

SQL Schema Compare Selecting Database Objects

By default, all the objects that were found to be different during the comparison of the two databases are marked for inclusion in the synchronization script. However, you can choose to exclude or include certain objects from the comparison by checking or clearing the **Synchronize** checkbox next to each object. Selecting or clearing a parent object also affects the dependent objects. For example, if the checkbox for a table is cleared, then all of its primary keys, foreign keys, indexes etc. are also cleared.

Cleared but included

You may be surprised to see included in the synchronization script an object that you explicitly excluded. Why are those objects included? To answer this you need to look under the comparison / synchronization options and see if the option to **Include Dependent Objects** is **ON**. When that option is **ON**, an included object will cause all the objects on which it depends to also be included in the synchronization script. For example, if you have a table T1 that contains foreign keys coming from tables T2 and T3 then, even though you may have cleared tables T2 and T3 if T1 is checked then both T2 and T3 will be included. When the **Include Dependent Objects** option is **OFF**, then your object selection is respected, however, in this case you should be aware that the resulting synchronization script might not execute successfully.

The screenshot displays the IDERA SQL Schema Compare tool interface. The top ribbon includes tabs for Home, Review Comparison, and Synchronize. The Synchronize tab is active, showing options like Refresh Result, Select All Objects, Clear Selection, Entity Filters, Filegroup Mappings, Command Line Config, and Manage Comparison. The main workspace shows a comparison between SQL 2016 AdventureWorks and SQL 2019 AdventureWorks. The 'Comparison Result' table lists various tables with their 'Synchronize' checkboxes. The 'Address' table checkbox is highlighted with a red box. The 'Schema Differences' section at the bottom shows the SQL scripts for the compared tables, with the 'Address' table script highlighted.

Synchronize	Entity Type	Schema	Name	Status
<input type="checkbox"/>	Table	HumanResources	EmployeeDepartmentHistory	=
<input type="checkbox"/>	Table	HumanResources	EmployeePayHistory	=
<input type="checkbox"/>	Table	HumanResources	JobCandidate	=
<input type="checkbox"/>	Table	HumanResources	Shift	=
<input type="checkbox"/>	Table	Person	Address	+
<input type="checkbox"/>	Table	Person	AddressType	+
<input checked="" type="checkbox"/>	Table	Person	Contact	+
<input type="checkbox"/>	Table	Person	ContactType	+
<input type="checkbox"/>	Table	Person	CountryRegion	=
<input type="checkbox"/>	Table	Person	StateProvince	=
<input checked="" type="checkbox"/>	Table	Production	BillOfMaterials	+

Schema Differences

Left Database Script

```
--Main Script - Different
CREATE TABLE [Person].[Address]
(
    [AddressID] [int] IDENTITY(1,1) NOT FOR REPLICATION,
    [AddressLine1] [nvarchar](60) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [AddressLine2] [nvarchar](60) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [City] [nvarchar](30) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [StateProvinceID] [int] NOT NULL,
    [PostalCode] [nvarchar](15) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [SpatialLocation] [geometry] NOT NULL
)
```

Right Database Script

```
--Main Script - Different
CREATE TABLE [Person].[Address]
(
    [AddressID] [int] IDENTITY(1,1) NOT FOR REPLICATION,
    [AddressLine1] [nvarchar](100) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [AddressLine2] [nvarchar](100) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [City] [nvarchar](30) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [StateProvinceID] [int] NOT NULL,
    [PostalCode] [nvarchar](15) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [SpatialLocation] [geometry] NOT NULL
)
```