# SQL to Tableau Data Converter

## Overview

**Project Name:** [Your Project Name]

**Author:** Adrien Choinière

**Date:** 2024-07-15

Table des matières

[SQL to Tableau Data Converter 1](#_Toc171958272)

[Overview 1](#_Toc171958273)

[**SQL to Tableau Conversion Application Documentation** 3](#_Toc171958274)

[**Introduction** 3](#_Toc171958275)

[**Prerequisites** 3](#_Toc171958276)

[**Usage Instructions** 3](#_Toc171958277)

[**Launching the Application** 3](#_Toc171958278)

[Application Menu 3](#_Toc171958279)

[Detailed Steps 4](#_Toc171958280)

[Explanation of Tags 4](#_Toc171958281)

[Excel Section 5](#_Toc171958282)

[Example Section 6](#_Toc171958283)

[Conclusion 12](#_Toc171958284)

**SQL to Tableau Conversion Application Documentation**

**Introduction**

Welcome to the SQL to Tableau Conversion Application documentation. This document provides detailed instructions on how to use the application to convert SQL content into Excel spreadsheets.

**Prerequisites**

Ensure the following prerequisites are met before using the application:

* .NET Framework installed on your machine.
* Access to SQL files (*.sql) and Excel files (*.xlsx) for input and output.  
   (Not tested on LibreOffice)

**Usage Instructions**

**Launching the Application**

You can launch the application either through the executable (Sql\_vers\_tableau\_words.exe) or via PowerShell.

**Executable Method**

1. **Navigate to the Application Directory:**
   * Open File Explorer and go to the directory where Sql\_vers\_tableau\_words.exe is located.
2. **Run the Application:**
   * Double-click on Sql\_vers\_tableau\_words.exe to start the application.

**PowerShell Method**

1. **Open PowerShell:**
   * Open PowerShell on your computer.
2. **Navigate to the Application Directory:**
   * cd "C:\MyCustumFolder\...\FolderWhereAppIsLocated"
3. **Execute the Application:**
   * .\Sql\_vers\_tableau\_words.exe

### Application Menu

Upon launching the application, you will see a menu with various options:

* **1. Provide Custom File Paths:**
  + Allows you to specify the paths for SQL and Excel files manually.
* **2. Execute with Files:**
  + Starts the conversion process using the specified SQL and Excel files.
* **3. Toggle Debug Mode:**
  + Activates or deactivates debug mode for detailed output.
* **4. Modify the Number of Lines to Clear:**
  + Adjusts the number of lines to clear in the Excel worksheet before writing data.
* **5. Choose Language:**
  + Switches between French and English languages for the application interface.
* **0. Quit:**
  + Exits the application.

### Detailed Steps

1. **Set Custom File Paths:**
   * Enter the full path for the SQL file when prompted.
   * Enter the full path for the Excel file when prompted.
2. **Execute with Files:**
   * Validates the file extensions (.sql for SQL file and .xlsx for Excel file).
   * Initiates the conversion process and writes data to the Excel file.
3. **Toggle Debug Mode:**
   * Enables or disables debug messages to the console for troubleshooting purposes.
4. **Modify the Number of Lines to Clear:**
   * Adjusts how many lines in the Excel worksheet are cleared before new data is written.
5. **Choose Language:**
   * Selects the application language between French and English.
6. **Quit:**
   * Exits the application gracefully.

### Explanation of Tags

#### --%%%C\_TabBalise

The --%%%C\_TabBalise tag serves as a delimiter indicating that the content following it will be placed in the next row of the Excel worksheet generated by the application.

#### --%%%Result\_Attendu:

This is an optional parameters. The --%%%Result\_Attendu: tag **MUST** be followed by either 0 or 1   
(e.g., --%%%Result\_Attendu:0 or --%%%Result\_Attendu:1).

It specifies whether an expected output message should be included in the subsequent column of the Excel worksheet.

* If --%%%Result\_Attendu:0 is present, the value « 0 » will be written in the next column.
* If --%%%Result\_Attendu:1 is present, the value « 1 » will be written in the next column.

If --%%%Result\_Attendu: is not included after --%%%C\_TabBalise, nothing will be written in the next column, as it is an optional parameter.

This tagging system ensures clarity and organization in the output Excel file, aligning each segment of SQL query output with its corresponding expected result status in the generated worksheet. See example below for more details.

### Excel Section

To ensure proper functionality of the application, follow these guidelines for managing the Excel file used by the program:

1. **Excel File Management**:
   * The Excel file provided by the program serves as a backup copy and must always be duplicated to another location on your computer before use. This ensures that the original backup remains intact for reference and recovery purposes.
2. **Starting Point**:
   * The program will always start writing data from the 3rd line of the Excel worksheet. This ensures consistency and prevents accidental overwriting of any existing data in the initial rows of the worksheet.
3. **Copying the Excel File**:
   * Before running the program, if not alraeydy done, manually copy the Excel file provided by the program to a designated location on your computer. You can then use your copy indefenetly.
   * Use this copied version of the Excel file as input for the program's execution. This practice ensures that any modifications made by the program do not affect the original backup file.

Before running the program, follow these steps to manage the Excel file effectively:

1. **Copy the Excel File**:
   * If not already done, manually copy the Excel file provided by the program to a designated location on your computer.
   * This copied version will be used indefinitely as input for the program's execution.
2. **Use Copied Version**:
   * Ensure that you always use the copied version of the Excel file as input for the program.
   * This practice safeguards the original backup file from any modifications made during the program's operation.
   * **Ensure the Excel file is closed before running the application. Otherwise, an error will be thrown.**

By adhering to these practices, you maintain the integrity of your data and ensure that the program operates effectively without compromising the original backup of the Excel file.

## Example Section

#### SQL Query Example:

Below is an anonymized example of the SQL query that demonstrates how the application processes and modifies database entries:

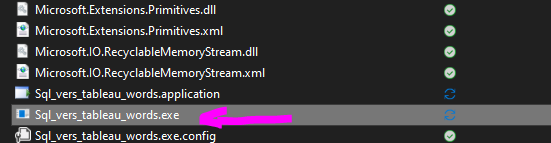
|  |
| --- |
| --%%%C\_TabBalise  SET dateformat ymd  --%%%C\_TabBalise  DECLARE @Item1 int,  @Cost1 int  BEGIN TRANSACTION;  --%%%C\_TabBalise  --%%%Result\_Attendu:1  --\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  --\*\*\*\*\* Modify Item1 \*\*\*\*\*\*  --\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  PRINT 'Modify Item1'  SELECT @Item1 = ID  FROM [RandomSchema].GenericTable  WHERE ItemName = 'Item1';  --%%%C\_TabBalise  --%%%Result\_Attendu:1  UPDATE [RandomSchema].GenericTable  SET LastModified = getdate(),  Description = REPLACE(Description, 'http://www.example-url.com', 'https://www.new-url.com')  WHERE ID = @Item1 AND Description LIKE '%http://www.example-url.com%';  --%%%C\_TabBalise  --%%%Result\_Attendu:0  SELECT \*  FROM [RandomSchema].GenericTable  WHERE ID = @Item1 AND Description LIKE '%http://www.example-url.com%';  --%%%C\_TabBalise  --%%%Result\_Attendu:1  --\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  --\*\*\*\*\* Modify Cost1 \*\*\*\*\*\*  --\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  PRINT 'Modify Cost1'  SELECT @Cost1 = ID  FROM [RandomSchema].GenericTable  WHERE ItemName = 'Cost1';  --%%%C\_TabBalise  --%%%Result\_Attendu:1  UPDATE [RandomSchema].GenericTable  SET LastModified = getdate(),  URL = REPLACE(URL, 'https://www.example-site.com', 'https://www.new-example-site.com')  WHERE ID = @Cost1 AND URL LIKE '%https://www.example-site.com%';  --%%%C\_TabBalise  --%%%Result\_Attendu:0  SELECT \*  FROM [RandomSchema].GenericTable  WHERE ID = @Cost1 AND URL LIKE '%https://www.example-site.com%';  --%%%C\_TabBalise ROLLBACK TRANSACTION; |

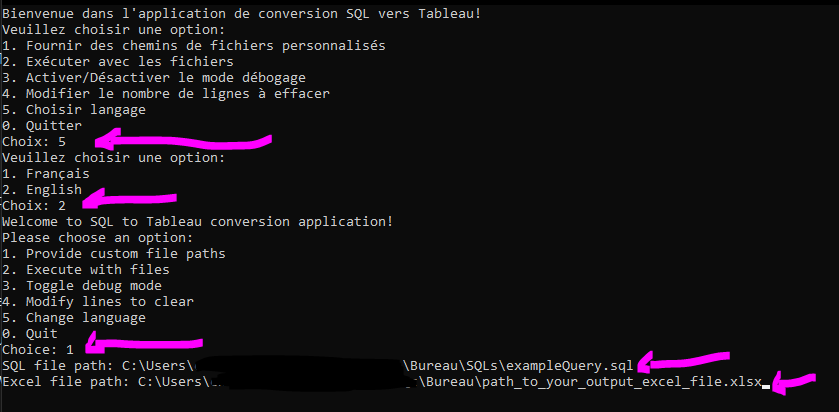
#### Excel Provided Sample

|  |  |
| --- | --- |
| Requête SQL | **Nb. rangées affectées** |
| *Faire un SELECT des champs touchés avant et après la mise à jour* |  |
| The program will start here and override this text. |  |
|  |  |
|  |  |
|  |  |
|  |  |

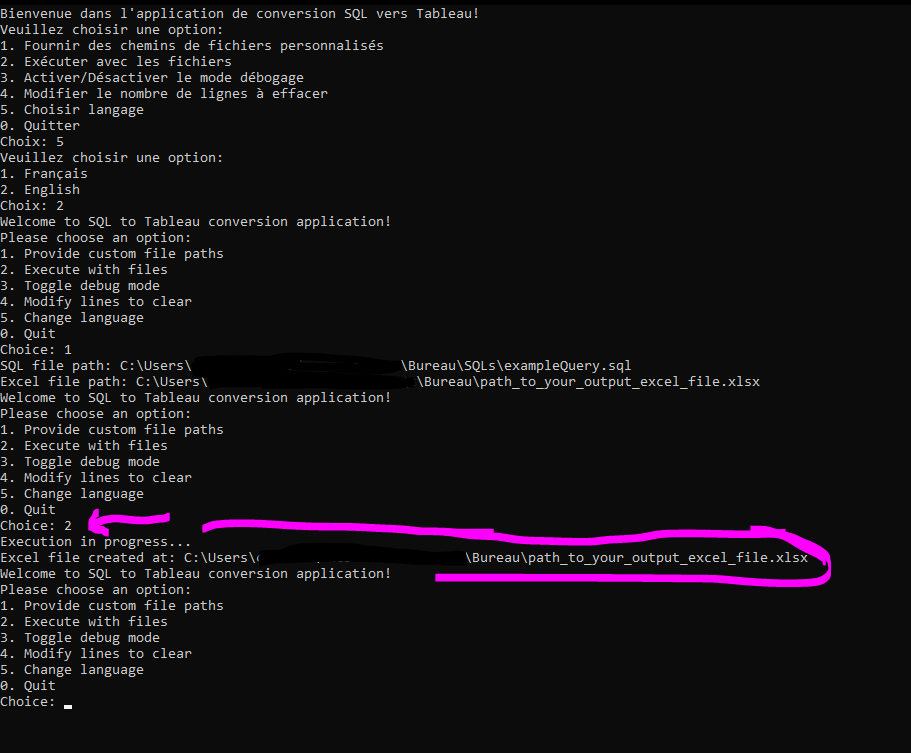
#### Live Example Protocol:

1. **Initial Setup:**
   * If it's the first time using the program, copy the Excel file provided to your desired location. Use this copied version for all future operations.
2. **Prepare SQL Query:**
   * Identify the path to the actual .sql query file that you want to transform.
   * Ensure that the --%%%C\_TabBalise and --%%%Result\_Attendu: tags are present in the SQL file where necessary.
3. **Run the Application:**
   * Open the application Sql\_vers\_tableau\_words.exe.
   * Select option 1.
   * Enter the full path, including the file name, of your SQL query file when prompted.
   * Enter the path and the file name of your copied Excel file when prompted.





1. **Execute the Program:**
   * Select option 2 to execute the program.
   * You can choose to leave the application running or close it after execution.



1. **Review Output:**
   * Open your Excel file to review the output results.
   * Verify if the output meets your expectations.



Expected Result from the Example:

|  |  |
| --- | --- |
| Requête SQL | **Nb. rangées affectées** |
| *Faire un SELECT des champs touchés avant et après la mise à jour* |  |
| SET dateformat ymd |  |
| DECLARE @Item1 int,  @Cost1 int    BEGIN TRANSACTION; |  |
| --\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  --\*\*\*\*\* Modify Item1 \*\*\*\*\*\*  --\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  PRINT 'Modify Item1'    SELECT @Item1 = ID   FROM [RandomSchema].GenericTable   WHERE ItemName = 'Item1'; | 1 |
| UPDATE [RandomSchema].GenericTable   SET LastModified = getdate(),   Description = REPLACE(Description, 'http://www.example-url.com', 'https://www.new-url.com')  WHERE ID = @Item1 AND Description LIKE '%http://www.example-url.com%'; | 1 |
| SELECT \*   FROM [RandomSchema].GenericTable   WHERE ID = @Item1 AND Description LIKE '%http://www.example-url.com%'; | 0 |
| --\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  --\*\*\*\*\* Modify Cost1 \*\*\*\*\*\*  --\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  PRINT 'Modify Cost1'    SELECT @Cost1 = ID   FROM [RandomSchema].GenericTable   WHERE ItemName = 'Cost1'; | 1 |
| UPDATE [RandomSchema].GenericTable   SET LastModified = getdate(),   URL = REPLACE(URL, 'https://www.example-site.com', 'https://www.new-example-site.com')  WHERE ID = @Cost1 AND URL LIKE '%https://www.example-site.com%'; | 1 |
| SELECT \*   FROM [RandomSchema].GenericTable   WHERE ID = @Cost1 AND URL LIKE '%https://www.example-site.com%'; | 0 |
| ROLLBACK TRANSACTION; |  |

1. **Copy Array to Word File:**
   * Once satisfied with the output in Excel, copy the array of cells to your Word file for documentation purposes.

**Pro Tips:**

* The paths to the SQL file and Excel file are stored in memory for the lifetime of the application session. Once entered, you can execute option 2 without re-entering the paths.
* It is recommended to keep the application open until you are confident with the output results in Excel, allowing for iterative adjustments if necessary.
* Ensure the Excel file is closed before running the application. Otherwise, an error will be thrown.
* It is possible to change the number of lines to clear in the Excel table with option 4 from the menu.

## Conclusion

This document has provided comprehensive instructions on using the SQL to Tableau Conversion Application. For further assistance or troubleshooting, refer to the error messages displayed or consult the code.