

# Improving the performance of SQL Data Compare

SQL Data Compare can take a long time to compare and deploy data sources. The speed of the comparison depends mostly on disk write speed, processor speed, memory, and the size of the databases.

When you run a comparison, SQL Data Compare retrieves the data from the two data sources and copies it to a temporary location on your local machine. If the SQL Servers are remote, the data is copied across the network to your local machine.

Although there is no limit on the size of the data sources that you can compare, the amount of available disk space restricts the amount of data that can be compared. You need roughly twice the size of the database for the comparison and up to four times the size of the database to generate the deployment SQL.

To speed up the comparison, and avoid running out of disk space, you can:

- [change the location of the temporary files](#)
- [filter the comparison at table-level, column-level, or row-level](#)
- [review the options for comparison behavior](#)

## Changing the location of the temporary files

SQL Data Compare uses temporary files when it compares the databases. To avoid running out of disk space, you can change the location of these files.

The location of the temporary files is defined by the **RGTEMP** environment variable, or the **TMP** variable if **RGTEMP** does not exist (see your Windows documentation for information about environment variables).

You are not recommended to edit the **TMP** variable, as this will affect all programs that use the variable. Instead, you can create a new environment variable called **RGTEMP**, and specify the required location, such as a different hard disk with more free space. The **RGTEMP** variable affects only Red Gate programs.



You may need to log out of Windows for the change to take effect.

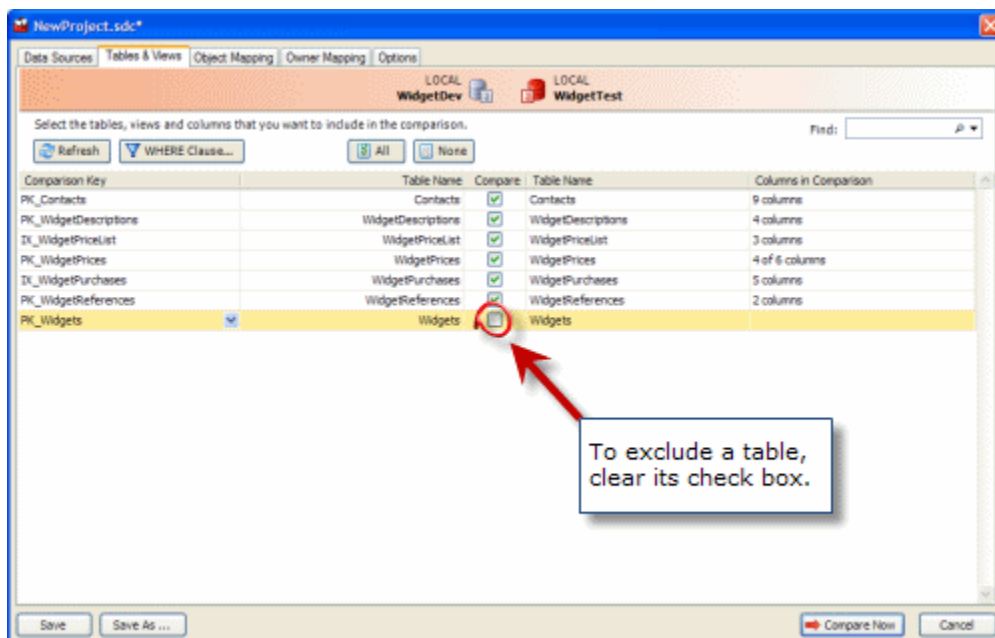
## Filtering the comparison

By default, SQL Data Compare selects all tables, all columns, and all rows for comparison. This means all the data that differs in both data sources is stored temporarily. By filtering the data, you compare only the data you are really interested in, and the size of the temporary files is reduced.

You filter data using the **Tables & Views** tab on the **Project Configuration** dialog box. The filtering techniques described here are useful when your tables contain a large amount of data that differs:

### Table filtering

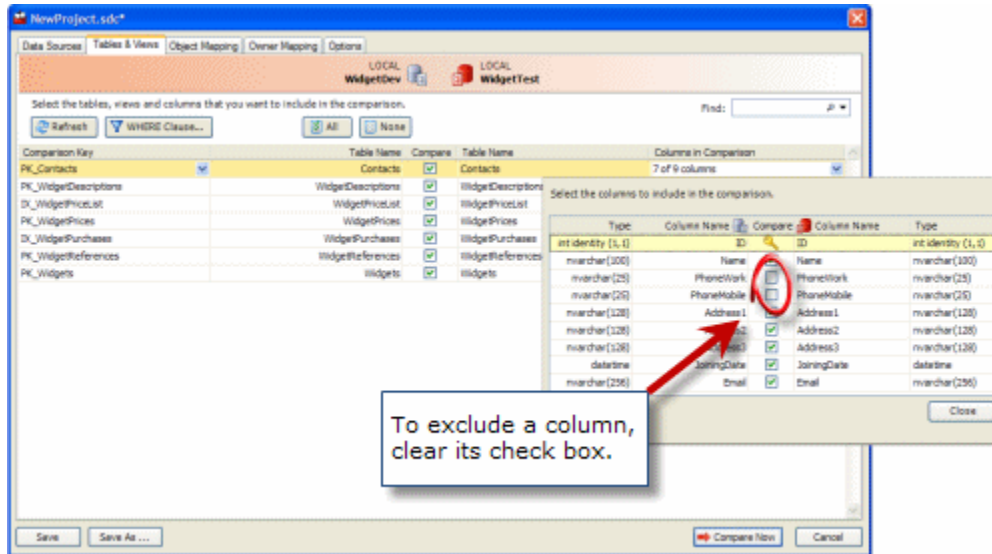
To exclude all the data in a table, on the **Tables & Views** tab of the **Project Configuration** dialog box, clear its check box:



The table will not be compared.

## Column filtering (vertical filtering)

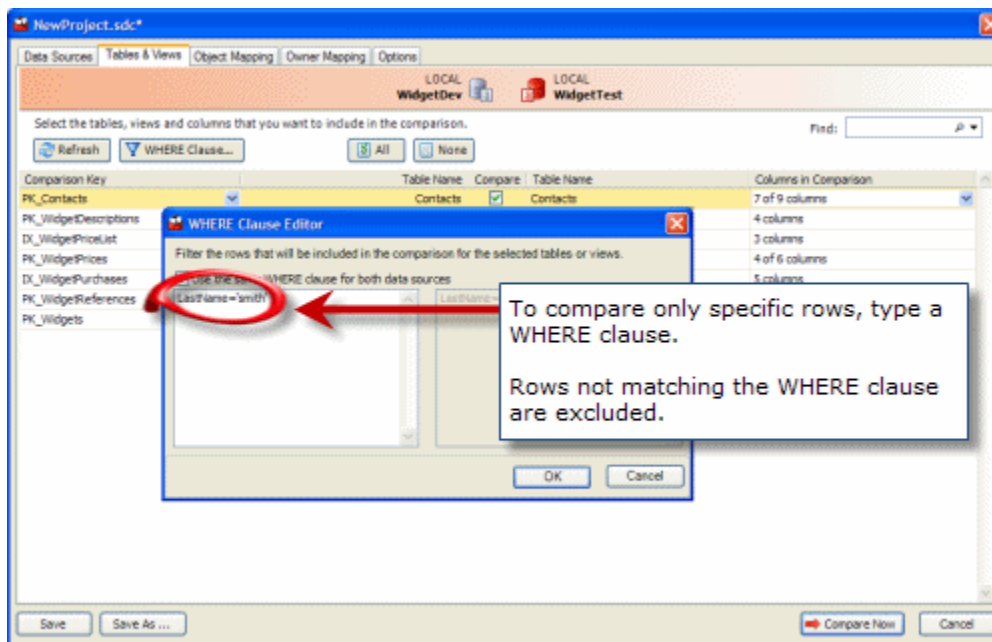
To exclude all the data in particular columns, click in the **Columns in Comparison** box for the associated table, and then clear the check boxes for the required columns:



Only the selected columns will be compared.

## Row filtering (horizontal filtering)

To exclude specific rows, click **WHERE Clause**, and then type a T-SQL WHERE clause in the box:



You can apply the same WHERE clause to multiple tables, by highlighting the required tables before opening the **WHERE Clause Editor**.

You can apply a different WHERE clause for each data source by clearing the **Use the same WHERE clause for both data sources** check box.



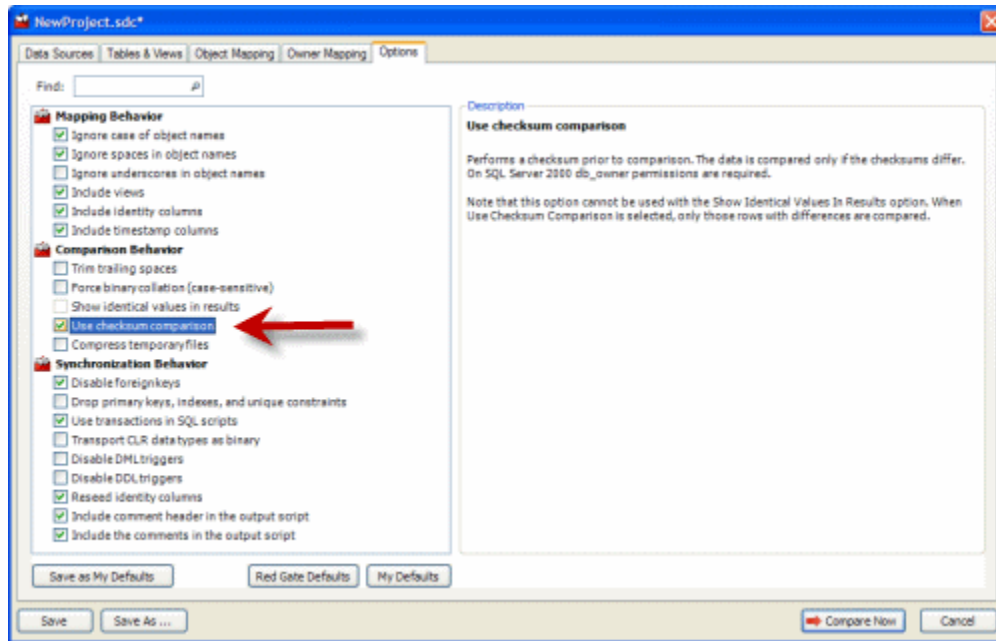
You can only filter rows if the data source is a database. WHERE clauses are not available for backups and scripts folders.

## Reviewing the options for comparison behavior

By default, SQL Data Compare includes identical values in the comparison results. If your data sources always contain similar data, on the **Options** tab of the **Project Configuration** dialog box, clear the **Show identical values in results** check box.

When you do this, SQL Data Compare does not store data that is identical, reducing the disk space that is required.

If you are comparing tables with large amounts of data that changes infrequently, select the **Use checksum comparison** check box. You must clear the **Show identical values in results** check box before you do this. If the checksums are equal, SQL Data Compare does not compare that table, and its data is not stored, reducing the disk space that is required.



If your table contains large data types such as text or images, these columns are not included in the checksum, and you are not recommended to use the **Use checksum comparison** option.