

```
/* Задание
Секционирование таблицы
Выбираем в своем проекте таблицу-кандидат для секционирования и добавляем
    партиционирование.
Если в проекте нет такой таблицы, то делаем анализ базы данных из WWI, выбираем
    таблицу и делаем ее секционирование,
с переносом данных по секциям (партициям) - исходя из того, что таблица большая,
    пишем скрипты миграции в секционированную таблицу */

use WideWorldImporters;

-- релизуем секционирование таблицы сумм заказов по городам

--создаем функцию партиционирования
CREATE PARTITION FUNCTION [fnAmounts](decimal(15,2)) AS RANGE RIGHT FOR VALUES
(5000, 10000, 15000, 20000, 25000, 30000, 35000);

GO

-- и схему
CREATE PARTITION SCHEME [schAmountsPartition] AS PARTITION [fnAmounts]
ALL TO ([PRIMARY]);
GO

-- и таблицу для секционирования

drop table if exists Sales.OrderAmountsByCities;

Create table Sales.OrderAmountsByCities(
    ItemID int not null IDENTITY(1,1),
    DeliveryCityID int not null,
    DateMonth int not null,
    DateYear int not null,
    OrderAmount decimal(15,2) not null)
GO

-- и кластерный индекс
ALTER TABLE Sales.Sales ADD CONSTRAINT PK_Sales_OrderAmountsByCities
PRIMARY KEY CLUSTERED (OrderAmount, DateYear, ItemID)
ON [schAmountsPartition](OrderAmount);

-- загрузим результат запроса в таблицу
EXEC sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
EXEC sp_configure 'xp_cmdshell', 1;
GO
RECONFIGURE;
GO

exec master..xp_cmdshell 'bcp "SELECT c.DeliveryCityID, DATEPART(MONTH,
    o.OrderDate) AS DateMonth, DATEPART(YEAR,o.OrderDate) AS DateYear, SUM
```

```
(ol.UnitPrice * ol.Quantity) as OrderAmount FROM WideWorldImporters.Sales.Orders
as o JOIN WideWorldImporters.Sales.Customers as c ON o.CustomerID = c.CustomerID
JOIN WideWorldImporters.Sales.OrderLines as ol ON o.OrderID = ol.OrderLineID
GROUP BY c.DeliveryCityID, DATEPART(MONTH, o.OrderDate), DATEPART
(YEAR,o.OrderDate) HAVING SUM(ol.UnitPrice * ol.Quantity) > 0" queryout "D:\doc
\edu\mssql\30 partition\OrderAmounts.txt" -c -C 65001 -T -t "\t"

BULK INSERT [Sales].[OrderAmountsByCities]
FROM 'D:\doc\edu\mssql\30 partition\OrderAmounts.txt'
WITH (FORMATFILE = 'D:\doc\edu\mssql\30 partition\OrderAmountsFmt.fmt'); -- файл
форматирования для автоматического заполнения идентификатора при загрузке
GO

-- проверим количество элементов в каждой секции
SELECT $PARTITION.fnAmounts(OrderAmount) AS Partition,
COUNT(*) AS [COUNT], MIN(OrderAmount),MAX(OrderAmount)
FROM Sales.OrderAmountsByCities
GROUP BY $PARTITION.[fnAmounts](OrderAmount)
ORDER BY Partition ;

-- перегруппируем секции для более равномерного распределения
ALTER PARTITION SCHEME [schAmountsPartition]
NEXT USED [PRIMARY];
Alter Partition Function fnAmounts() SPLIT RANGE ('1000');

ALTER PARTITION SCHEME [schAmountsPartition]
NEXT USED [PRIMARY];
Alter Partition Function fnAmounts() SPLIT RANGE ('2000');

ALTER PARTITION SCHEME [schAmountsPartition]
NEXT USED [PRIMARY];
Alter Partition Function fnAmounts() SPLIT RANGE ('3000');

ALTER PARTITION SCHEME [schAmountsPartition]
NEXT USED [PRIMARY];
Alter Partition Function fnAmounts() SPLIT RANGE ('4000');

Alter Partition Function fnAmounts() MERGE RANGE ('15000');
Alter Partition Function fnAmounts() MERGE RANGE ('20000');
Alter Partition Function fnAmounts() MERGE RANGE ('25000');
Alter Partition Function fnAmounts() MERGE RANGE ('30000');
Alter Partition Function fnAmounts() MERGE RANGE ('35000');

-- запрос по сумме заказа использует кластерный индекс для поиска в двух секциях
(см. partition.sqlplan)
Select *
FROM Sales.OrderAmountsByCities as o
WHERE o.OrderAmount < 1500 AND o.DateYear = 2015
```