Partition

--write a procedure that recieves tableName, columnName, schemaName

--and dynamicly creates a partition on the table and column passed in

--create database

use master

go

create database partitionDB

--create filegroups

use master

go

alter database partitionDB add filegroup fg1

go

alter database partitionDB add filegroup fg2

go

alter database partitionDB add filegroup fg3

go

--create files for the filegroups

ALTER DATABASE partitionDB ADD FILE

( NAME = N'df1', FILENAME = N'D:\CourseMaterials\Company\df1.ndf' ,

SIZE = 8192KB , FILEGROWTH = 65536KB ) TO FILEGROUP [fg1]

GO

ALTER DATABASE [partitionDB] ADD FILE

( NAME = N'df2', FILENAME = N'D:\CourseMaterials\Company\df2.ndf' ,

SIZE = 8192KB , FILEGROWTH = 65536KB ) TO FILEGROUP [fg2]

GO

ALTER DATABASE [partitionDB] ADD FILE

( NAME = N'df3', FILENAME = N'D:\CourseMaterials\Company\df3.ndf' ,

SIZE = 8192KB , FILEGROWTH = 65536KB ) TO FILEGROUP [fg3]

GO

use [partitionDB]

go

--create new procedure that recieves recieves tableName, columnName, schemaName

--and dynamicly creates a partition on the table and column passed in

alter procedure createPartition (@tableName varchar(25), @columnName varchar(25),

@schemeName varchar(25))

AS

BEGIN

--declare @columnType varchar(25) = 'datetime'

--select @columnType = DATA\_TYPE

--FROM INFORMATION\_SCHEMA.COLUMNS

--WHERE TABLE\_NAME = 'newPartitionTable'

--and COLUMN\_NAME = @columnName

DECLARE @DynamicSQL nvarchar(1000);

--partition logic

--drop partition function partFunc\_year

SET @DynamicSQL ='CREATE PARTITION FUNCTION partFunc\_year ( datetime )

AS RANGE LEFT

FOR VALUES ( ''01-01-2017'', ''01-01-2018'', ''01-01-2019'' ) '

EXEC(@DynamicSQL);

set @DynamicSQL =''

--select $partition.partFunc\_year('01-01-2018')

--partition mapping to filegroup

--drop partition scheme partScheme\_year

SET @DynamicSQL ='CREATE PARTITION SCHEME '+ @schemeName +

' AS PARTITION partFunc\_year

TO (fg1, fg2, fg3, fg1)'

EXEC(@DynamicSQL);

set @DynamicSQL =''

--create table on partition schema

--drop table newPartitionTable

SET @DynamicSQL = 'create table ' + @tableName + '

(id int IDENTITY(1,1) NOT NULL, '+ @columnName+ ' datetime not null)

ON '+@schemeName+ '('+@columnName+')';

EXEC(@DynamicSQL);

set @DynamicSQL =''

--create table @schemaName.@tableName

--(id int IDENTITY(1,1) NOT NULL, @columnName datetime not null)

--ON partSchema\_year(@columnName)

end

--call procedure

createPartition 'newPartitionTable', hireDate, 'partScheme\_year'

--add data to check it works

insert into newPartitionTable

select dateadd(dd , checksum(newid()) % datediff (dd , '20170101' , '20181220' )

, '20190101')

from sys.messages

-- Check the the table allocation status

SELECT \*

FROM sys.Dm\_db\_index\_physical\_stats(Db\_id(),

OBJECT\_ID('dbo.newPartitionTable' ), NULL, NULL, 'DETAILED')

use master

go

drop database partitionDB

select \*,DATA\_TYPE

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'newPartitionTable'

and COLUMN\_NAME = 'hiredate'

SELECT

OBJECT\_NAME(SI.object\_id) AS PartitionedTable

, DS.name AS PartitionScheme

FROM sys.indexes AS SI

JOIN sys.data\_spaces AS DS

ON DS.data\_space\_id = SI.data\_space\_id

WHERE DS.type = 'PS'

--AND OBJECTPROPERTYEX(SI.object\_id, 'BaseType') = 'U'

AND SI.index\_id IN(0,1);

select \* from sys.data\_spaces

select \* from sys.partitions