34

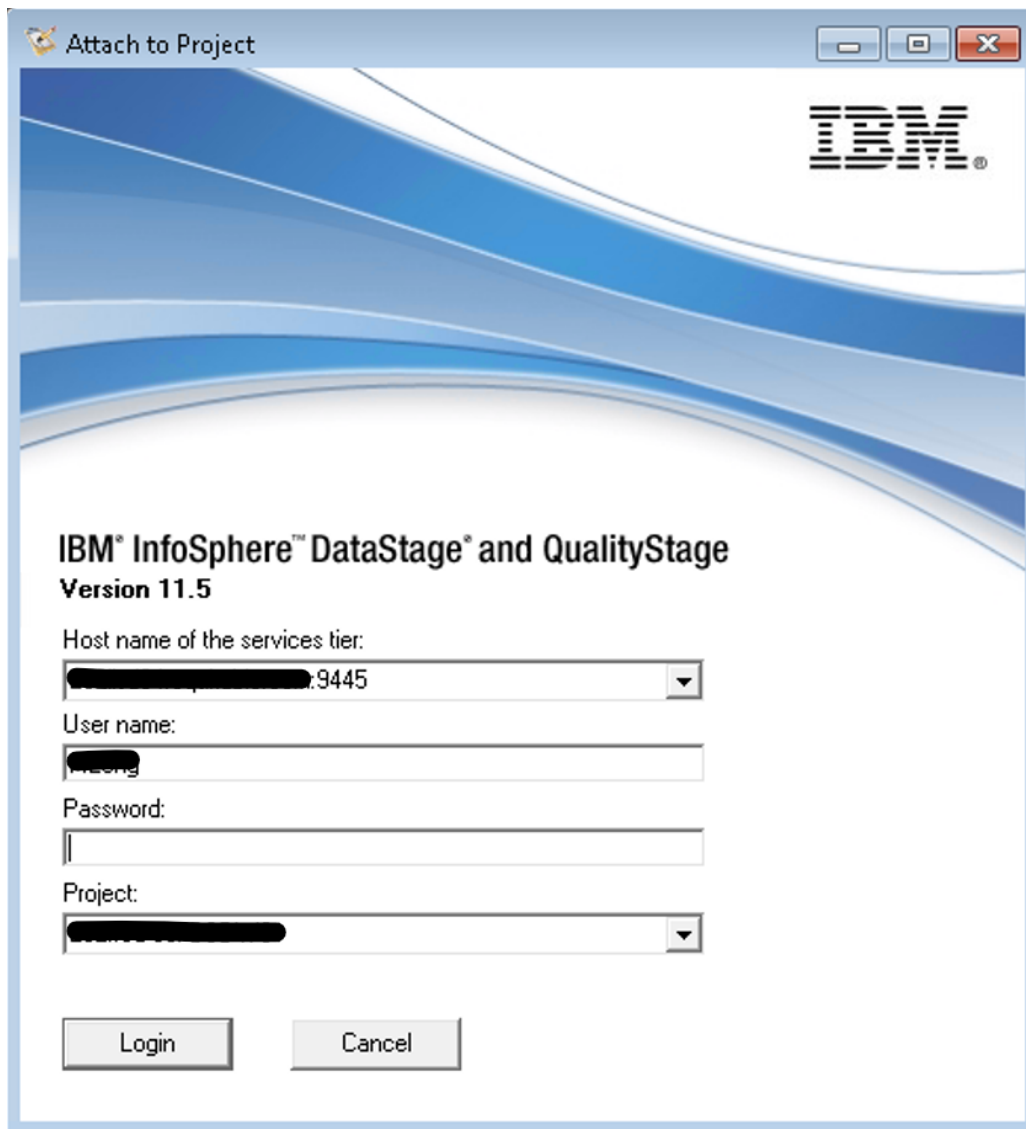
# 1 Introduction

This document has been created for the purpose of providing guidance and recommendations for InfoSphere DataStage implementations, to help align configuration to Azure Synapse Analytics Data Warehouse and maximize data throughput leveraging available cloud options.

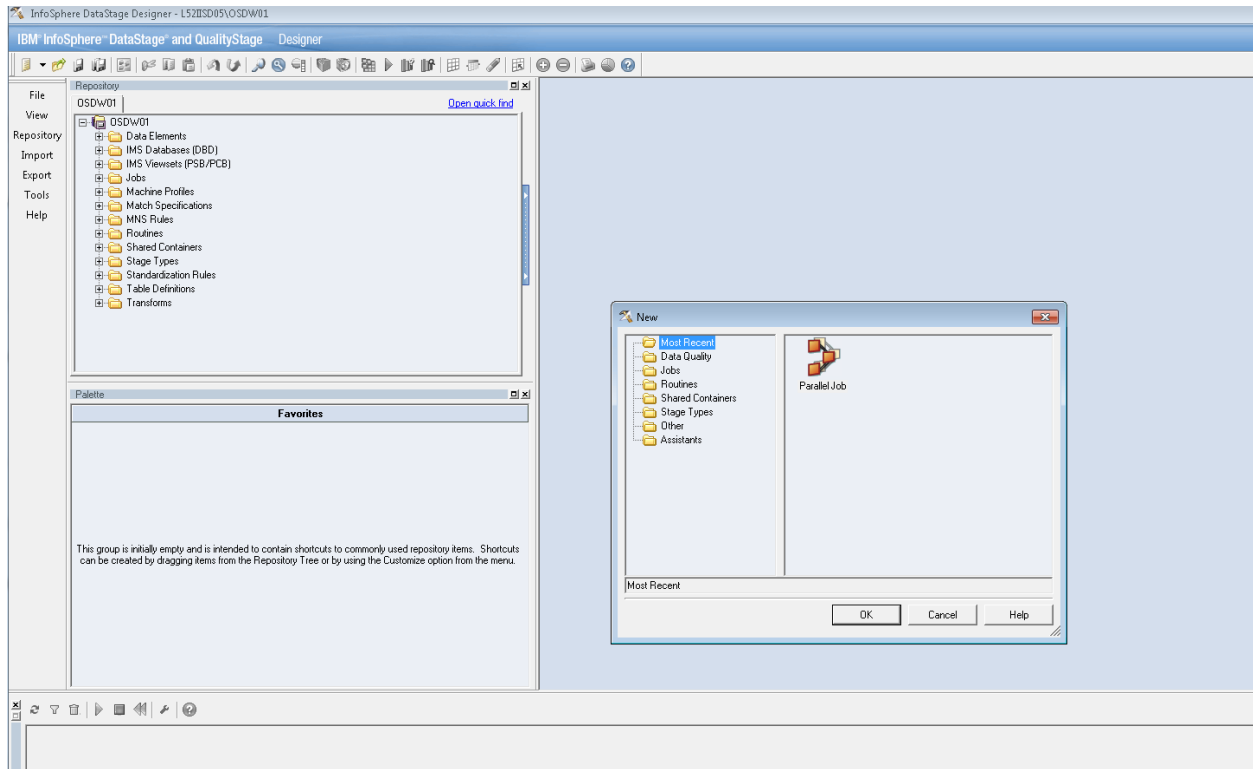
## 2 InfoSphere DataStage

### 2.1 Utilizing InfoSphere DataStage

- A client InfoSphere DataStage application is installed on workstation/laptop (Designer)
- Clicking on the Designer icon opens the InfoSphere DataStage application which requires authentication to the server with name, password, and project name
- Selection of a job is made from a tree structure on the left showing the jobs and sequences in hierarchical fashion
- InfoSphere DataStage is initiated from an icon on the desktop designated as "Designer"

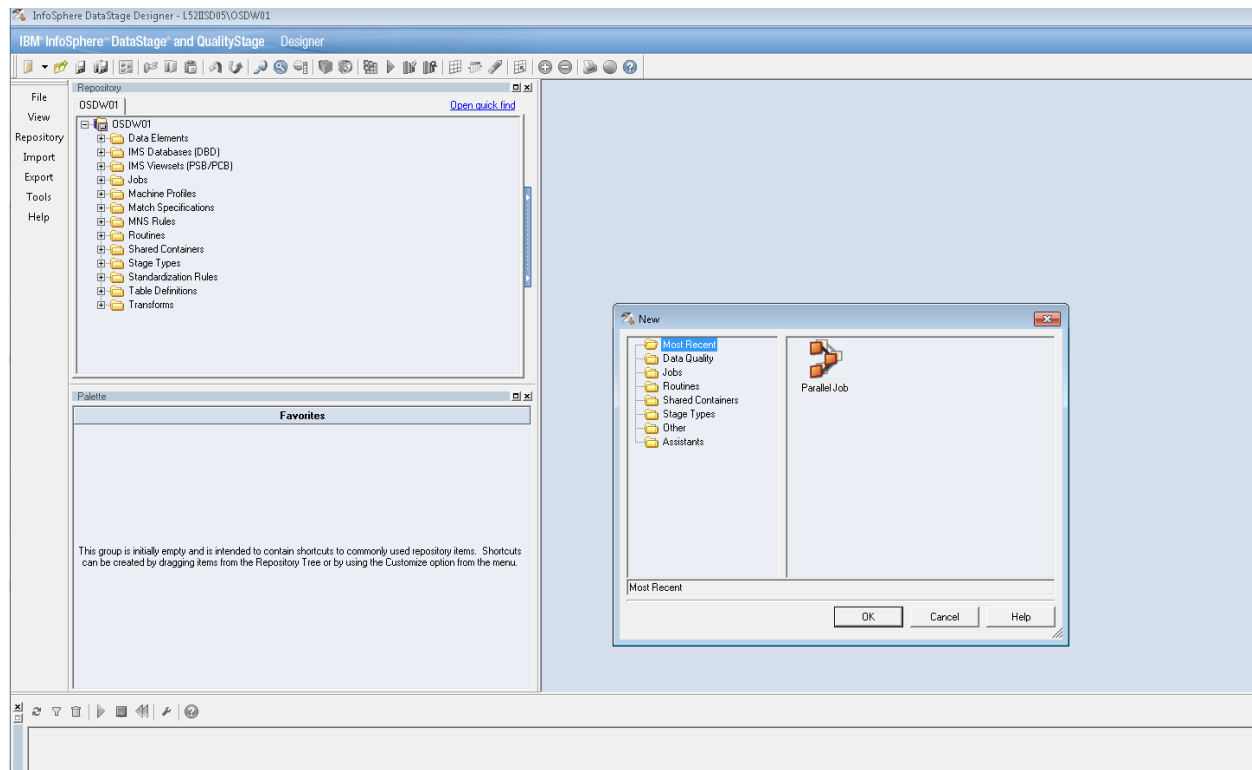


## 2.2 InfoSphere DataStage Initial Screen



## 2.3 Select Job from the Project

- Click in the tree list for your final selection.
- The jobs that are designated in green are "sequence" jobs. These are job agents to select one or multiple jobs to run within the selection.



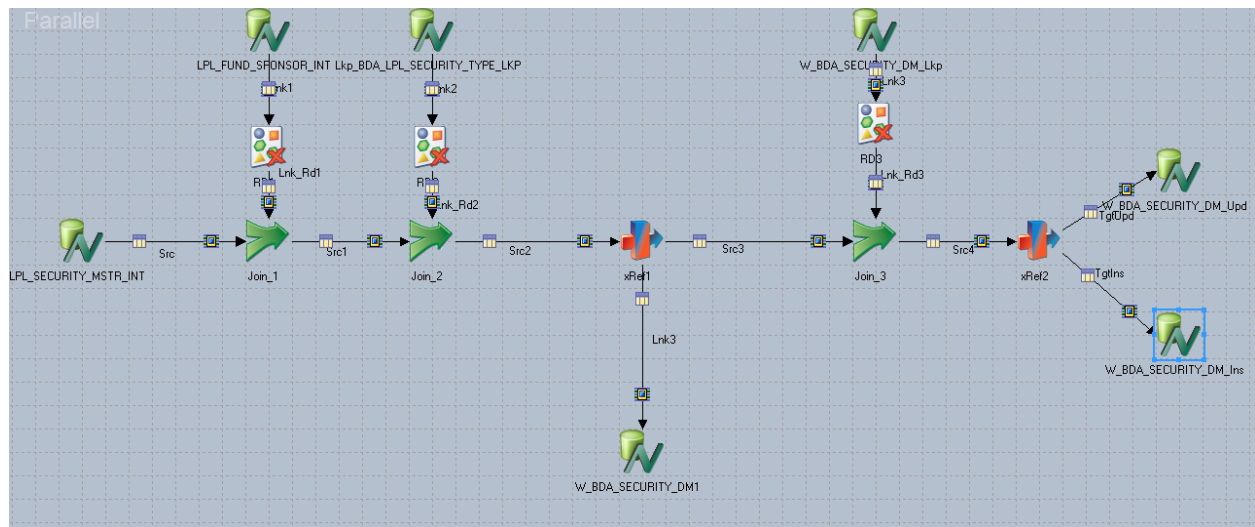
## 2.4 Working with the InfoSphere DataStage Palette

- On the left of the screen there are selections for the palette.
- The Database category allows the user to select the connectivity type for this job.
- The Processing category allows the user to select the type of processing to occur, i.e. transformer
- The File category allows the user to select a type of target or source file (as opposed to a database) and requires no connectivity information

## 2.5 Sample Netezza Job – Update and Insert

This sample job both updates and inserts rows to a table that has a key in Netezza





## 3 Connectivity with InfoSphere DataStage

### 3.1 Selection for Connectivity of Available Source and Targets

- Selection of ODBC Connector will only allow singleton transactions
- Selection of DRS Connector will allow for selection of Bulk Insert option
- All stored procedures run on Azure Synapse Analytics must be executed via the Stored Procedure Database option from this selection screen

#### 3.1.1 Overview of Connectivity

Use the DRS Connector stage to access relational database management systems by using the native interfaces that are available for the corresponding databases. The choice of the database type is not determined when the stage is placed on the job canvas but is instead specified when the stage is configured.

#### 3.1.2 Configuration

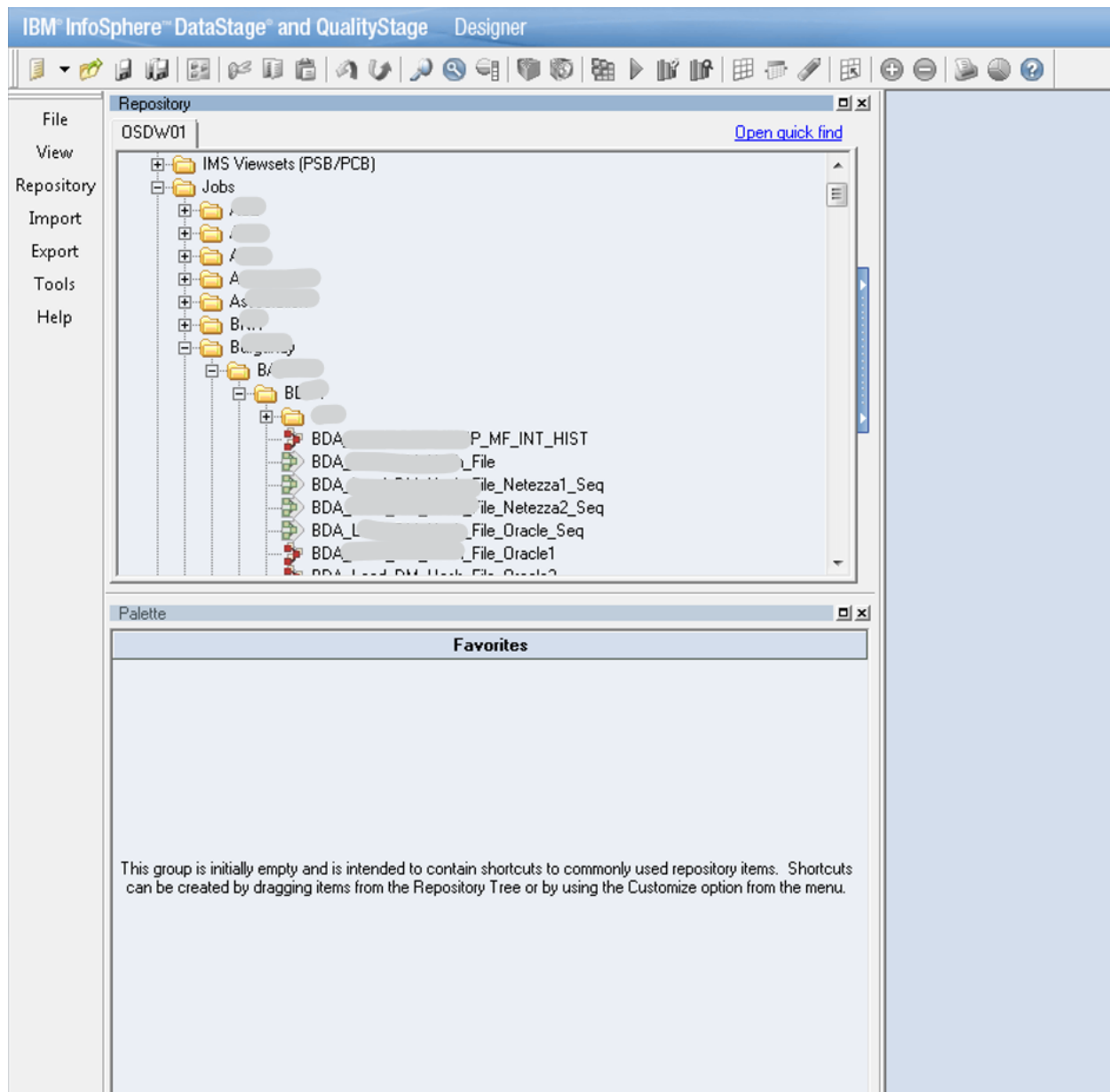
The DRS Connector stage supports IBM® DB2®, Oracle, and ODBC data sources. For other database types, you can configure the DRS Connector stage to use the ODBC database type and access the databases through the ODBC drivers that are included with InfoSphere Information Server.

#### 3.1.3 Settings for the DRS Connector stage

After the DRS Connector stage is placed on the job canvas, it needs to be configured to perform the operation intended by the job design.

#### 3.1.4 Defining a DRS Connector stage connection to the database

When a new DRS Connector stage is added to the server or parallel canvas, you must specify the connection information for the database that the stage connects to at run time.



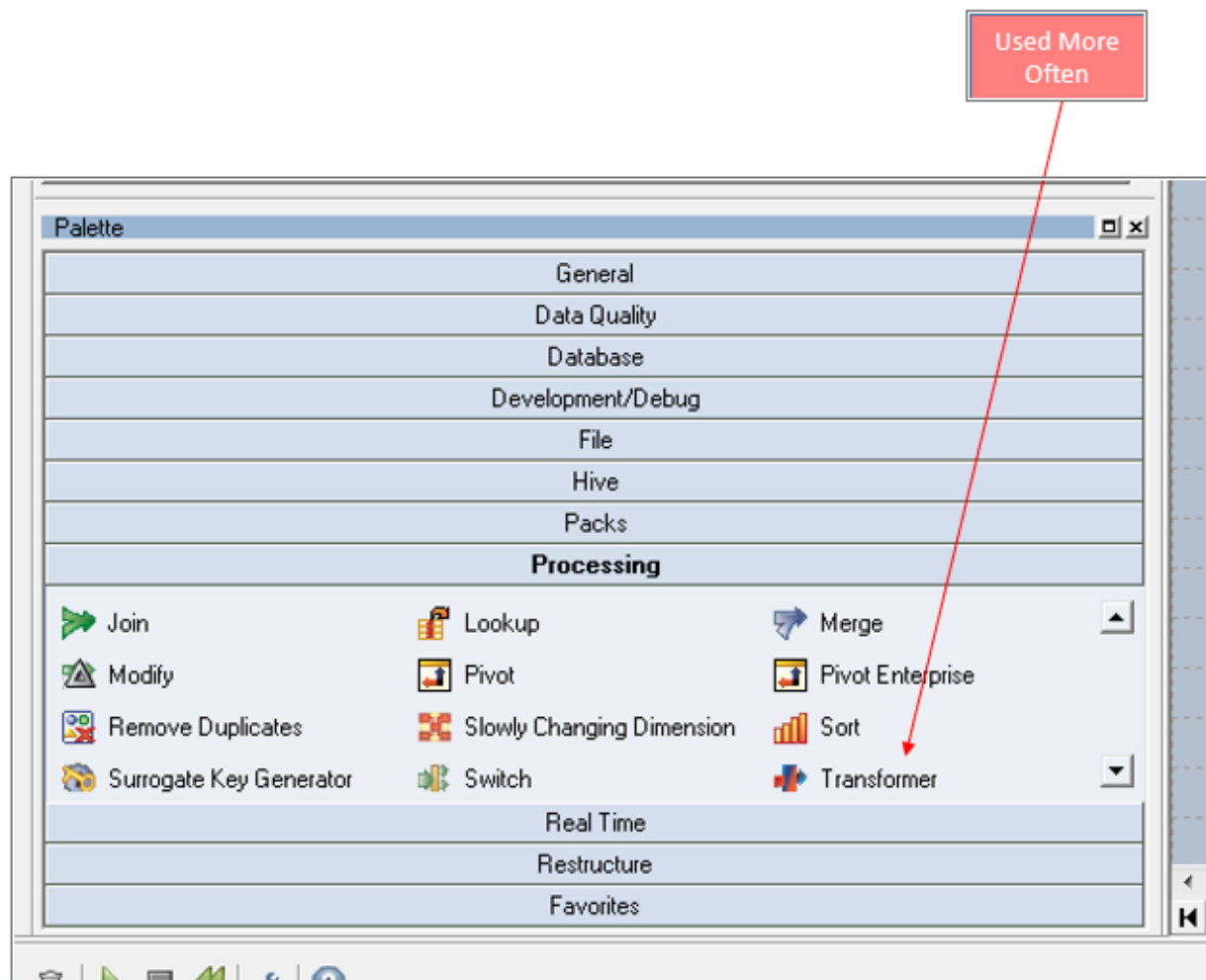
## 3.2 Singleton Bulk Inserts

- Selection of **ODBC** Connector will only allow singleton transactions
- Selection of **DRS** Connector will allow for selection of Bulk Insert option
- All stored procedures run on Azure Synapse Analytics must be executed via the Stored Procedure Database option from this selection screen

### 3.3 DRS Connection

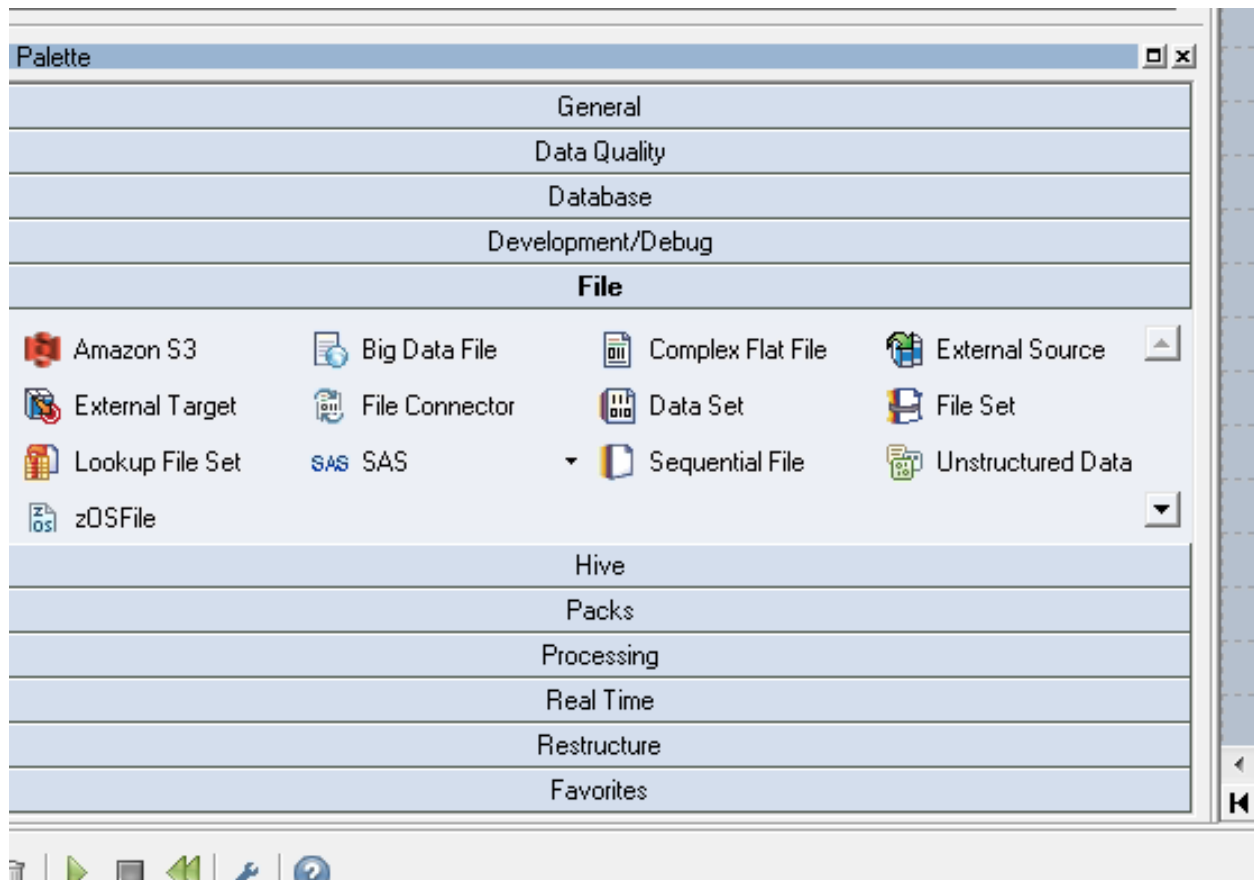
- Use the DRS Connector stage to access relational database management systems by using the native interfaces that are available for the corresponding databases.
- The choice of the database type is not determined when the stage is placed on the job canvas but is instead specified when the stage is configured.
- The DRS Connector stage supports IBM® DB2®, Oracle, and ODBC data sources. For other database types, you can configure the DRS Connector stage to use the ODBC database type and access the databases through the ODBC drivers that are included with InfoSphere Information Server.
- After the DRS Connector stage is placed on the job canvas, it needs to be configured to perform the operation intended by the job design.
- When a new DRS Connector stage is added to the server or parallel canvas, you must specify the connection information for the database that the stage connects to at run time.
- Information Links
  - [DRS Connection Overview](#)
  - [DRS Configuration](#)
  - [Settings for the DRS Connector stage](#)
  - [Defining a DRS Connector stage connection to the database](#)

### 3.4 Selection of Choices for Processing



## 4 Ingestion, Processing and Insertion via InfoSphere DataStage

### 4.1 Selection of File Options

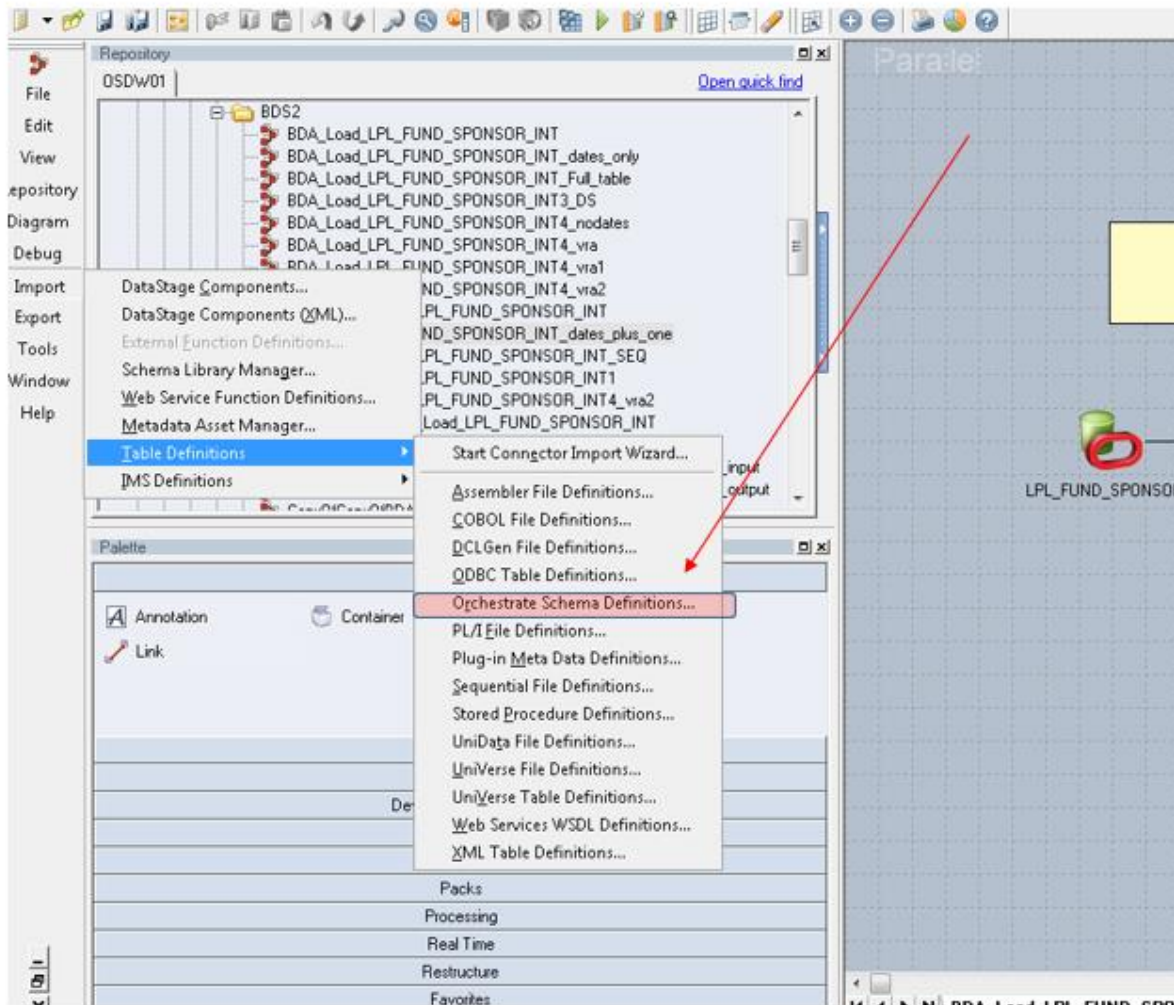


### 4.2 Setup Database Tables for Use

- Any Database table(s) used in this job must be imported into the job
- Every table that is used in the job must be imported individually.
- The name used to set up the ODBC connectivity for the database is utilized in the "DSN Name" box
- Authorization credentials are specified in each individual import stage

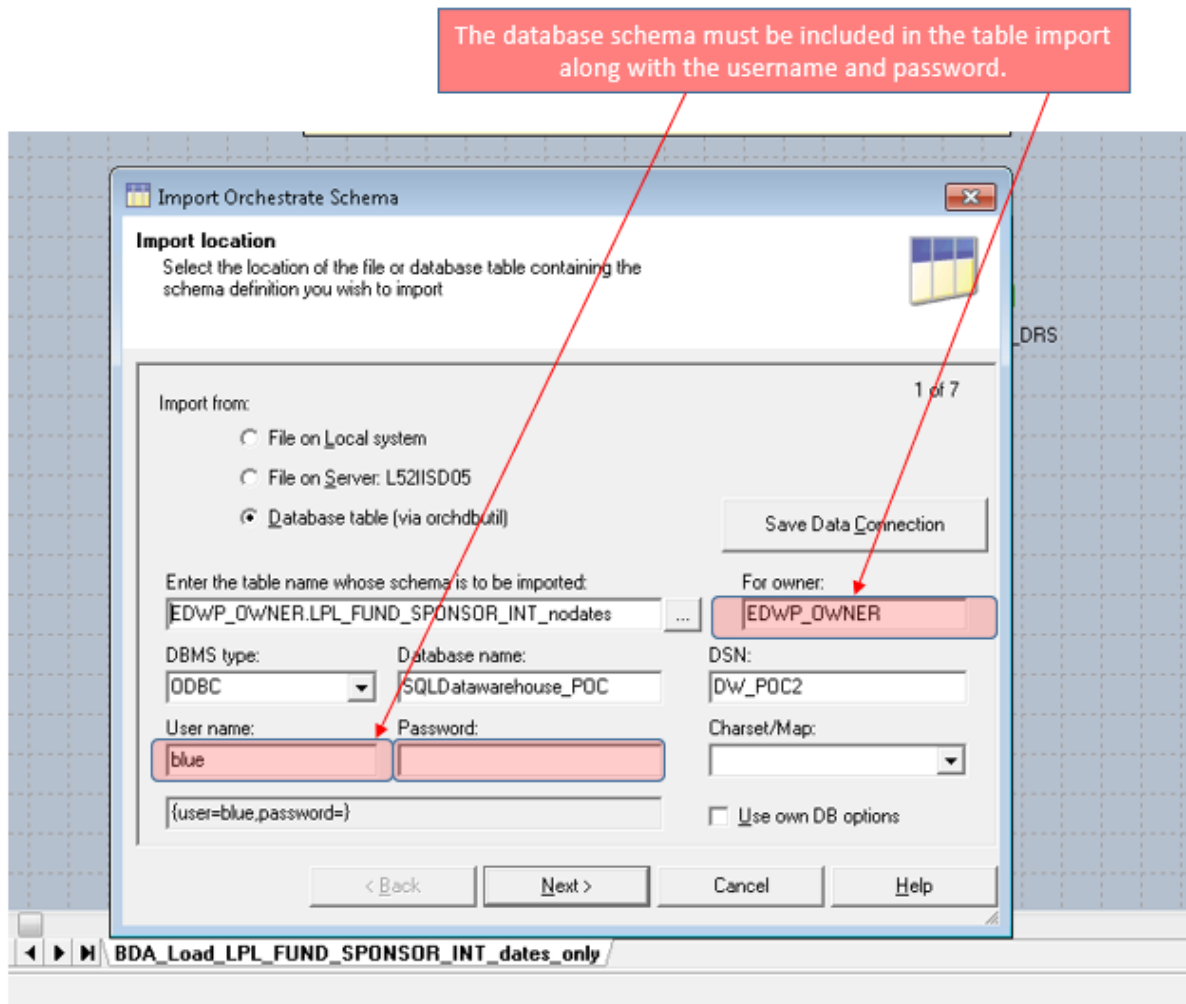
## 4.3 Import Table Selections

Every table that is used in the job must be imported individually. Select Import / Table Definitions / Orchestrate Schema Definitions.



## 4.4 Import each table used

The database schema must be included in the table import along with the username and password.

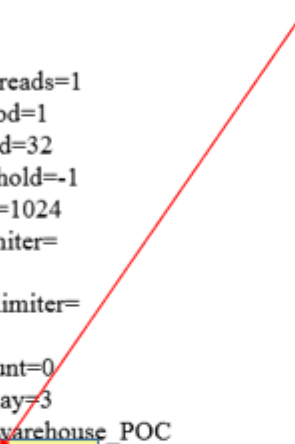


## 4.5 InfoSphere DataStage ODBC.ini driver parameters

In order to utilize the Bulk Load option, the **EnableBulkLoad** parameter must be set to 1.

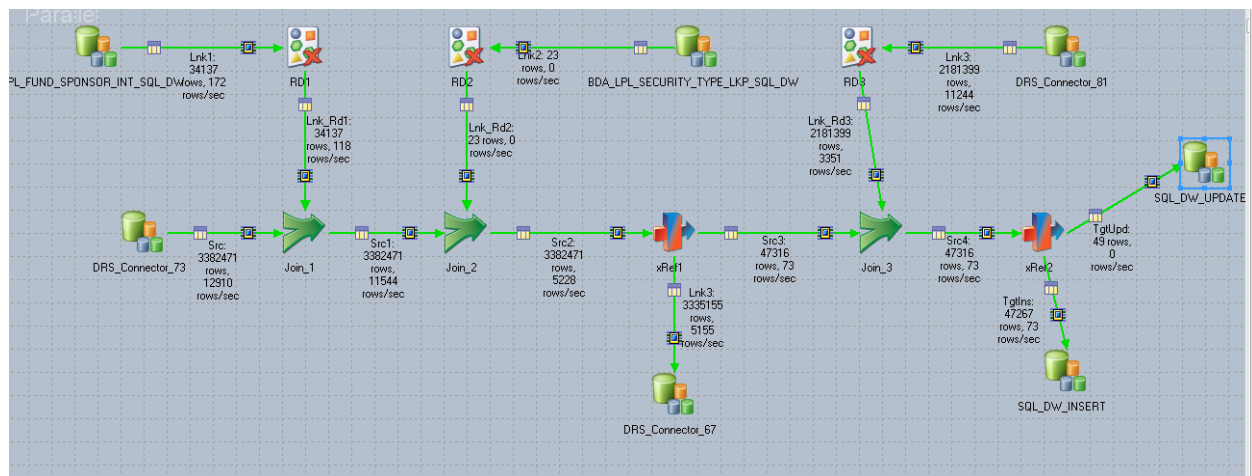


```
Driver=/software/IBM/InformationServer/Server/branded_odbc/lib/VMsqli00.so
Description=DataDirect SQL Server Wire Protocol driver
AlternateServers=
AlwaysReportTriggerResults=0
AnsiNPW=1
ApplicationName=
ApplicationUsingThreads=1
AuthenticationMethod=1
BulkBinaryThreshold=32
BulkCharacterThreshold=-1
BulkLoadBatchSize=1024
BulkLoadFieldDelimiter=
BulkLoadOptions=2
BulkLoadRecordDelimiter=
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=SQLDatawarehouse_POC
EnableBulkLoad=1
EnableQuotedIdentifiers=1
EncryptionMethod=1
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
FetchTSWTZasTimestamp=0
FetchTWFSasTime=1
GSSClient=native
HostName=sqldatawarehousepoc.database.windows.net
HostNameInCertificate=
InitializationString=
Language=
```

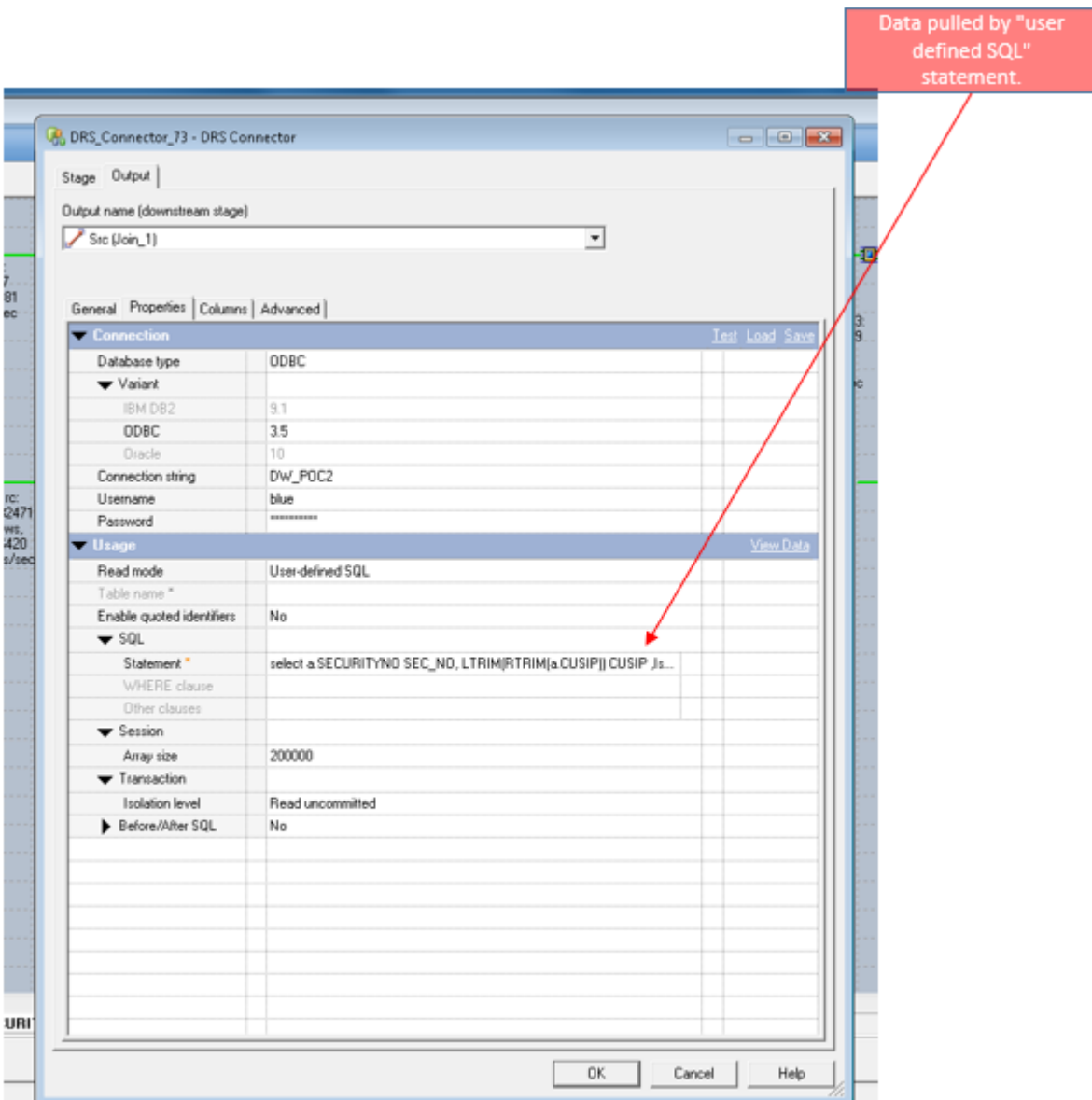


## 4.6 Converted Sample Job Update & Insert

Converted all Netezza sources and targets to Azure Synapse Analytics



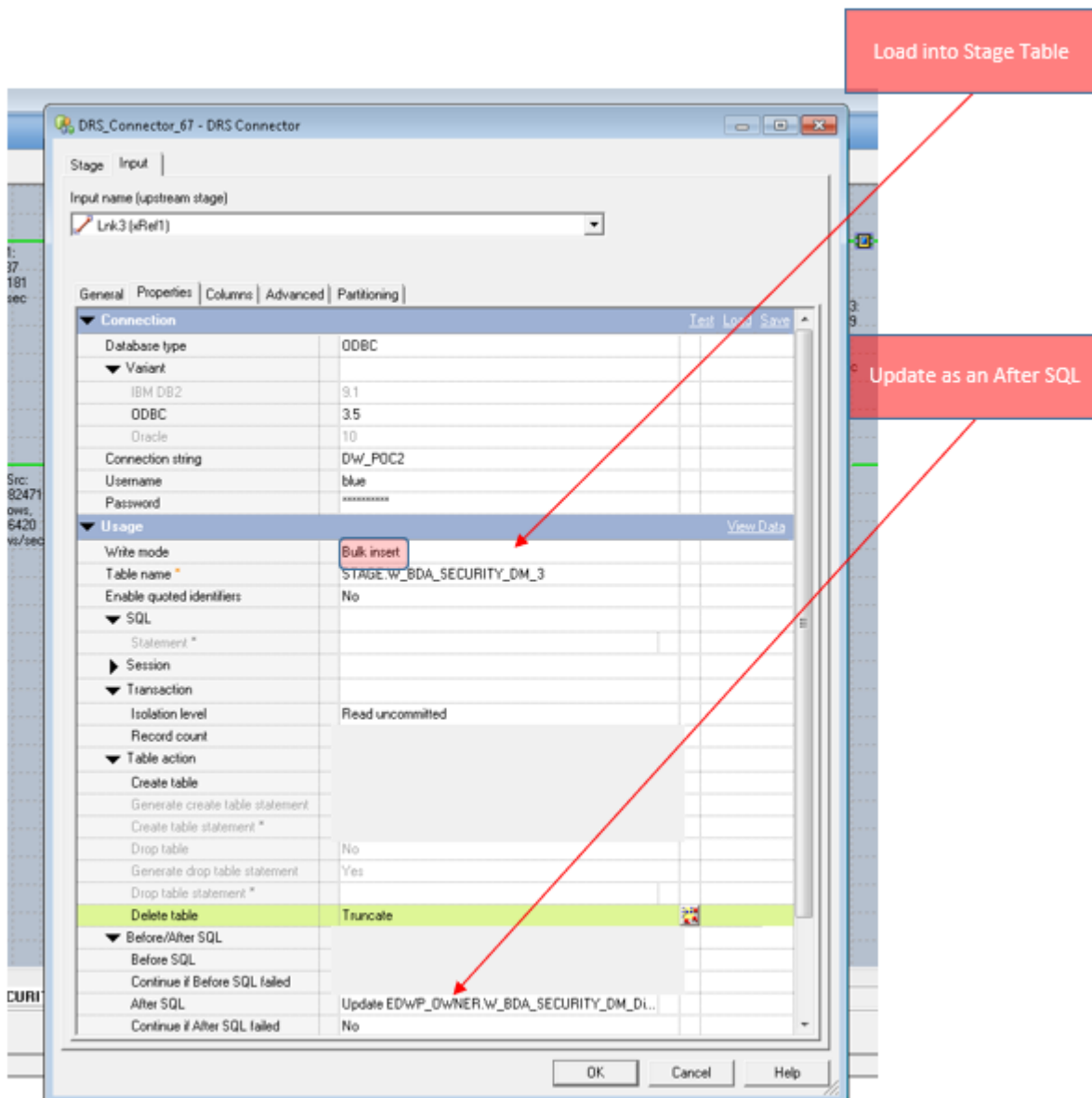
## 4.7 Utilize same method of pulling information



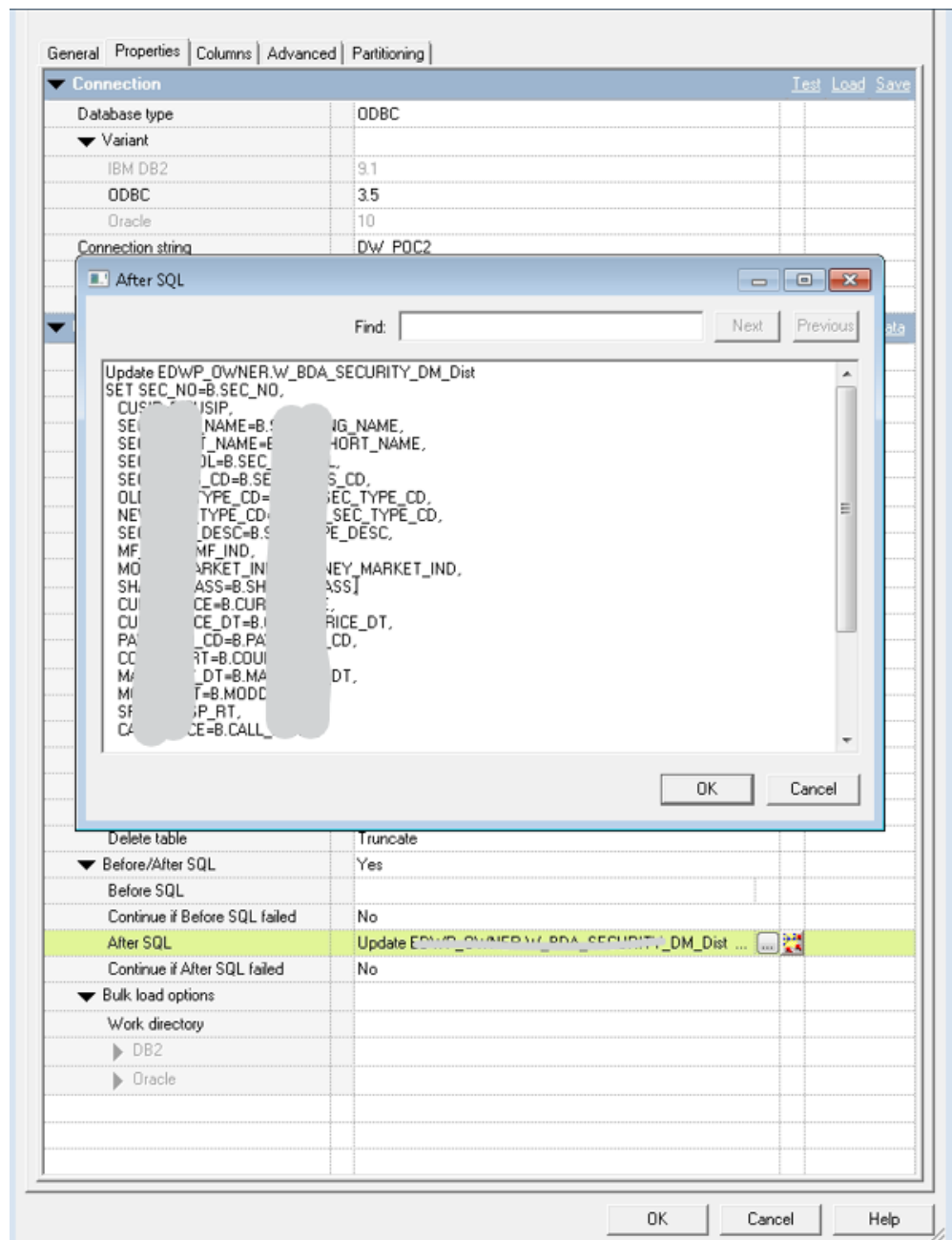
## 4.8 For updating data, first, ingest into staging table using Bulk Insert.

In order to update data, the data must first be loaded into a staging table (as designated in the table name) using a "Bulk Insert" statement in the "write mode" column. The update SQL

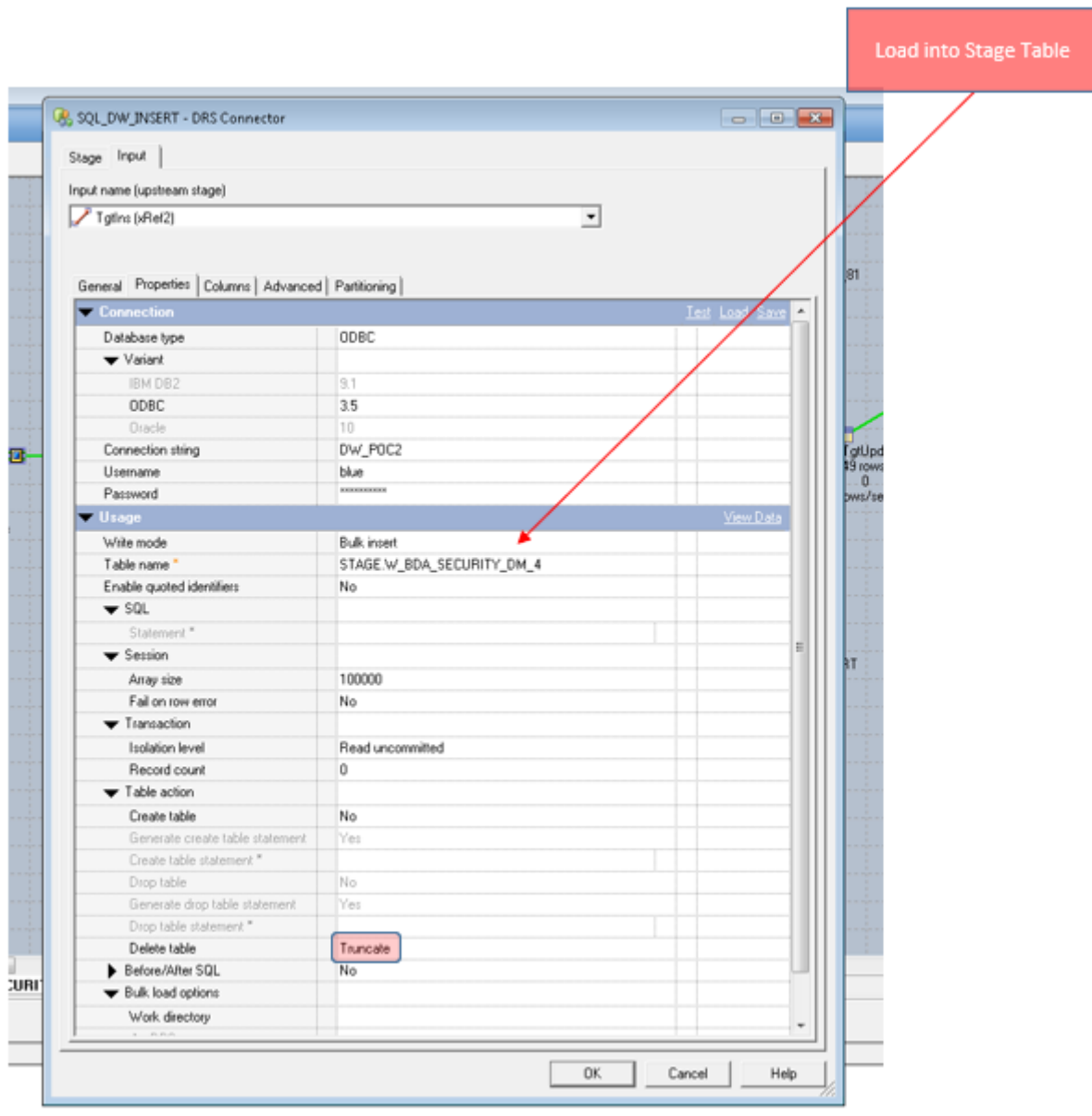
statement is placed in the After SQL box at the bottom. This will allow the Bulk Insert statement to load the data to the staging table represented in the Table Name box and then will process the update after all the data has reached the staging table.



## 4.9 Detail example of the Update Statement



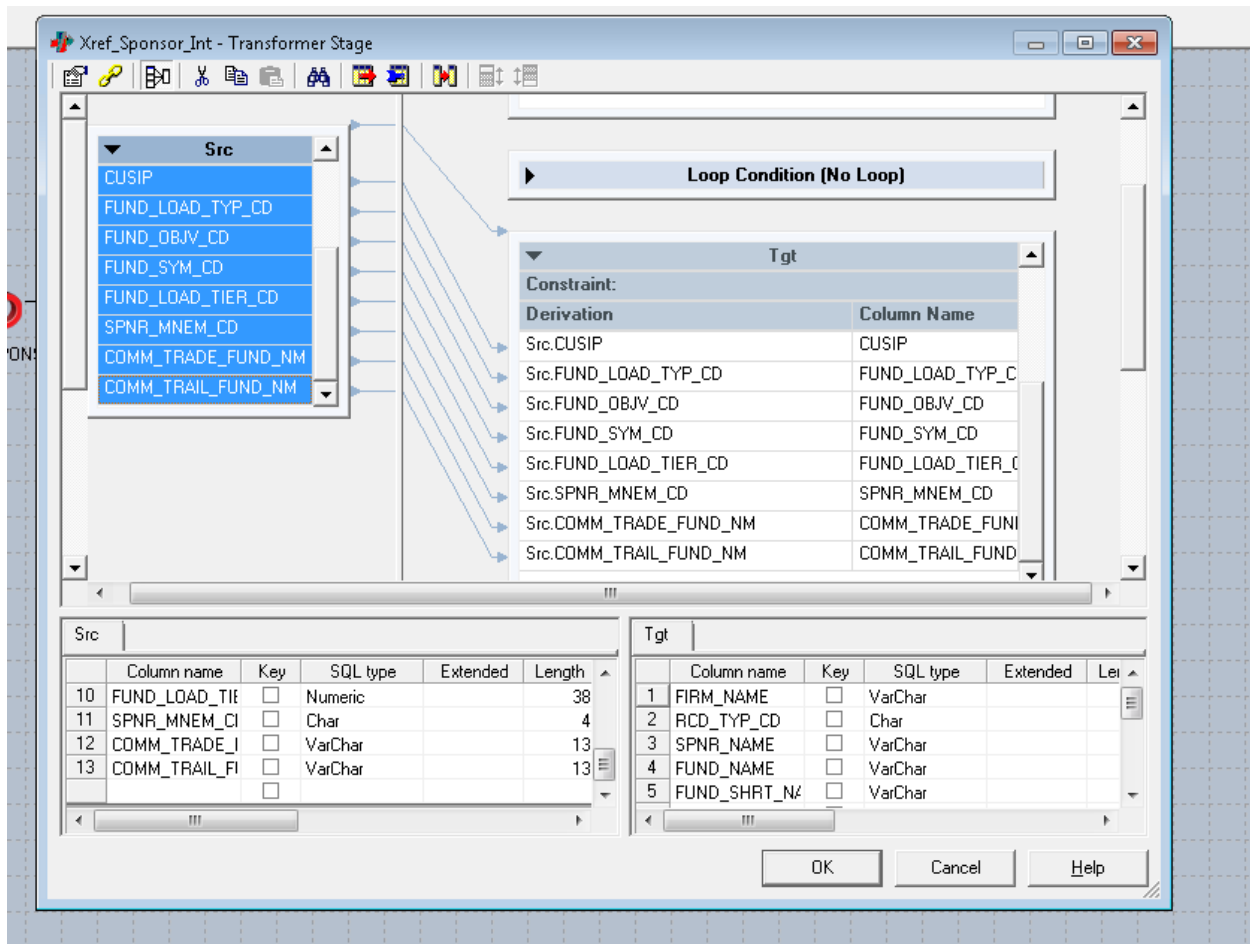
## 4.10 Detail example of the Insert statement



## 4.11 Changes to Transformers

- When changing the sources and targets, assure that you are replacing the existing stages by deleting them and then pasting the new ones in.
- Do not delete the connection arrows.
- Open each Transformer processing module and change necessary sources and/or targets information

## 4.12 Change Transformer Processor



## 4.13 Executing Stored Proc on Azure Synapse Analytics

- If an additional step is required to creating a sequence
- Create the stored procedure on Azure Synapse Analytics SQL Pool
- Create a job to execute the stored procedure
- Set up a "Sequence" job to execute the job that inserts and updates and then executes the stored proc

## 4.14 Stored Procedure to Build Key and Insert

```

CREATE PROCEDURE STAGE_INSERT AS
BEGIN
DECLARE @security_id bigint
Select @security_id = MAX(ROW_ID) FROM SAMPLE_SCHEMA.W_BDA_SECURITY_DM_Dist
--
if exists(select 1 from sys.tables where name = 'W_BDA_SECURITY_DM_4_ROWNUM') Drop Table STAGE.W_BDA_SECURITY_DM_4_ROWNUM
create table STAGE.W_BDA_SECURITY_DM_4_ROWNUM
with (distribution = replicate_heap) as
select (@security_id + row_number() over(order by sec_no, cusip asc)) as NEW_ROW_NUM,
SEC_NO, CUSIP,
SEC_LONG_NAME,
SEC_SHORT_NAME,
SEC_SYMBOL,
SEC_CLASS_CD,
OLD_SEC_TYPE_CD,
NEW_SEC_TYPE_CD,
SEC_TYPE_DESC,
MF_IND, MONEY_MARKET_IND, SHARE_CLASS, CURR_PRICE, CURR_PRICE_DT, PAY_FREQ_CD, COUPON_RT, MATURITY_DT,
MODDY_RT, SP_RT, CALL_PRICE, DIVIDEND, FACTOR_DT, FACTOR, NOTE, INTEGRATION_TSTP
FROM STAGE.W_BDA_SECURITY_DM_4

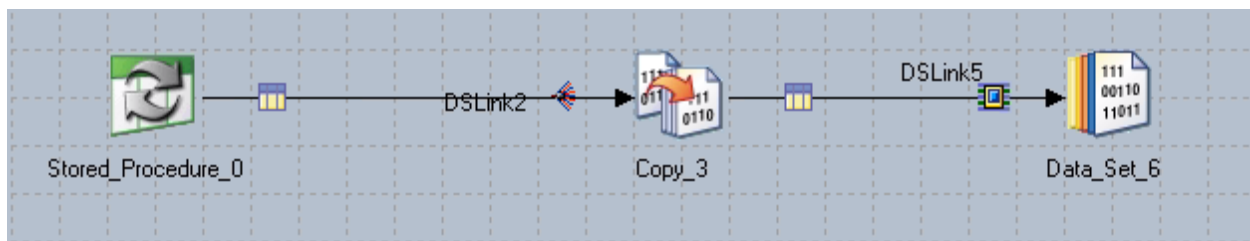
INSERT INTO SAMPLE_SCHEMA.W_BDA_SECURITY_DM_Dist
(ROW_ID,
SEC_NO,
CUSIP,
SEC_LONG_NAME,
SEC_SHORT_NAME,
SEC_SYMBOL,
SEC_CLASS_CD,
OLD_SEC_TYPE_CD,
NEW_SEC_TYPE_CD,
SEC_TYPE_DESC,
MF_IND, MONEY_MARKET_IND, SHARE_CLASS, CURR_PRICE, CURR_PRICE_DT, PAY_FREQ_CD, COUPON_RT, MATURITY_DT,
MODDY_RT, SP_RT, CALL_PRICE, DIVIDEND, FACTOR_DT, FACTOR, NOTE, INTEGRATION_TSTP
)
SELECT
B.NEW_ROW_NUM, B.SEC_NO, B.CUSIP, B.SEC_LONG_NAME,
B.SEC_SHORT_NAME, B.SEC_SYMBOL,
B.SEC_CLASS_CD, B.OLD_SEC_TYPE_CD, B.NEW_SEC_TYPE_CD, B.SEC_TYPE_DESC, B.MF_IND, B.MONEY_MARKET_IND,
B.SHARE_CLASS, B.CURR_PRICE, B.CURR_PRICE_DT, B.PAY_FREQ_CD, B.COUPON_RT, B.MATURITY_DT,
B.MODDY_RT, B.SP_RT, B.CALL_PRICE, B.DIVIDEND, B.FACTOR_DT, B.FACTOR, B.NOTE, B.INTEGRATION_TSTP
FROM STAGE.W_BDA_SECURITY_DM_4_ROWNUM B
END

```

Stage Table with sequencer

Target Table

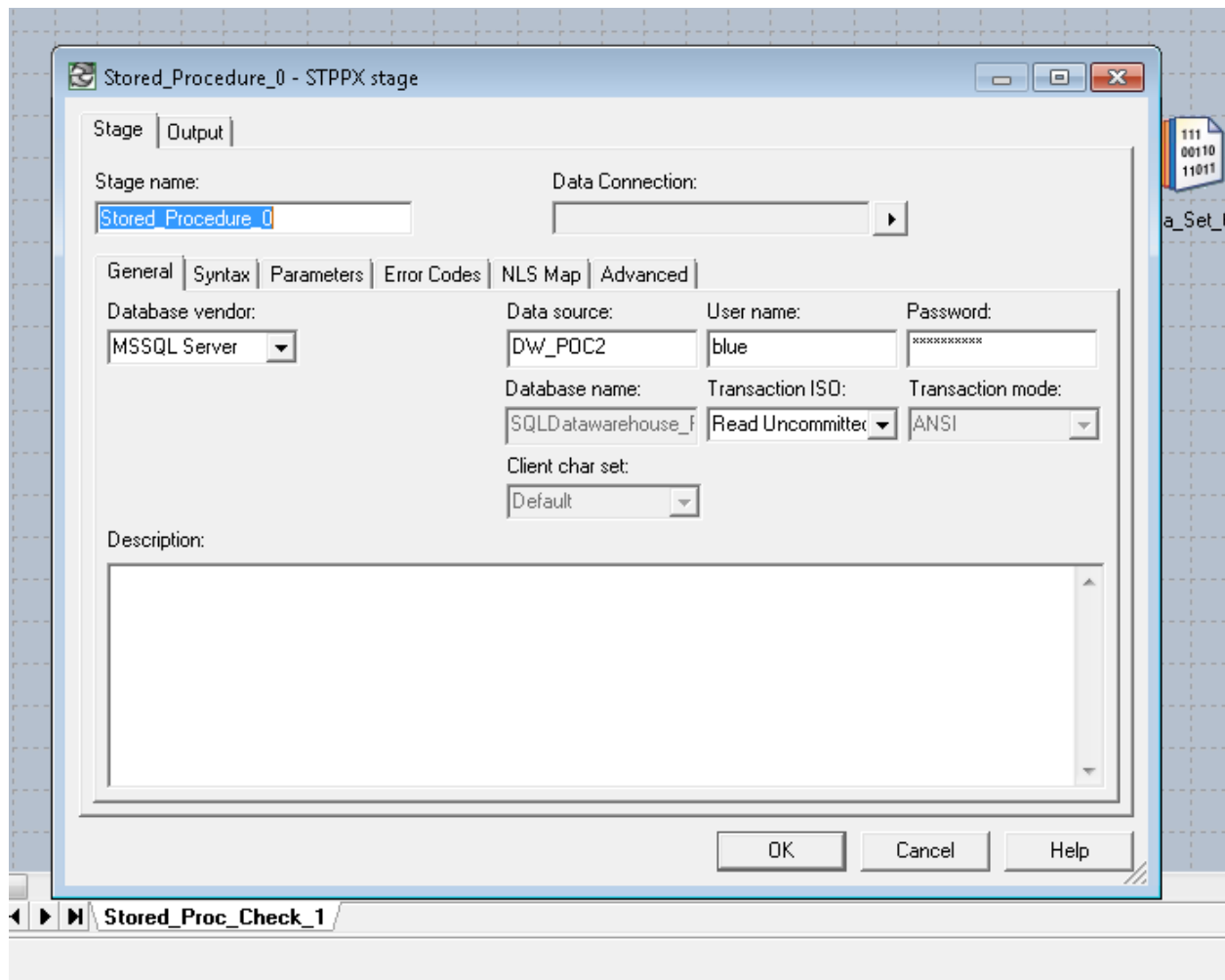
## 4.15 InfoSphere DataStage Job to Execute a Stored Procedure



## 4.16 Stored Procedure General Setup

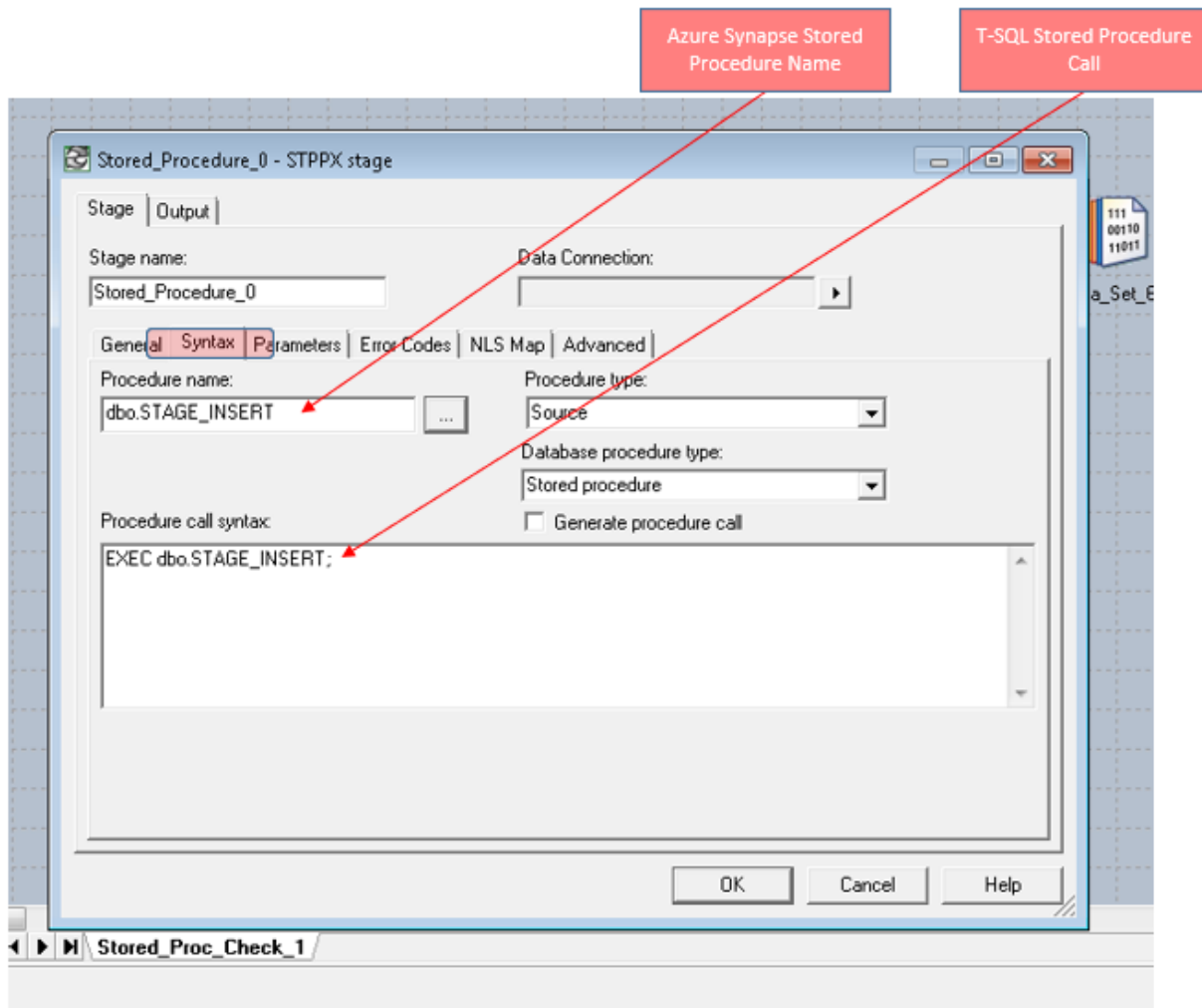
How is a Stored Proc setup for execution on Azure Synapse Analytics.





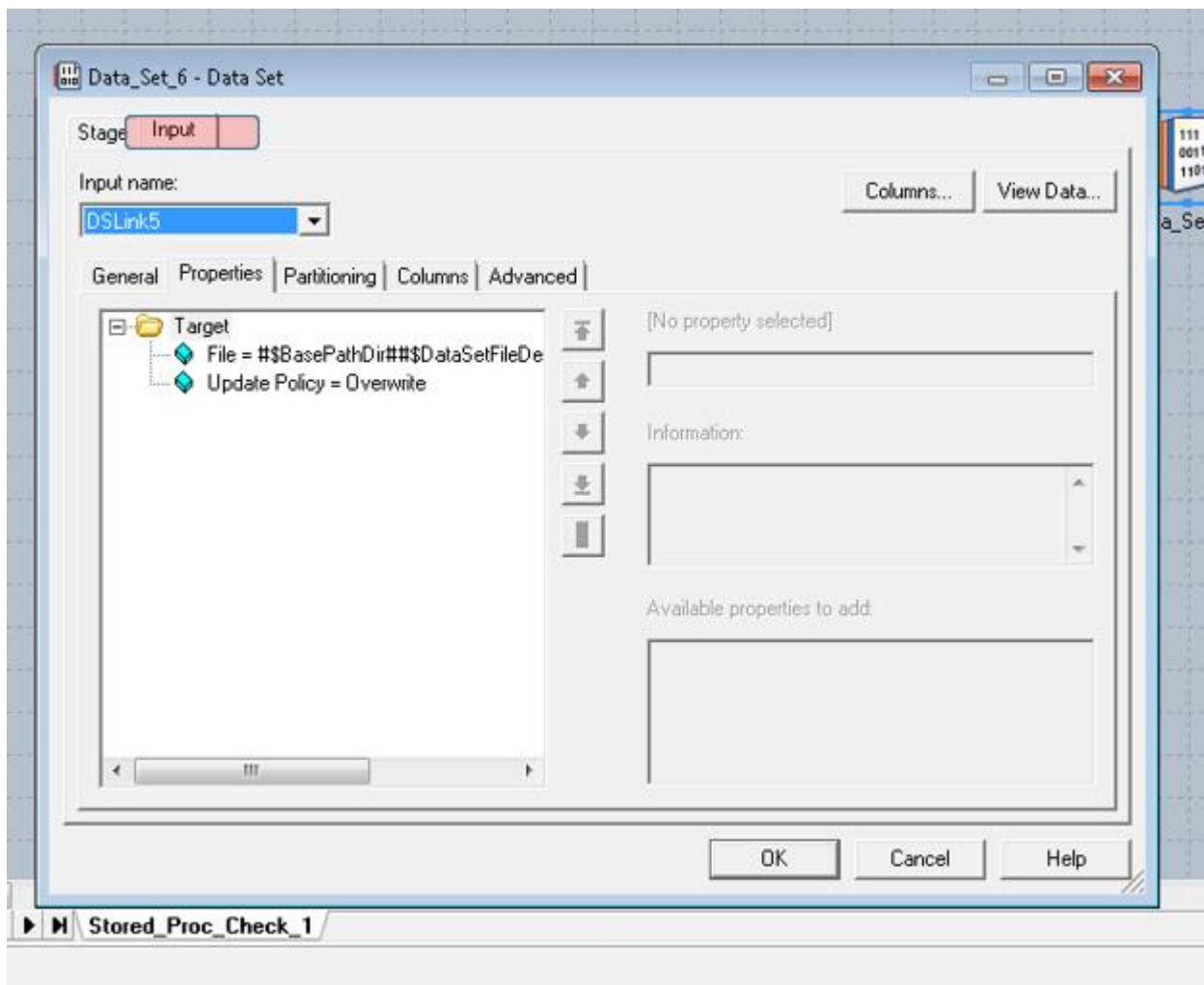
## 4.17 Set Up Syntax for Stored Procedure Execution

The Procedure Name is the name of the stored proc on Azure Synapse Analytics. The Procedure Call Syntax shows the execute statement for the stored proc.

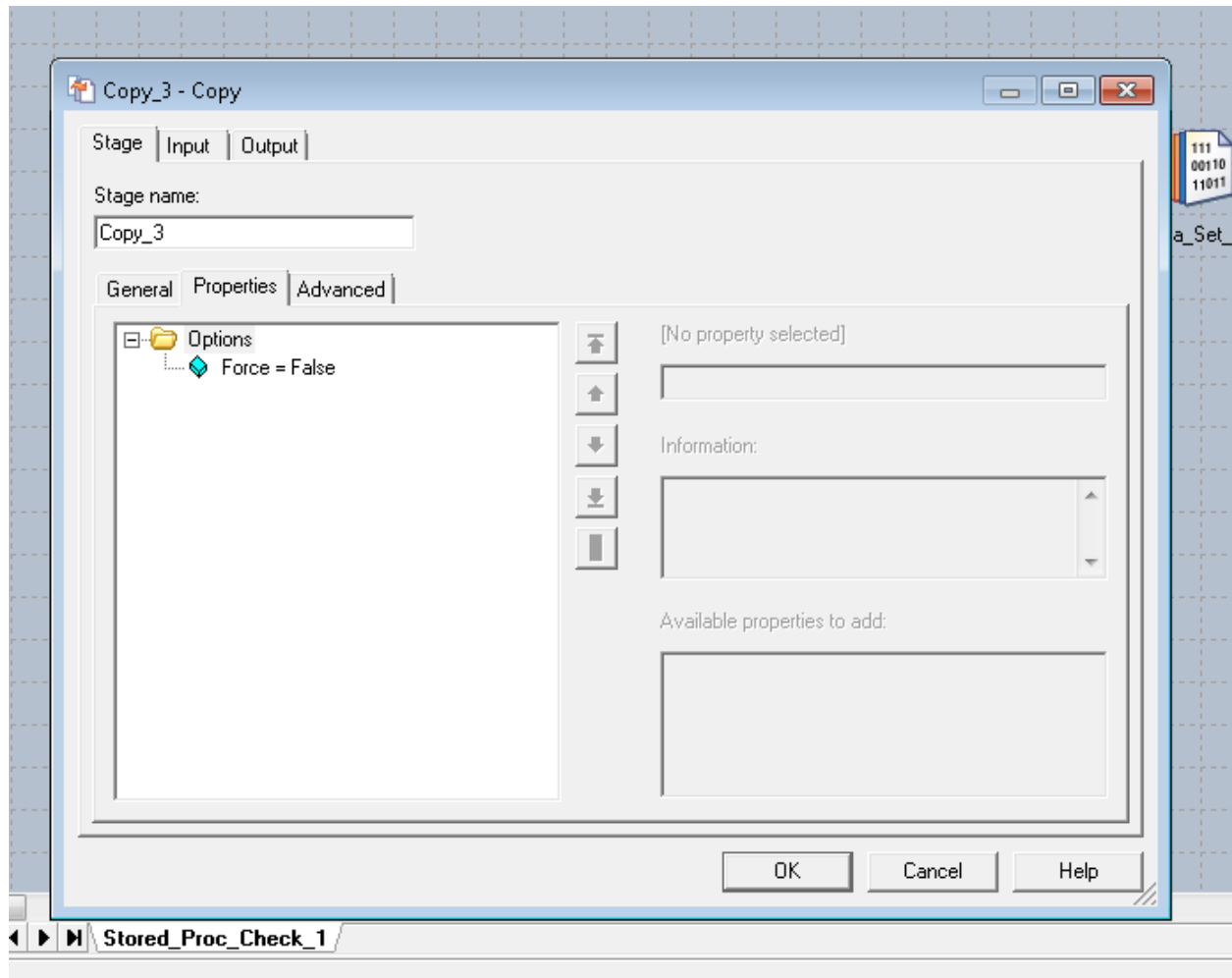


## 4.18 Set up "Copy" process Input Properties

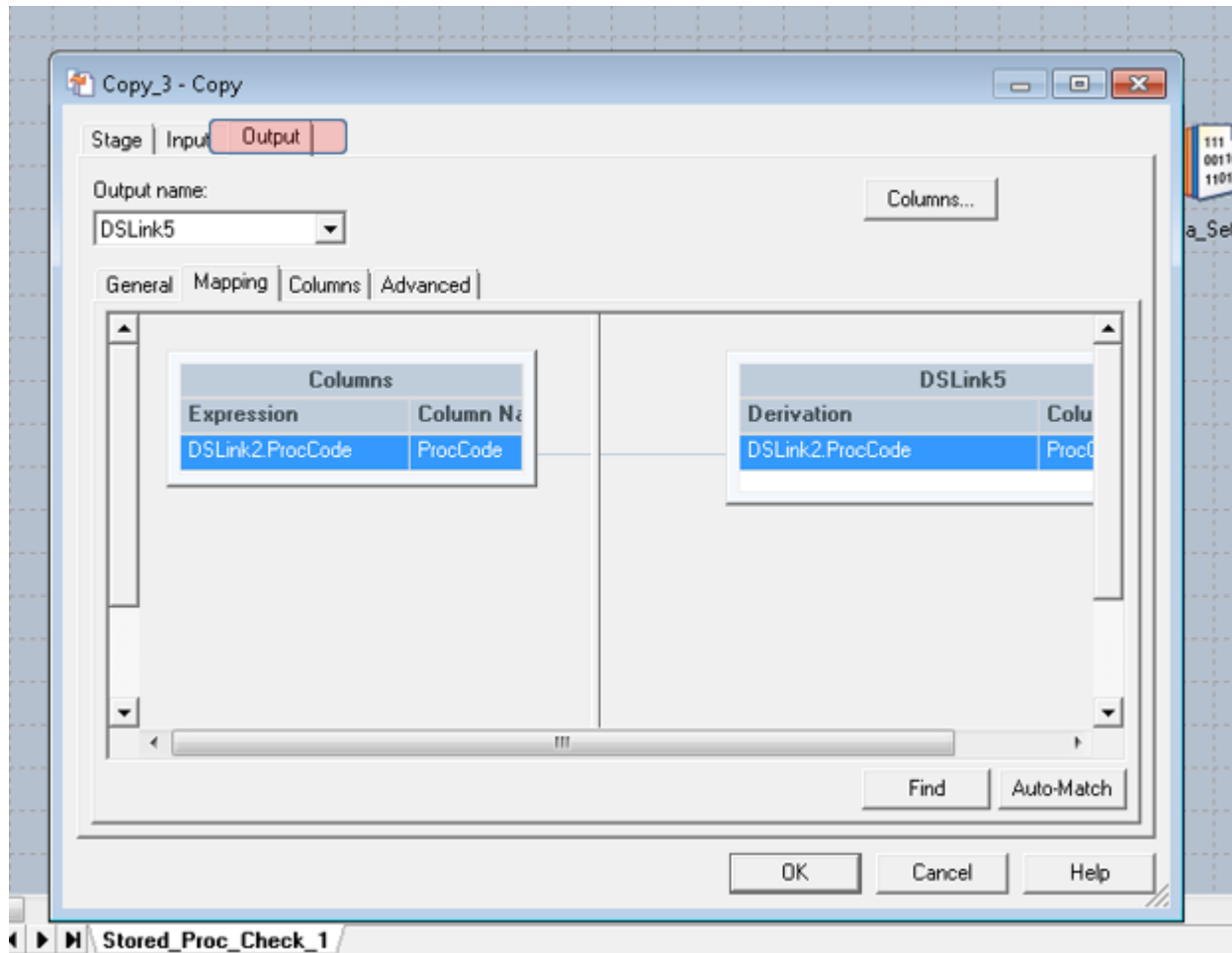
The properties set up here were defaulted on creation of this process.



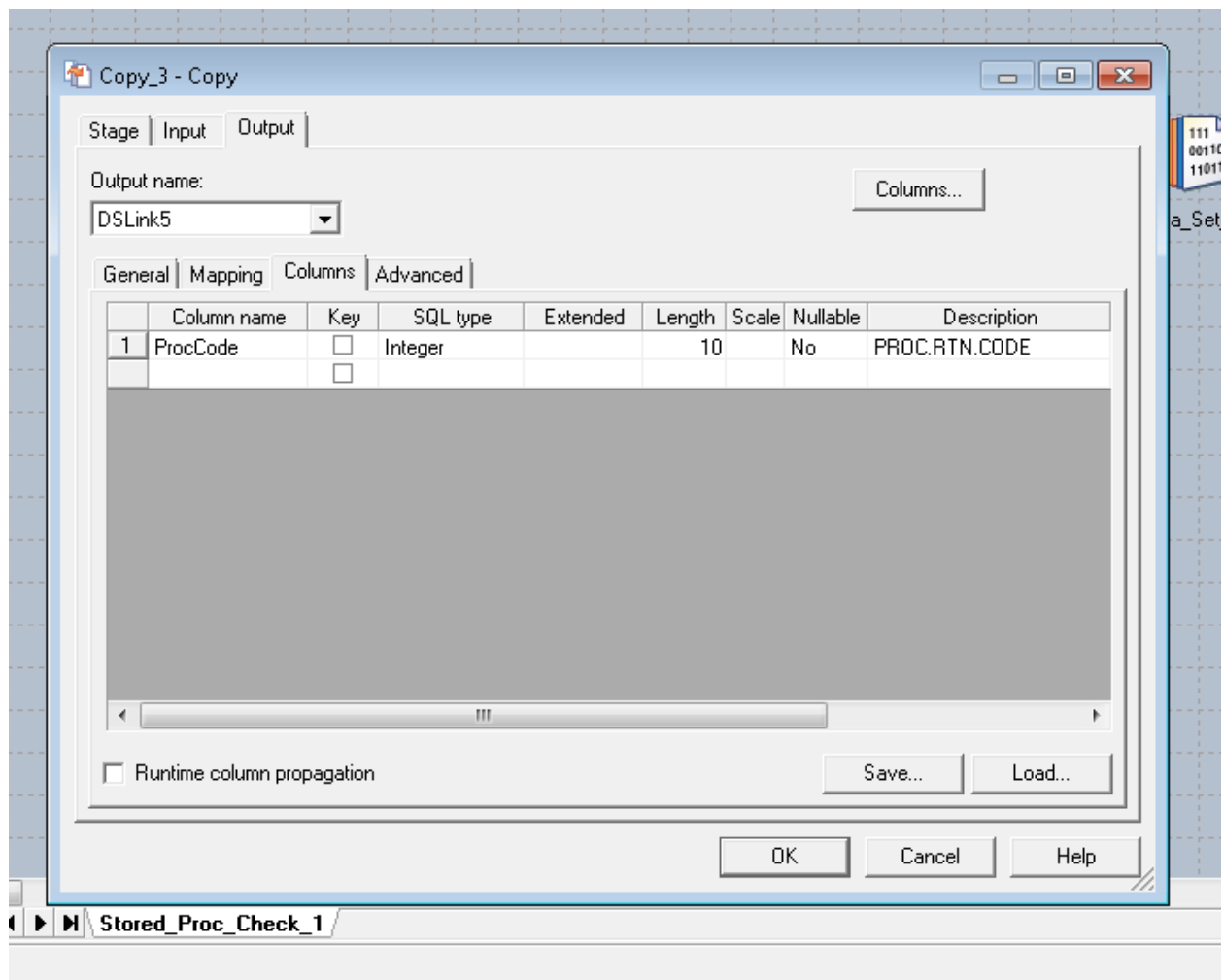
## 4.19 Set up "Copy" Process Output Properties for Stored Proc



## 4.20 Set up "Copy" process for Output Mapping



## 4.21 Set up "Copy" process Output Columns



## 5 Large Dataset Ingestion

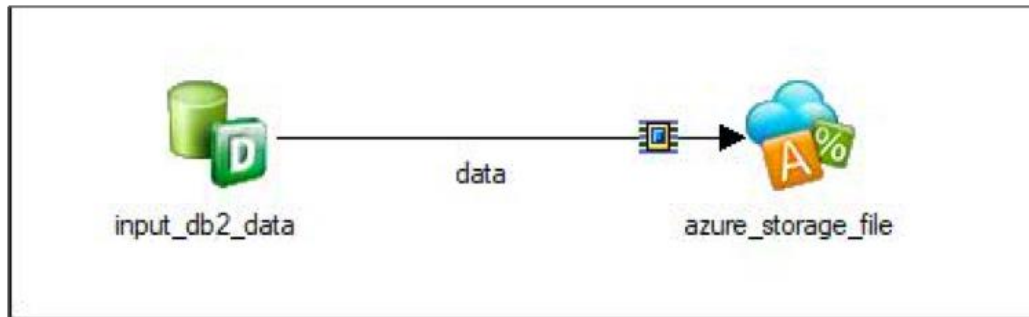
### 5.1 Large Data Ingestion in Azure Synapse Strategy

- Large Data Set ingestion should follow a pattern
  - Land data into Azure Blob Container or Data Lake Store
  - Ingest into a staging table in Azure Synapse Analytics using COPY INTO
  - CTAS or Insert into Final Table
  - Refresh statistics as needed
- InfoSphere DataStage can orchestrate the reading from source (e.g. flat files) and ingesting into Blob Storage as a first step



### 5.2 Writing Data into Azure Storage with InfoSphere DataStage

- Large Data Set ingestion can take advantage of COPY INTO process from Azure Synapse Analytics from data landed in Blob Storage
- Requirements: IBM InfoSphere Information Server InfoSphere DataStage 11.7fp1 and above
- High Level Steps:
  - Configure Azure Storage Connector Connection Properties
  - Configure Azure Storage Connector to write to Azure Blob Storage
  - Additional Configuration for Parallel Write
- [Documentation Link](#)



### 5.3 Bulk Loading into Azure Synapse with InfoSphere DataStage

- COPY INTO statement can be executed:
  - Within a Stored Procedure
  - Or as an ad-hoc T-SQL
- COPY INTO will take the data from Blob Storage/ADLS and ingest into Azure Synapse Analytics in parallel
- Data should be landed on an uncompressed rowstore table for performance reasons
- T-SQL should execute a last step of ingesting or CTAS into a final table.
- Recompute of statistics is always recommended when data has changed significantly

### 5.4 Bulk Loading with COPY INTO statement



```

CREATE SCHEMA stage;
-- Create the Staging Table
-- Drop Table [stage].[SalesData]
Create table [stage].[SalesData](
    Region varchar(50)
    ,Product varchar(50)
    ,SaleDate date
    ,Amount DECIMAL (20,10)
)
with (distribution = round_robin, HEAP);

```

No compression

```

/* Documentation
https://docs.microsoft.com/en-us/sql/t-sql/statements/copy-into-transact-sql?view=azure-sqldw-latest
*/

```

```

COPY INTO [stage].[SalesData] (Region 1, Product 2, SaleDate 3, Amount 4)
FROM 'https://account.blob.core.windows.net/yourcontainer'
WITH (
    FILE_TYPE = 'CSV',
    CREDENTIAL=(IDENTITY='Shared Access Signature', SECRET='Your secret here'),
    FIELDQUOTE = '"',
    FIELDTERMINATOR=',',
    ROWTERMINATOR = '\n',
    Identity_INSERT = 'OFF',
    ENCODING = 'UTF8',
    DATEFORMAT = 'ymd',
    FIRSTROW = 2
);

```

Not necessary if using  
AAD / Role based  
access control

```

CREATE SCHEMA Sales;
-- CTAS with HASH distribution on Date and Columnar format
-- DROP TABLE Sales.Transactions
CREATE TABLE Sales.Transactions
WITH
(
    DISTRIBUTION = HASH(SaleDate)
    ,CLUSTERED COLUMNSTORE INDEX
) AS SELECT * FROM [stage].[SalesData];

```

CTAS into final table,  
with compression and  
distribution

Statistics are created

```

Create Statistics stat_stage_salesdata_Region on Sales.Transactions (Region) with fullscan;
Create Statistics stat_stage_salesdata_Product on Sales.Transactions (Product) with fullscan;
Create Statistics stat_stage_salesdata_SaleDate on Sales.Transactions (SaleDate) with fullscan;
Create Statistics stat_stage_salesdata_Amount on Sales.Transactions (Amount) with fullscan;

```

## 6 Samples Scripts

### 6.1 Ingestion into Azure Synapse Analytics SQL Pool

```
CREATE SCHEMA stage;
-- Create the Staging Table
-- Drop Table [stage].[SalesData]
Create table [stage].[SalesData](
    Region varchar(50)
    ,Product varchar(50)
    ,SaleDate date
    ,Amount DECIMAL (20,10)
)
with (distribution = round_robin, HEAP);

/* Documentation
https://docs.microsoft.com/en-us/sql/t-sql/statements/copy-into-transact-sql?view=azure-sqldw-latest
*/

COPY INTO [stage].[SalesData] (Region 1, Product 2, SaleDate 3, Amount 4)
FROM 'https://account.blob.core.windows.net/yourcontainer'
WITH (
    FILE_TYPE = 'CSV',
    CREDENTIAL=(IDENTITY= 'Shared Access Signature', SECRET='Your secret here'),
    FIELDQUOTE = '"',
    FIELDTERMINATOR=',',
    ROWTERMINATOR = '\n',
    Identity_INSERT = 'OFF',
    ENCODING = 'UTF8',
    DATEFORMAT = 'ymd',
    FIRSTROW = 2
);
```

```

CREATE SCHEMA Sales;

-- CTAS with HASH distribution on Date and Columnar format
-- DROP TABLE Sales.Transactions
CREATE TABLE Sales.Transactions
WITH
(
    DISTRIBUTION = HASH(SaleDate)
    ,CLUSTERED COLUMNSTORE INDEX
) AS SELECT * FROM [stage].[SalesData];

Create Statistics stat_stage_salesdata_Region on Sales.Transactions (Region) with
fullscan;

Create Statistics stat_stage_salesdata_Product on Sales.Transactions (Product) with
fullscan;

Create Statistics stat_stage_salesdata_SaleDate on Sales.Transactions (SaleDate) with
fullscan;

Create Statistics stat_stage_salesdata_Amount on Sales.Transactions (Amount) with
fullscan;

```