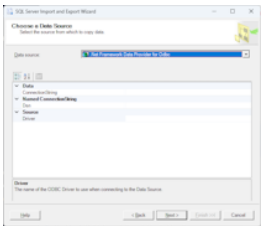
INSERT INTO...SELECT FROM between two compatible tables had obstacles and required busy work:

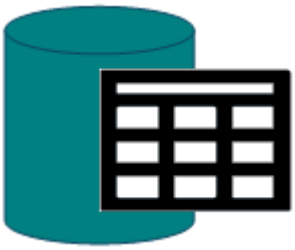
Excel Spreadsheet From SourceTable

1. User chooses an .XLSX file as a data source



SQL Server Import and Export Wizard

2. User chooses a table in SQL Server to export to



StageTable in SQL Server (has same structure as .XLSX file

3. At this point StageTable has the Excel data in it, but the wizard often saves data in less-than-optimal data types, e.g. integer data from XLSX gets saved as float in the stage table.

while the export allows a lot of field-level configuration, this is time-consuming, so few, if any tweaks to field-level settings are made. We want to go through the wizard as quickly as possible.

4. Past experience has shown that a direct import from the wizard into the DestTable leads to type conflicts that prevent the import from succeeding. The additional step of using a staging table avoids a lot of grief.

Run an INSERT INTO DestTable SELECT * FROM SourceTable statement
since DestTable matches SourceTable in column names and number of columns and is close enough type-wise that implicit type conversions should work

5. Unfortunately there are obstacles

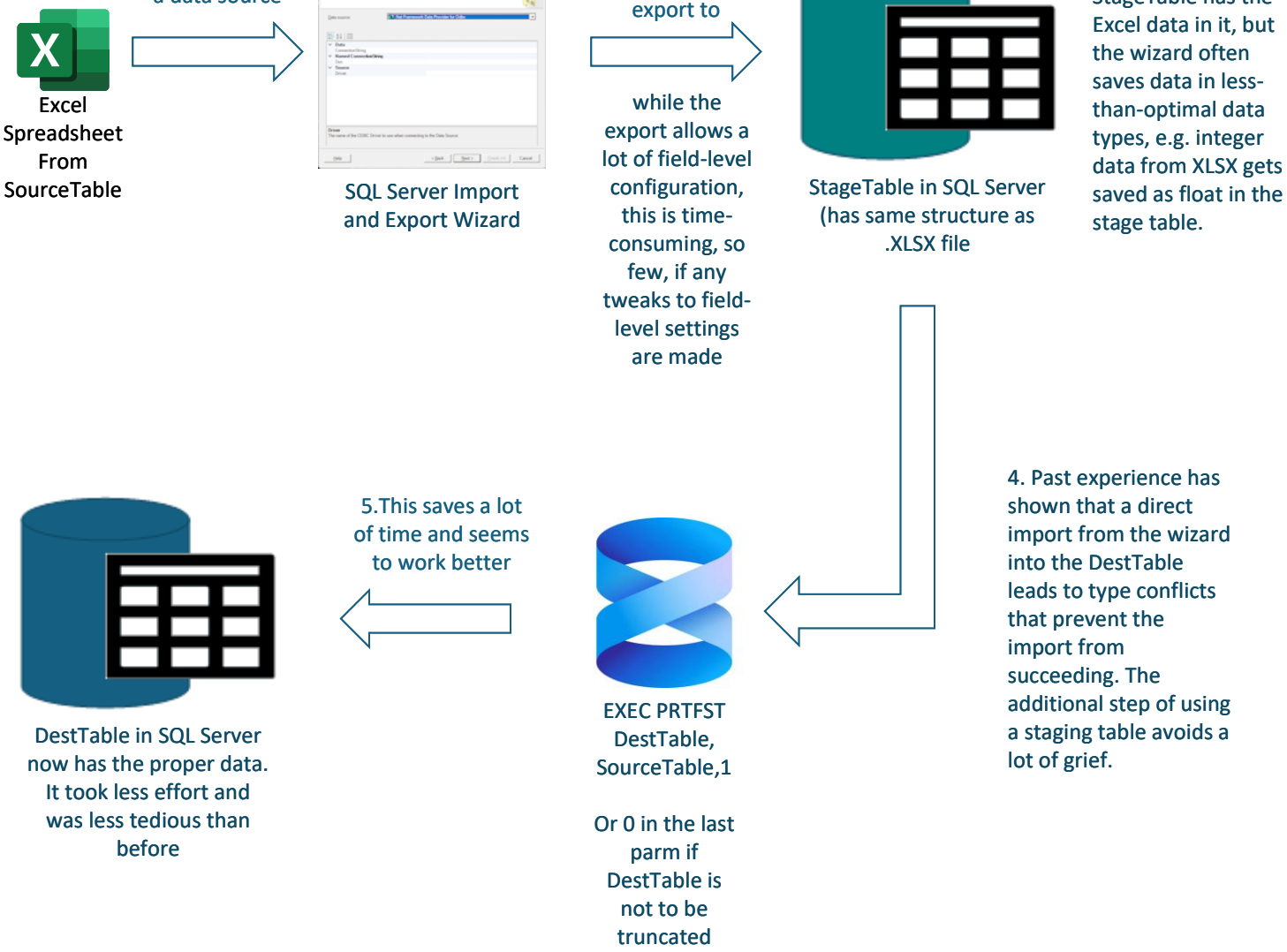
A. The destination table has an identity column. INSERT INTO...SELECT FROM won't work. You must turn IDENTITY_INSERT ON and use a specific column list

B. You may need to TRUNCATE the destination table first, but tend to forget. You end up with duplicates or PRIMARY KEY VIOLATION errors.

C. You can't just use a template because some tables have IDENTITY columns and some don't.

D. Manually adding column lists is tedious. I used to do this, then keep copies of scripts, but with hundreds of tables, I can't always find the script I need or make multiple copies of the same script.

Using PROCEDURE PRTFST overcame many of those problems:



PRTFST uses dynamic SQL to create an INSERT INTO DestTable(<specific column list>) SELECT * FROM StageTable <specific column list> statement , so you probably cannot use this in some business environments with security measures like those used against SQL injections. There may be other restrictions based on policies or access rights.

But if you are testing in your own SSMS environment that you uses as a mockup, you do not have to worry about those limitations.

1. It takes care of whether or not you need to create an SET IDENTITY_INSERT ON/OFF pair of statements by looking at the settings of DestTable.
2. You have to provide a parameter to tell the stored proc to truncate or not truncate DestTable first, so you are less likely to forget this setting.
3. You can always do an INSERT INTO DestTable(<specific column list>) SELECT * FROM StageTable <specific column list> statement between compatible tables.