Steps to Move a Table with Primary Key from Primary FG to a Different Filegroup in SQL Server

https://www.sqlmatters.com/Articles/Moving%20a%20Table%20to%20a%20Different%20Filegroup.aspx

Introduction

- By default, a database has a single filegroup called PRIMARY and all tables are stored in this filegroup. However, it's possible to create other filegroups and store tables in these other filegroups.
- One of the most common reasons for doing this is if you have some high performance disks and want to store certain tables on those faster disks.
- This is especially useful for data warehouses and other large databases where you might have some data that is accessed more frequently and where response times are particularly important.
- This article shows how to create another filegroup and move existing tables into the new filegroup (if you are creating new tables then you can just create them on the new filegroup). I have used the AdventureWorks2008R2 demo database which can be downloaded from the CodePlex website.

Identifying which Filegroup a Table is currently on

To check which filegroup a table is currently on, just run the following query:

```
USE DBName –Put your required DB Name here.
go
SELECT    tbl.name AS [Table Name],
       CASE WHEN dsidx.type='FG' THEN dsidx.name ELSE '(Partitioned)' END AS [File Group]
FROM      sys.tables AS tbl
JOIN      sys.indexes AS idx
ON        idx.object_id = tbl.object_id
AND       idx.index_id <= 1
LEFT JOIN sys.data_spaces AS dsidx
ON        dsidx.data_space_id = idx.data_space_id
ORDER BY  [File Group], [Table Name]
```

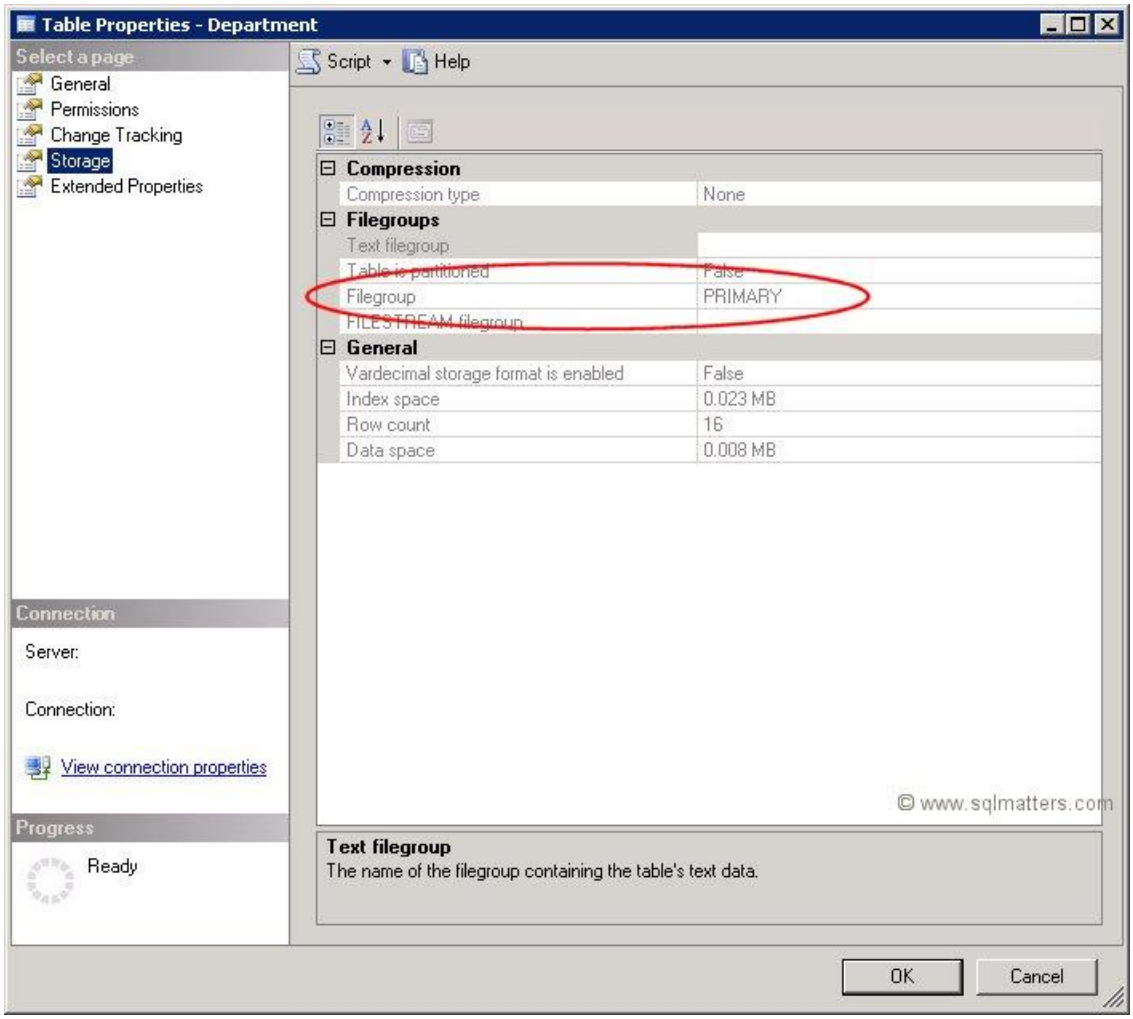
This query lists all tables and their associated filegroup, which will be 'PRIMARY' if you've just used defaults. If the table is partitioned, and therefore potentially stored on multiple filegroups, it will be listed as '(partitioned)'.

This is the result the author got for the first few tables in **AdventureWorks2008R2** database:

Results | Messages

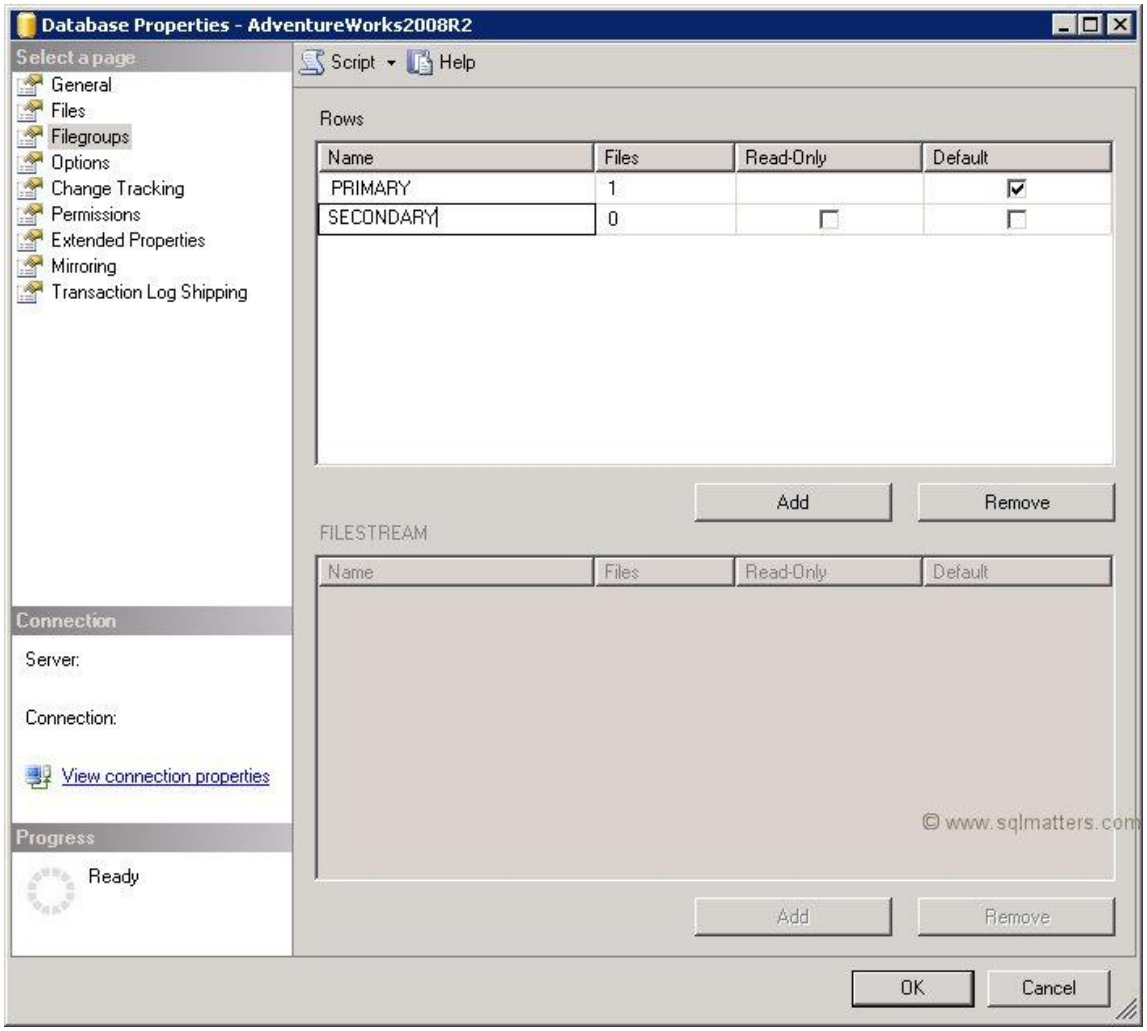
	Table Name	File Group
1	Address	PRIMARY
2	AddressType	PRIMARY
3	AWBuildVersion	PRIMARY
4	BillOfMaterials	PRIMARY
5	BusinessEntity	PRIMARY
6	BusinessEntityAddress	PRIMARY
7	BusinessEntityContact	PRIMARY
8	captured_columns	PRIMARY

You can also get this information for a specific table by looking at the Storage page of the Properties window of the table in SQL Server Management Studio (to see the Properties window right click on the table in Object Explorer and select 'Properties') :

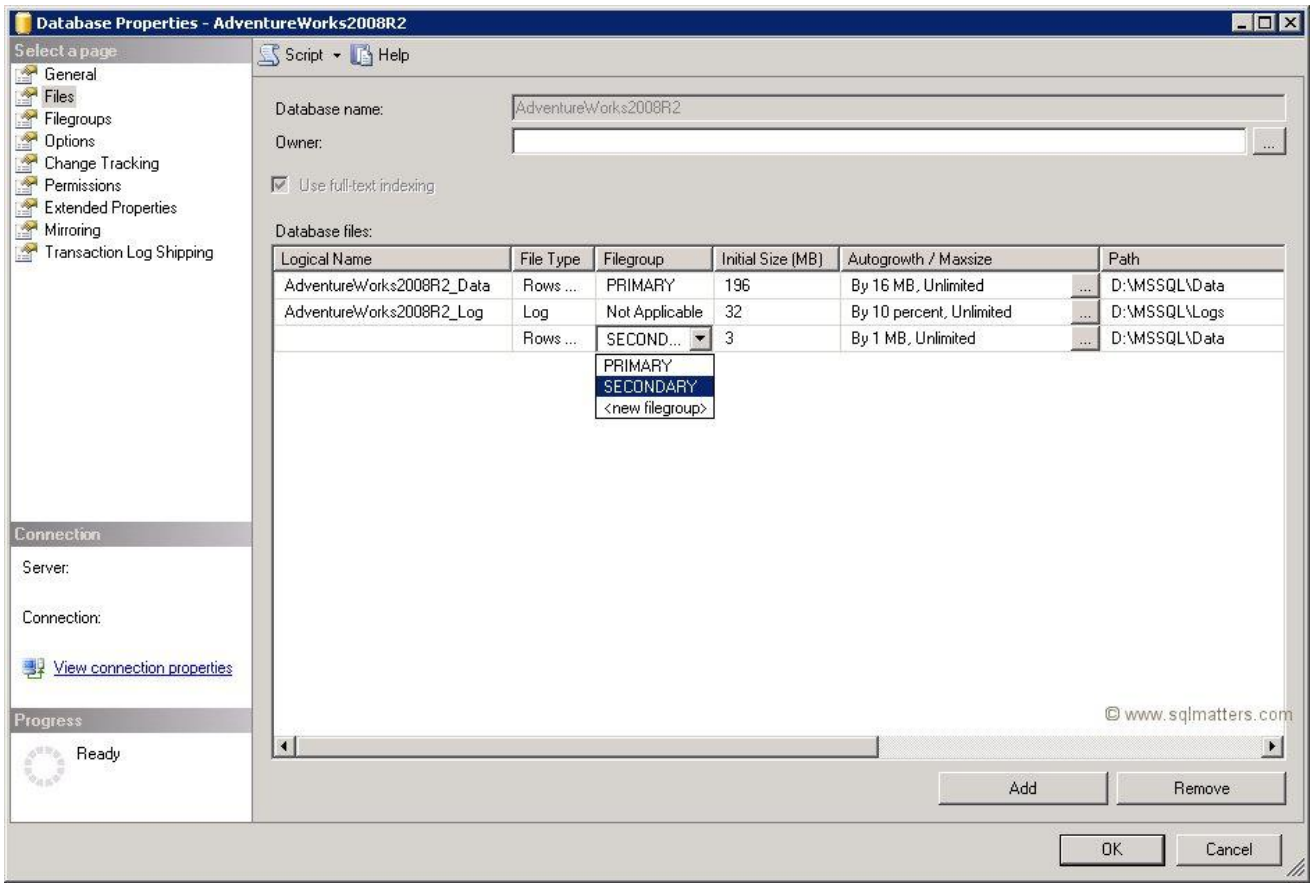


Creating a New Filegroup

- If the filegroup you want to move the table to doesn't already exist, then it will need creating.
- To do this right click on the database within Object Explorer and select Properties.
- In the properties window select the 'Filegroups' item and then click the 'Add' button.
- The name of the new filegroup can then be typed into the text box as shown below (Author has chosen to call it SECONDARY) :



- The next stage is to go to the ‘Files’ page in the same Properties window and add a file to the filegroup (a filegroup always contains one or more files) using the ‘Add’ button.
- A logical name and sizing information can also be set up as shown below (make sure you select the new filegroup):



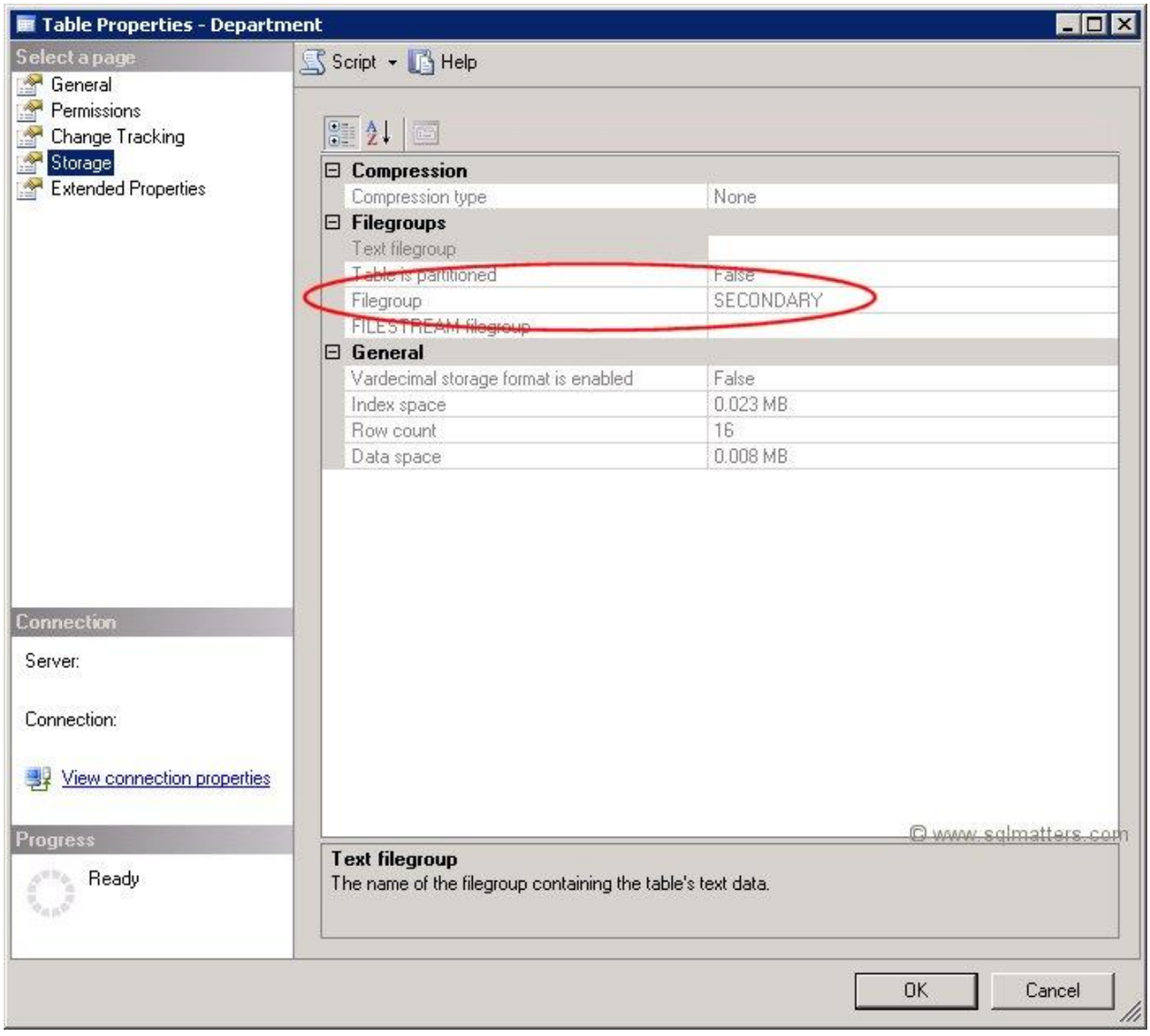
Click OK to create the new filegroup and file.

Moving an Existing Table to the new Filegroup

To move a table to a different filegroup involves moving the table's clustered index to the new filegroup. While this may seem strange at first this is not that surprising when you remember that the leaf level of the clustered index actually contains the table data. Moving the clustered index can be done in a single statement using the DROP_EXISTING clause as follows (using one of the AdventureWorks2008R2 tables as an example) :

```
USE DBNAME

go

CREATE UNIQUE CLUSTERED INDEX PK_Department_DepartmentID

ON HumanResources.Department (DepartmentID)

WITH (DROP_EXISTING=ON,ONLINE=ON)  ON SECONDARY
```

This recreates the same index but on the SECONDARY filegroup, this can be checked by looking at the table's properties again (compare this to the first screenshot above):



This article has shown how to create a new filegroup and move an existing table into this filegroup (or any other filegroup).