

Implementing In-Memory SQL Database Objects

Lab 3 – Creating Memory Optimized Tables

Overview

Due to the improved performance and reduced disk space that columnstore indexes provide, you have been tasked with taking the FactInternetSales table from disk and into memory.

Before starting this lab, you should view **Module 3 – Implementing Memory Optimized Tables** in the course *Implementing In-Memory SQL Database Objects*. Then, if you have not already done so, follow the instructions in the **Getting Started** document for this course to set up the lab environment.

If you find some of the challenges difficult, don't worry – you can find suggested solutions for all of the challenges in the **Lab Solution** folder for this module.

What You'll Need

To complete the labs, you will need the following:

- A SQL Server instance with the AdventureWorksDW sample database. Review the Getting Started document for information about how to provision this.
- The lab files for this course

Challenge 1: Create a Memory Optimized Columnstore Table

In this exercise, you will create a memory optimized version of the FactInternetSales disk based table, using the Memory Optimization Advisor. Ensure that you have successfully completed Lab 2 and created the **CCI_FactInternetSales** index on the **FactInternetSales** table.

Use the Memory Optimization Advisor

1. In SQL Server Management Studio, run the **Memory Optimization Advisor** on the **FactInternetSales** table.
2. Note that there are several issues that need to be resolved before the Memory Optimization Advisor can automatically convert the table.

Enable the Memory Optimization Advisor to Create a Memory Optimized FactInternetSales Table

1. Navigate to the **Setup** folder, click **Drop Columnstore Indexes on FactInternetSales.sql**, and then click **Open**.
2. Click **Execute** to drop the existing columnstore index and foreign keys.
3. In Object Explorer, right-click **dbo.FactInternetSales**, and then click **Memory Optimization Advisor**.
4. In the **Table Memory Optimization Advisor** window, on the **Introduction** page, click **Next**.
5. On the **Migration validation** page, click **Next**.
6. Click **Next**.
7. On the **Migration options** page, select **Also copy table data to the new memory optimized table**, and then click **Next**.
8. On the **Index creation** page scroll down and in the column list, select **SalesOrderNumber** and **SalesOrderLineNumber**, and then click **Next**.
9. On the **Summary** page, click **Script**.
10. Inspect the generated Transact-SQL. Note the code to create the memory optimized filegroup and the renaming of the existing table. The Memory Optimization Advisor won't suggest columnstore indexes as they are not applicable in all situations. Therefore, you have to add these manually.
11. Locate the following lines of Transact-SQL:

```
)WITH ( BUCKET_COUNT = 2097152)  
)WITH ( MEMORY_OPTIMIZED = ON ,  
DURABILITY = SCHEMA_AND_DATA )
```
12. Add the following Transact-SQL between the two WITH statements:

```
,INDEX  
CCI_OnlineFactInternetSales  
CLUSTERED COLUMNSTORE
```

13. On the toolbar, click **Execute**.

Examine the Performance of the Memory Optimized Table

1. In the **Setup** folder, open the **Query FactProductInventory.sql** script file.
2. Configure SQL Server Management Studio to include the actual execution plan.
3. Execute the script against the **AdventureWorksDW** database, and then review the disk space used and execution plan.