

Known issues

IDERA strives to ensure our products provide quality solutions for your SQL Server needs. The following known issues are described in this section. ***If you need further assistance with any issue***, please contact [IDERA Support](#).

Known issues in 7.1.x

Removes Azure v12 support for assemblies and CLR objects

IDERA SQL Comparison Toolset 7.1 removes Azure v12 support for assemblies and CLR objects including procedures, functions, triggers, types etc.

Two different databases appear equal to some users

IDERA SQL Comparison Toolset includes a number of options that can cause the following issues:

- IDERA SQL Schema Compare index comparison options may “hide” differences between two different indexes and make them appear equal. Currently, IDERA SQL Schema Compare supports seven index options available by clicking **Comparison Options > Index Options**. If an option is unchecked, the property affected by it is not compared.
- When check constraints have the same name on different tables, they may not be identified as different when using IDERA SQL Schema Compare. For example, the check constraints would look as follows:

LOC_TX.g_1835_ck

LOC_CA.g_1835_ck

Note the following operation of IDERA SQL Schema Compare:

1. SQL Server supports two types of check constraints: column-constraints and table-constraints. A column constraint is applied to one column only while a table constraint affects multiple columns.
2. Table-constraints have a table-scope, i.e. the constraints on the left database are compared with constraints of the same table on the right database.
3. Column constraints have a table and column scope, so constraints on the left database are compared with the constraints of the same table, under the same column, on the right database.

Using the example, the constraints of the LOC_TX table on the left database are compared with those of the LOC_TX table on the right database. Constraints of LOC_TX and LOC_CA are not mixed together.

IDERA SQL Schema Compare can ignore the name of a check constraint if the option **Ignore check constraint name** is checked. This option affects only column constraints and it is located under **Comparison Options > Constraint Options**.

Comparison grid appears to not work properly if a specific option is checked

To some users, it appears as if the **Select All / Clear All** feature in the comparison grid does not work properly. An object still can appear in the synchronization script even after being unchecked in the comparison grid. This behavior is by design and has to do with the dependencies in the database. To ensure that an object is excluded permanently, the user must uncheck the comparison option **Include dependent objects**.

Apparent error during comparison

Some users may see an error occurring during object pairing before the schema comparison. This issue usually results from a database created with case-sensitive collation and containing objects with the same name but different case, such as **Employee** and **employee**. Note that while this error is triggered, IDERA SQL Schema Compare does continue to compare the databases. However, from the set of objects with the same name, only one of them is compared and the remainder is ignored.

Total compare from data to schema IDERA [SQL Comparison Toolset](#). [Learn more](#) >>

IDERA Website	Products	Purchase	Support	Community	About Us	Resources	Legal
-------------------------------	--------------------------	--------------------------	-------------------------	---------------------------	--------------------------	---------------------------	-----------------------