## Tables or views that could not be compared

Objects with rows that can not be matched automatically are shown in the **Tables or views that could not be compared** group. To match rows in the two data sources, SQL Data Compare requires:

- · a comparison key for each table or view
- mappings between objects in the data sources

In some cases, SQL Data Compare is unable to map objects and select comparison keys automatically. For example, if there are substantial schema differences between the data sources. This article provides information on manually setting comparison keys.



For more detailed information on setting mappings, see: Mapping data sources.

## What is a comparison key?

When comparing data sources, SQL Data Compare looks for a matching primary key or other unique identifier in each data source to use as the *compariso n* key. This enables matching rows to be identified, and their differences to be compared.

For example, the databases WidgetSales and WidgetDeploy both contain the table [dbo].[WidgetPrices]:



Since rows can be inserted and deleted, we can not be certain that the third row in *WidgetSales* is the same as the third row in *WidgetDeploy*. Simply comparing rows in the order in which they appear in the table can result in a meaningless comparison. Similarly, rows can not be matched based on their *P rice* values - more than one widget can have the same price.

In [dbo].[WidgetPrices], RecordID is a primary key. No two widgets will have the same RecordID, and the rows are uniquely identified. Two matching RecordID values in WidgetSales and WidgetDeploy therefore represent the same piece of real world data.

Where there is no matching primary key, or other unique identifier you must set an appropriate comparison key.

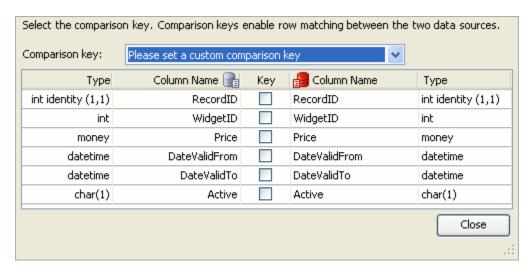
For example, if you know that two widgets with the same name are never added to the database on the same day, you can select the two columns *WidgetN* ame and *DateValidFrom* to form the comparison key. The rows in the tables can now be matched.

## Setting comparison keys

SQL Data Compare automatically selects a comparison key, if your:

- · tables contain a matching primary key, unique index, or unique constraint
- · views contain a matching unique, clustered index

If SQL Data Compare is unable to identify a suitable comparison key for a table or view, *Not Set* is shown in the **Comparison Key** box, under **Tables & Views** on the Project Configuration dialog. You set the comparison key by clicking the appropriate **Comparison Key** box. A dialog box is displayed, for example:



Select the columns in the table or view that will comprise the comparison key by using the check boxes.



- A comparison key cannot include columns whose data type is image, ntext, nvarchar(max), sql\_variant, text, varbinary(max), varchar (max), or xml.
- You cannot specify custom comparison keys if you are using a backup as a data source; however, you can select an alternative unique index or unique constraint.



For more information on setting up your comparison, see:

- Setting data sources
- Selecting tables and views