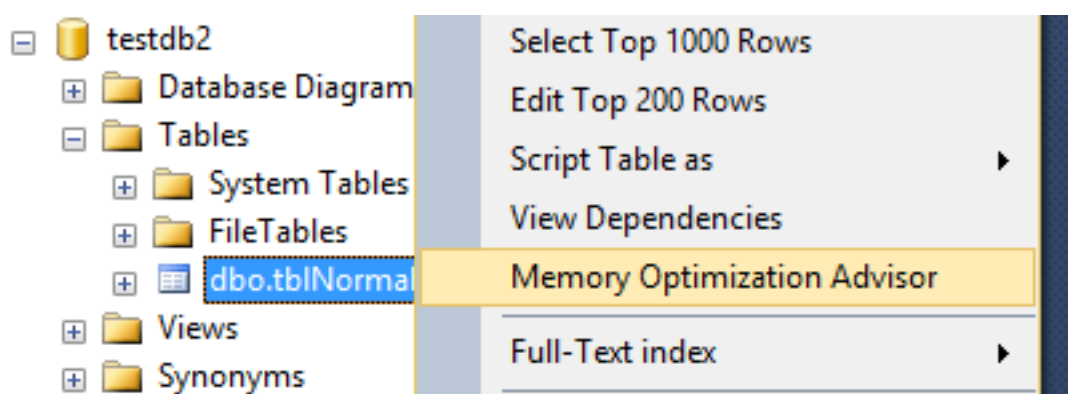


در SQL Server 2014، به Management studio آن ابزارهای جدیدی اضافه شده‌اند تا کار تبدیل و مهاجرت جداول معمولی، به جداول بهینه سازی شده‌ی برای حافظه را ساده‌تر کنند. برای مثال امکان جدیدی به نام Transaction performance collector جهت بررسی کارایی تراکنش‌های جداول و یا رویه‌های ذخیره شده در محیط کاری جاری، طراحی شده‌است. پس از آن، این اطلاعات را آنالیز کرده و بر اساس میزان استفاده از آن‌ها، توصیه‌هایی را در مورد مهاجرت یا عدم نیاز به مهاجرت به سیستم جدید OLTP درون حافظه‌ای ارائه می‌دهد. در ادامه این ابزارهای جدید را بررسی خواهیم کرد.

ابزار Memory Optimization Advisor

Memory Optimization Advisor یک Wizard مانند است که از آن برای گرفتن مشاوره در مورد تبدیل جداول موجود مبتنی بر دیسک سخت، به نمونه‌های بهینه سازی شده برای حافظه می‌توان استفاده کرد. کار آن بررسی ساختار جداولی است که قصد مهاجرت آن‌ها را دارید. برای مثال همانطور که [پیشتر نیز عنوان شد](#)، جداول بهینه سازی شده برای حافظه محدودیت‌هایی دارند؛ مثلاً نباید کلید خارجی داشته باشند. این Wizard یک چنین مواردی را آنالیز کرده و گزارشی را ارائه می‌دهد. پس از اینکه مراحل آن‌را به پایان رساندید و مشکلاتی را که گزارش می‌دهد، برطرف نمودید، کد تبدیل جدول را نیز به صورت خودکار تولید می‌کند. برای دسترسی به آن، فقط کافی است بر روی نام جدول خود کلیک راست کرده و گزینه‌ی memory optimization advisor را انتخاب کنید.



در دو قسمت اول این Wizard، کار بررسی ساختار جدول در حال مهاجرت صورت می‌گیرد. اگر نوع داده‌ای در آن پشتیبانی نشود یا قیود ویژه‌ای در آن تعریف شده باشند، گزارشی را جهت رفع، دریافت خواهید کرد. پس از رفع آن، به صفحه‌ی گزینه‌های مهاجرت می‌رسیم:

Table Memory Optimization Advisor

Review Optimization Options

Introduction
Migration validation
Migration warnings
Migration options
Primary Key migration
Index migration
Summary
Migration progress

Specify options for memory optimization:

Memory-optimized filegroup: InMemory_InMemory

Logical file name: InMemory_InMemory

File path: D:\SQL_Data\MSSQL11.MSSQLSERVER\MSSQL\DATA ...

Rename the original table as: tblNormal_old

Estimated current memory cost (MB): 0

☒ Also copy table data to the new memory optimized table.

By default, this table will be migrated to a memory-optimized table with both schema and data durability.

☐ Check this box to migrate this table to a memory-optimized table with no data durability.

همانطور که ملاحظه می‌کنید، گروه فایل ایجاد شده [در قسمت قبل](#)، به صورت خودکار انتخاب شده‌است. در ادامه می‌توان نام دیگری را برای جدول مبتنی بر دیسک وارد کرد. در اینجا به صورت خودکار کلمه‌ی old به آخر نام جدول اضافه شده‌است. در حین تولید جدول جدید بهینه سازی شده‌ی بر اساس ساختار جدول فعلی، این جدول قدیمی به صورت خودکار تغییر نام خواهد یافت و کلیه اطلاعات آن حفظ می‌شود. همچنین تخمینی را نیز از مقدار حافظه‌ی مورد نیاز برای نگهداری این جدول جدید درون حافظه‌ای نیز ارائه می‌دهد. در این مثال چون رکوردی در جدول انتخابی وجود نداشته‌است، تخمین آن صفر است. عدد ارائه شده توسط آن بسیار مهم است و باید به همین میزان برای سیستم خود حافظه تهیه نمائید و یا از حافظه‌ی موجود استفاده کنید. در پایین صفحه می‌توان انتخاب کرد که آیا داده‌های جدول فعلی، به جدول درون حافظه‌ای انتقال یابند یا خیر. به علاوه نوع ماندگاری اطلاعات آن نیز قابل تنظیم است. اگر گزینه‌ی آخر را انتخاب کنید به معنای حالت SCHEMA_ONLY است. حالت پیش فرض آن SCHEMA_AND_DATA می‌باشد که در [قسمت‌های قبل](#) بیشتر در مورد آن بحث شد.

در دو صفحه‌ی بعد، کار انتخاب hash index و range index انجام می‌شود:

Table Memory Optimization Advisor

Review Primary Key Conversion

Introduction
Migration validation
Migration warnings
Migration options
Primary Key migration
Index migration
Summary
Migration progress

Please choose the appropriate conversion for this primary key:

Column	Type	Collation
<input checked="" type="checkbox"/> CustomerID	int	

Select a new name for this primary key:

Select the type of this primary key:

☒ Use NONCLUSTERED HASH index
A NONCLUSTERED HASH index provides the most benefit for point lookups. It provides no discernible benefit if a query is running a Range scan.
Bucket Count:
The Bucket Count of a NONCLUSTERED HASH index is the number of buckets in the hash table. It is recommended to set the fill factor to 50 to 60% if the table requires a lot of space for growth. Bucket Count will be rounded up to the nearest power of two.

☐ Use NONCLUSTERED index
A NONCLUSTERED index provides the most benefit for range predicates and ORDER BY clauses. NONCLUSTERED indexes are unidirectional. It provides no benefit for ORDER BY clauses with orders different from the index.
Sort column and order:

Column	Sort Order
--------	------------

< Previous Next > Migrate Cancel

در اینجا hash index بر روی فیلد ID تولید شده‌است، به همراه تعیین bucket count آن و در صفحه‌ی بعدی range index بر روی فیلد تاریخ تعریف گردیده‌است:

Table Memory Optimization Advisor

Review Index Conversion

Introduction
Migration validation
Migration warnings
Migration options
Primary Key migration
Index migration
Summary
Migration progress

Please choose the appropriate conversion for this index:

Column	Type	Collation
<input checked="" type="checkbox"/> CustomerSince	datetime	

Select the type of this index:

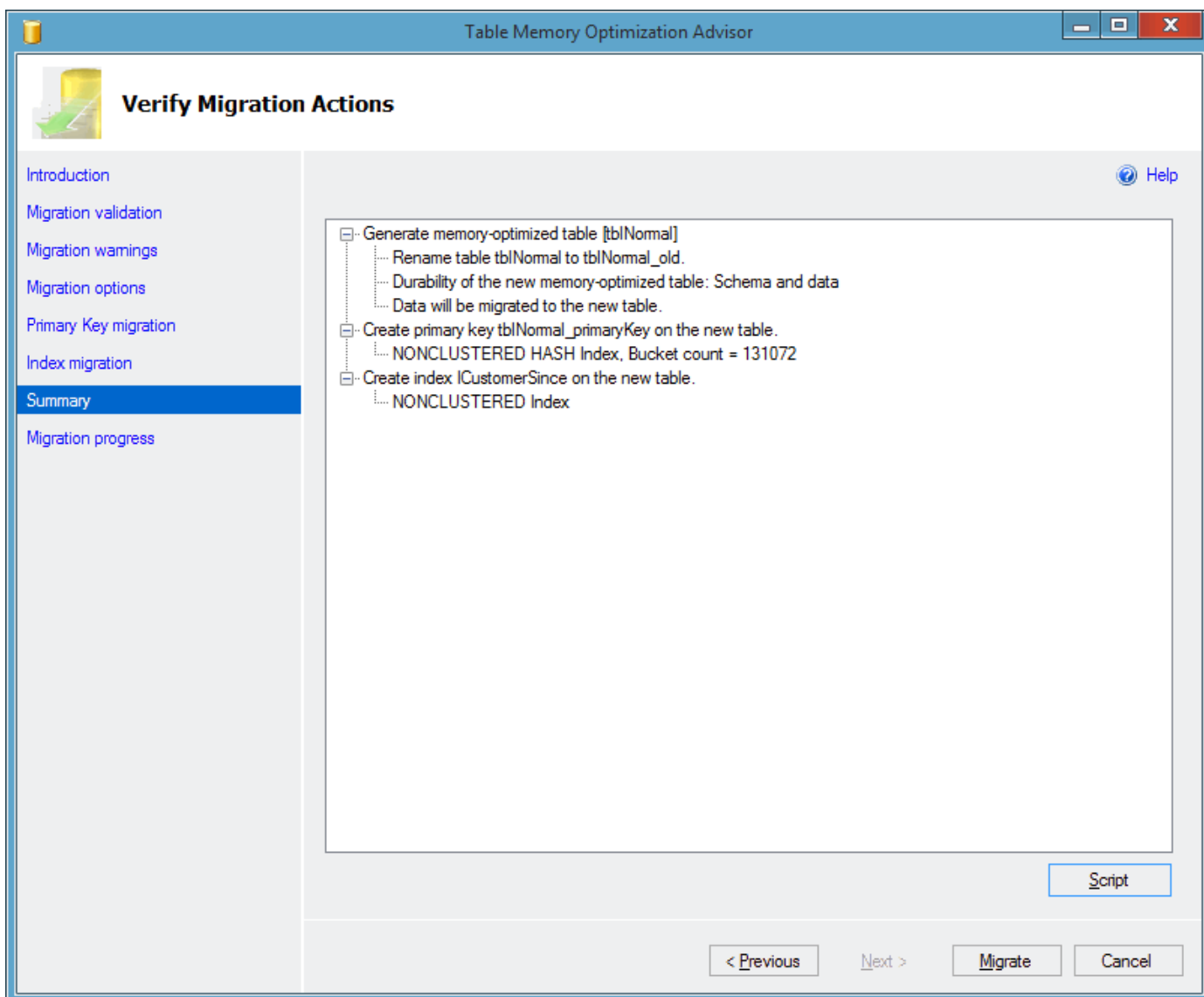
☐ Use NONCLUSTERED HASH index
A NONCLUSTERED HASH index provides the most benefit for point lookups. It provides no discernible benefit if a query is running a Range scan.
Bucket Count:
The Bucket Count of a NONCLUSTERED HASH index is the number of buckets in the hash table. It is recommended to set the fill factor to 50 to 60% if the table requires a lot of space for growth. Bucket Count will be rounded up to the nearest power of two.

☒ Use NONCLUSTERED index
A NONCLUSTERED index provides the most benefit for range predicates and ORDER BY clauses. NONCLUSTERED indexes are unidirectional. It provides no benefit for ORDER BY clauses with orders different from the index.
Sort column and order:

Column	Sort Order
CustomerSince	ASC

< Previous Next > Migrate Cancel

در آخر می‌توان با کلیک بر روی دکمه‌ی Script، صرفاً دستورات T-SQL تغییر ساختار جدول را دریافت کرد و یا با کلیک بر روی دکمه‌ی migrate به صورت خودکار کلیه موارد تنظیم شده را اجرا نمود.



خلاصه‌ی این مراحل که توسط دکمه‌ی Script آن تولید می‌شود، به صورت زیر است:

```
USE [testdb2]
GO

EXEC dbo.sp_rename @objname = N'[dbo].[tblNormal]', @newname = N'tblNormal_old', @objtype = N'OBJECT'
GO

USE [testdb2]
GO

SET ANSI_NULLS ON
GO

CREATE TABLE [dbo].[tblNormal]
(
    [CustomerID] [int] NOT NULL,
    [Name] [nvarchar](250) COLLATE Persian_100_CI_AI NOT NULL,
    [CustomerSince] [datetime] NOT NULL,
)
INDEX [ICustomerSince] NONCLUSTERED
(
    [CustomerSince] ASC
),
CONSTRAINT [tblNormal_primaryKey] PRIMARY KEY NONCLUSTERED HASH
(
    [CustomerID]
)WITH (BUCKET_COUNT = 131072)
```

```

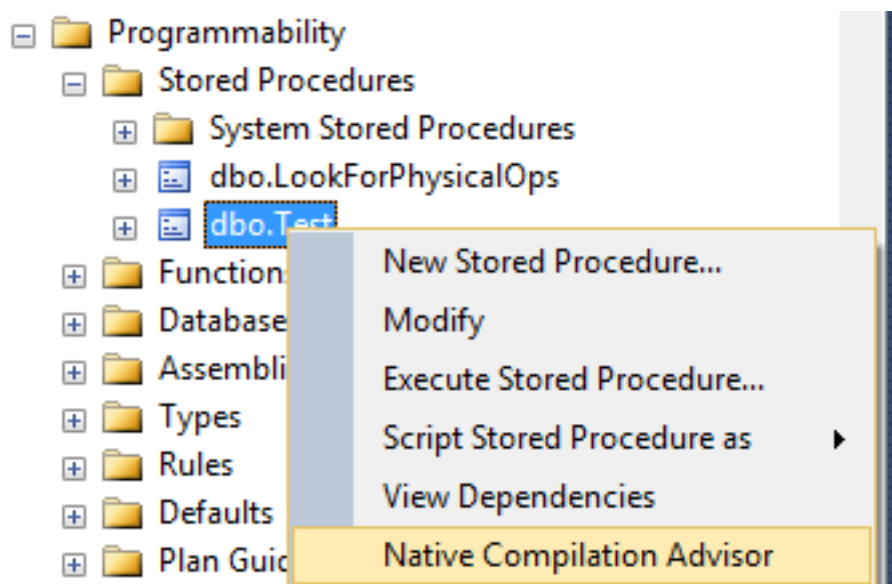
)WITH ( MEMORY_OPTIMIZED = ON , DURABILITY = SCHEMA_AND_DATA )
GO

INSERT INTO [testdb2].[dbo].[tblNormal] ([CustomerID], [Name], [CustomerSince]) SELECT [CustomerID],
[Name], [CustomerSince] FROM [testdb2].[dbo].[tblNormal_old]
GO

```

که در آن ابتدا کار تغییر نام جدول قبلی صورت می‌گیرد. سپس یک جدول جدید با ویژگی MEMORY_OPTIMIZED = ON را ایجاد می‌کند. در ساختار این جدول، hash index و range index تعریف شده، قابل مشاهده هستند. در آخر نیز کلیه اطلاعات جدول قدیمی را به جدول جدید منتقل می‌کند.

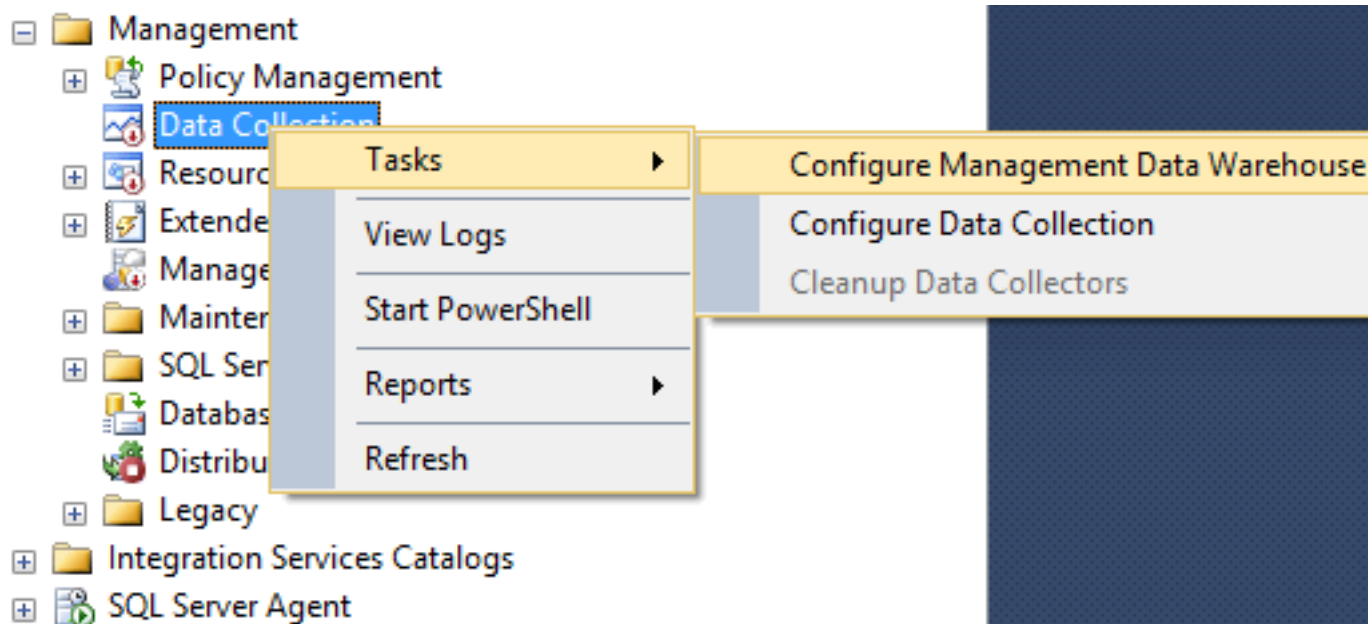
علاوه بر memory optimization advisor مخصوص جداول، ابزار دیگری نیز به نام **Native compilation advisor** برای آنالیز رویه‌های ذخیره شده تهیه شده است:



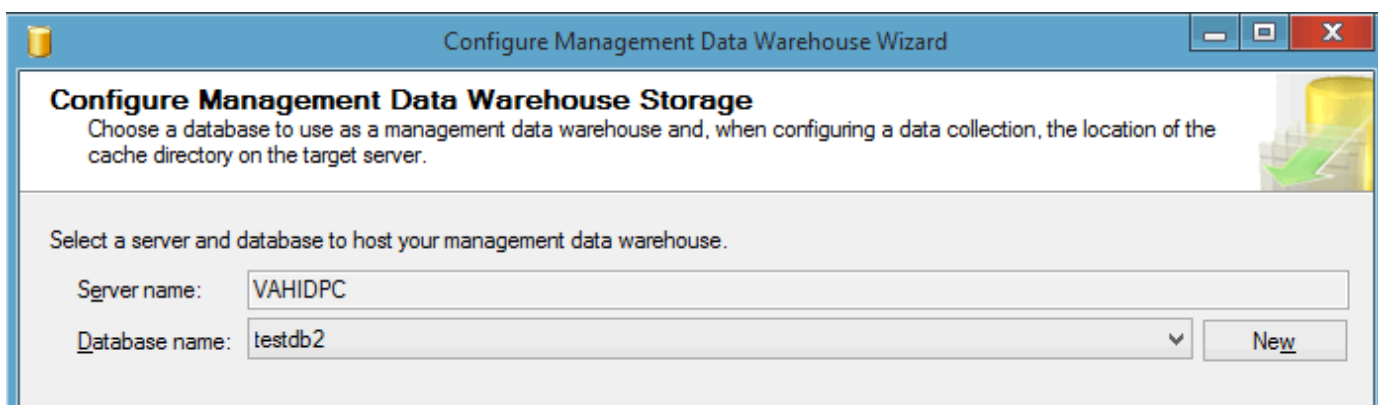
آیا سیستم فعلی ما واقعا نیازی به ارتقاء به جداول درون حافظه‌ای دارد؟

تا اینجا در مورد نحوه‌ی ایجاد جداول درون حافظه‌ای و یا نحوه‌ی تبدیل جداول موجود را به ساختار جدید بررسی کردیم. ولی آیا واقعا یک چنین تغییری برای ما سودمند هستند؟ برای پاسخ دادن به این سؤال ابزاری به نام AMR به SQL Server 2014 اضافه شده است (Analyze, Migrate, Report). کار آن تحت نظر قرار دادن جداول و رویه‌های ذخیره شده‌ی بانک اطلاعاتی است و سپس بر اساس بار سیستم، تعداد درخواست‌های همزمان و میزان استفاده از جداول و تراکنش‌های مرتبط با آن‌ها، گزارشی را ارائه می‌دهد. بر این اساس بهتر می‌توان تصمیم گرفت که کدام جداول بهتر است به جداول درون حافظه‌ای تبدیل شوند. برای تنظیم آن باید مراحل ذیل طی شوند:

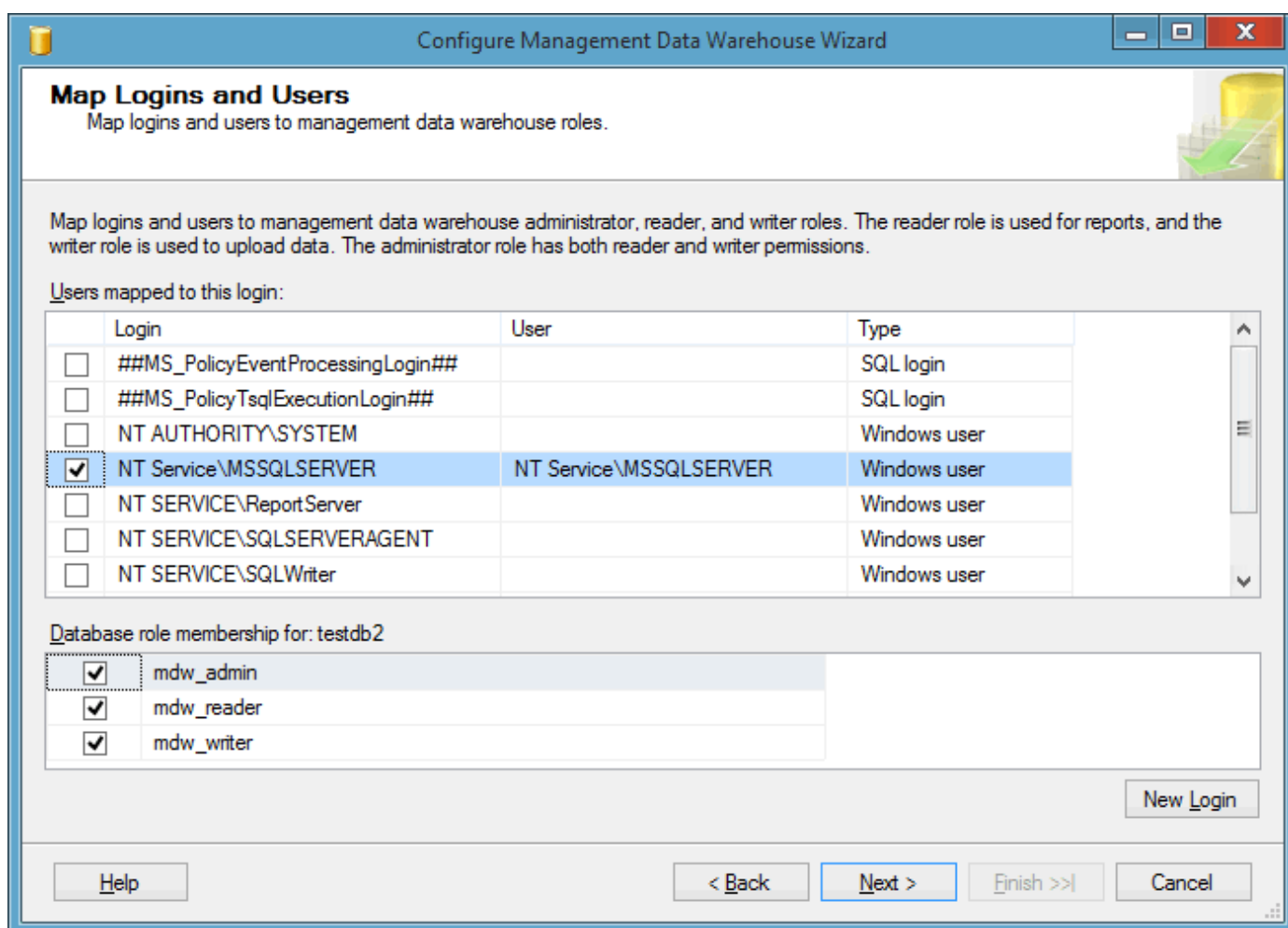
در Management Studio، به برگه‌ی Object Explorer آن مراجعه کنید. سپس پوشه‌ی Management آن را یافته و بر روی گزینه‌ی Data Collection کلیک راست نمایید:



در اینجا گزینه‌ی Configure Management Data Warehouse را انتخاب نمائید. در صفحه‌ی باز شده، ابتدا بانک اطلاعاتی مدنظر را انتخاب نمائید. همچنین بهتر است بر روی دکمه‌ی new کلیک کرده و یک بانک اطلاعاتی جدید را برای آن ایجاد نمائید، تا دچار تداخل اطلاعاتی و ساختاری نگردد:

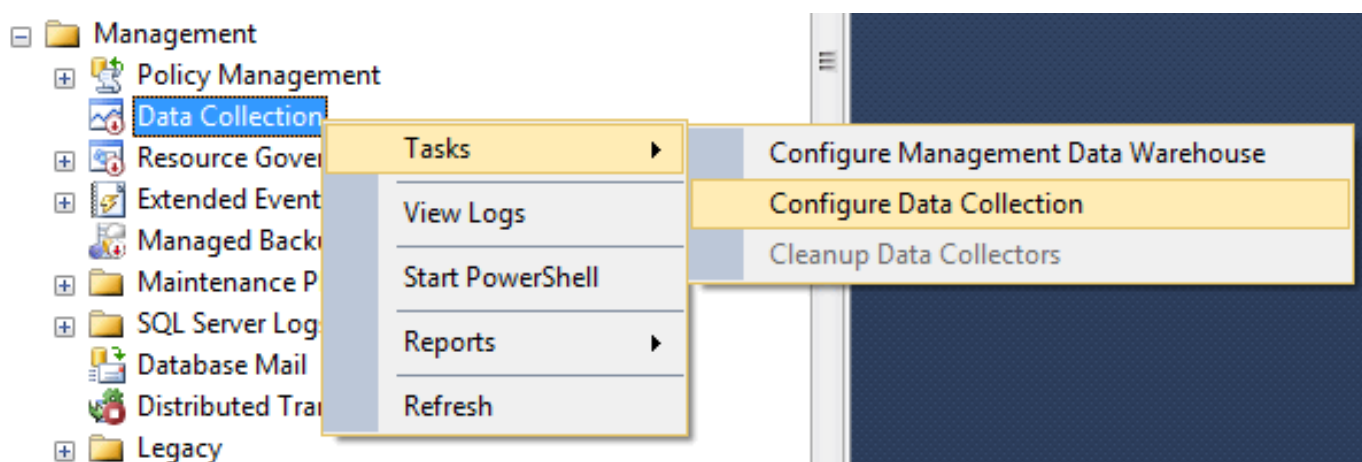


در ادامه نام کاربری را که قرار است کار مدیریت ثبت و جمع آوری اطلاعات را انجام دهد، به همراه نقش‌های آن انتخاب نمائید:



و در آخر در صفحه‌ی بعدی بر روی دکمه‌ی Finish کلیک کنید.

پس از ایجاد و انتخاب بانک اطلاعاتی Management Data Warehouse، نوبت به تنظیم گزینه‌های جمع‌آوری اطلاعات است:



در اینجا ابتدا سرور جاری را انتخاب کنید. پس از آن به صورت خودکار در لیست بانک‌های اطلاعاتی قابل انتخاب، تنها همان بانک

اطلاعاتی جدیدی را که برای مرحله‌ی قبل ایجاد کردیم، می‌توان مشاهده کرد.

Configure Data Collection Wizard

Setup Data Collection Sets
Choose a set of Data Collectors to create and start.

Select a server and database that is the host for your management data warehouse.

Server name: VAHIDPC ...

Database name: DWMem

Enter where you want to cache collected data locally before it is uploaded to the management data warehouse. A blank value uses the TEMP directory of the collector process.

Cache directory: ...

Select data collector sets you want to enable:

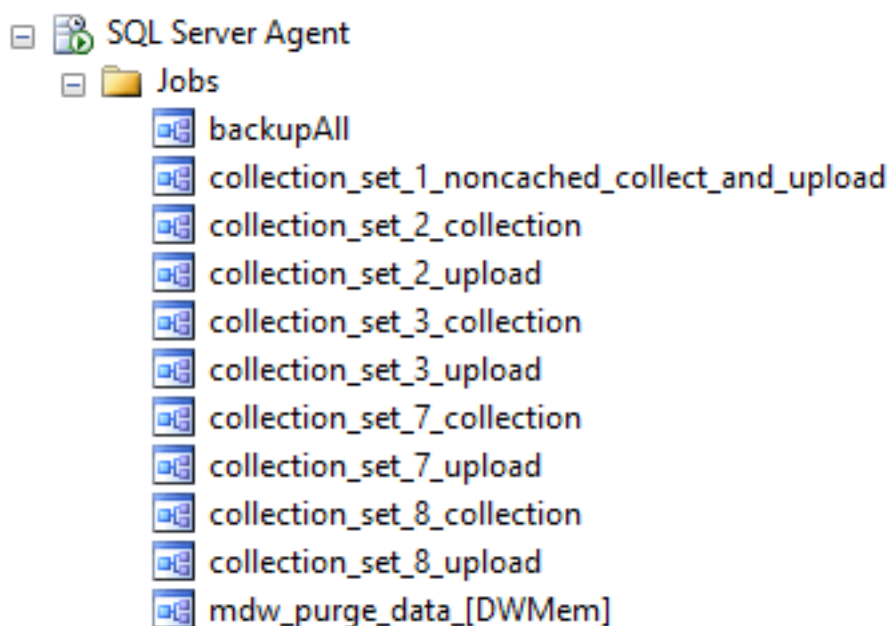
	Name	Description
<input checked="" type="checkbox"/>	System Data Collection Sets	Collect performance statistics for general purpose troubleshooting.
<input checked="" type="checkbox"/>	Transaction Performance Collection Sets	Collect statistics for transaction performance issues.

☐ Use a SQL Server Agent proxy for remote uploads.

Help < Back Next > Finish >> Cancel

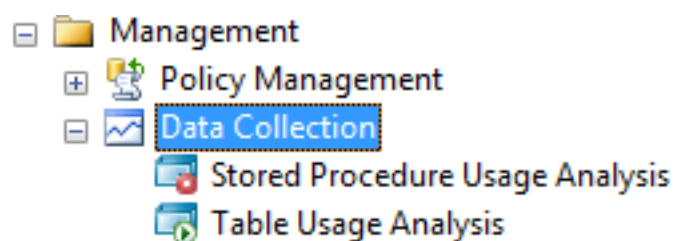
در صفحه‌ی بعد، گزینه‌ی «Transaction Performance Collection Sets» را انتخاب نمائید که دقیقاً گزینه‌ی مدنظر ما جهت یافتن آماری از وضعیت تراکنش‌های سیستم است. در ادامه بر روی گزینه‌های next و finish کلیک کنید تا کار تنظیمات به پایان برسد.

اکنون اگر به لیست وظایف تعریف شده در SQL Server agent مراجعه کنید، می‌توانید، وظایف مرتبط با جمع‌آوری داده‌ها را نیز مشاهده نمائید:

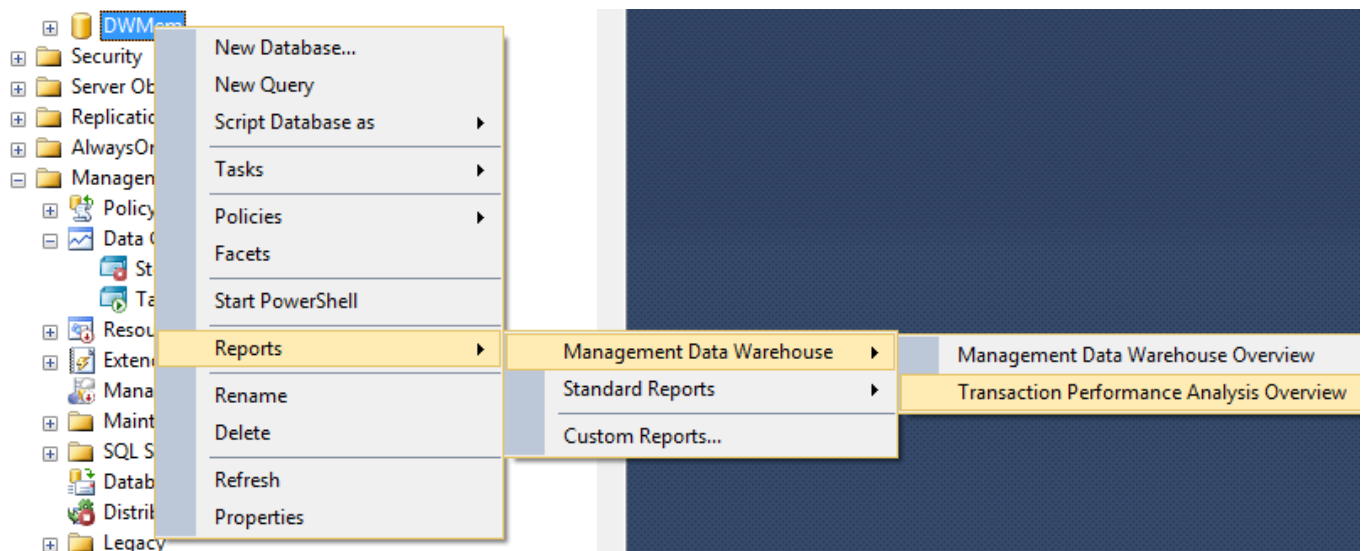


وظایف Stored Procedure Usage Analysis هر نیم ساعت یکبار و وظایف Table Usage Analysis هر 15 دقیقه یکبار اجرا می‌شوند. البته امکان اجرای دستی این وظایف نیز مانند سایر وظایف SQL Server وجود دارند.

همچنین در پوشه‌ی management، گزینه‌ی Data collection نیز دو زیر شاخه اضافه شده‌اند که نمایانگر آنالیز میزان مصرف جداول و رویه‌های ذخیره شده می‌باشند:



پس از این کارها باید مدتی صبر کنید (مثلاً یک ساعت) تا سیستم به صورت معمول کارهای متداول خودش را انجام دهد. پس از آن می‌توان به گزارشات AMR مراجعه کرد.



برای اینکار بر روی بانک اطلاعاتی Management Data Warehouse که در ابتدای عملیات ایجاد شد، کلیک راست نمائید و سپس مراحل ذیل را طی کنید:

Reports > Management Data Warehouse > Transaction Performance Analysis Overview

Transaction Performance Analysis Overview

Microsoft SQL Server 2014

on VAHIDPC at 02/06/2014 02:45:27 پ.ظ

Welcome to the AMR tool for in-memory OLTP.

This report helps you identify bottlenecks in your database and provide assistance to migrate them to in-memory OLTP. To begin, click on the last snapshot upload time hyperlink of one of these options to see the report.



Tables Analysis

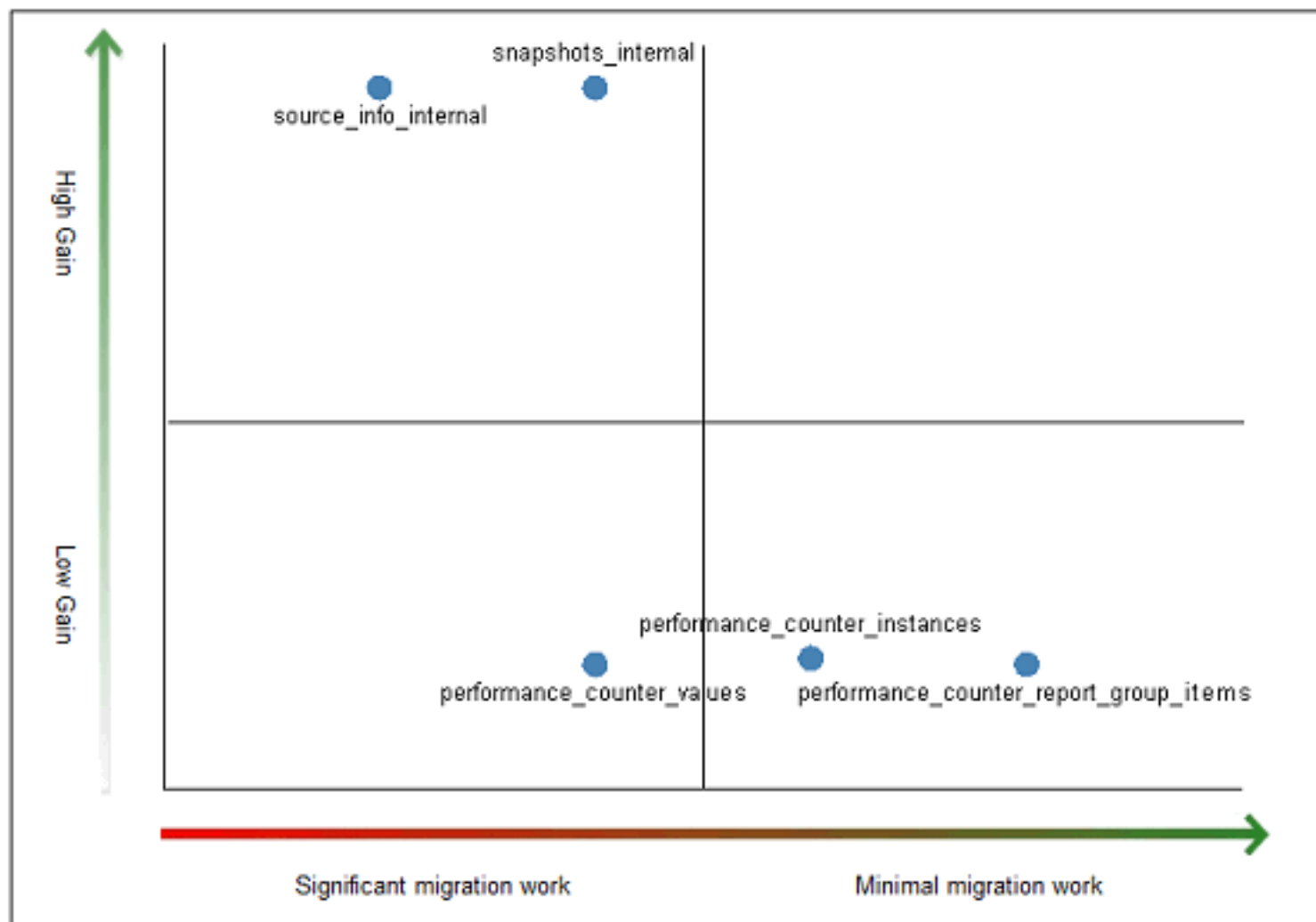


Stored Procedure Analysis

Instance Name	Usage Analysis	Contention Analysis	Usage Analysis
VAHIDPC	02/06/2014 02:45:04 پ.ظ	02/06/2014 02:45:04 پ.ظ	(No Data)

در گزارش ایجاد شده، ذیل گزینه‌ی usage analysis لینک‌هایی وجود دارند که با مراجعه به آن‌ها، چارت‌هایی از میزان مصرف بانک‌های اطلاعاتی مختلف سیستم ارائه می‌شود. اگر پیام No data available را مشاهده کردید، یعنی هنوز باید مقداری صبر کنید تا کار جمع‌آوری اطلاعات به پایان برسد.

در این چارت‌ها بانک‌های اطلاعاتی که در سمت راست، بالای تصویر قرار می‌گیرند، انتخاب مناسبی برای تبدیل به بانک‌های اطلاعاتی درون حافظه‌ای هستند. محور افقی آن از چپ به راست بیانگر میزان کاهش سختی انتقال یک جدول به جدول درون حافظه‌ای است (با در نظر گرفتن تمام مسایلی که باید تغییر کنند یا نوع‌های داده‌ای که باید اصلاح شوند) و محور عمودی آن نمایانگر میزان بالا رفتن پاسخ‌دهی سیستم در جهت انجام کار بیشتر است.



هر زمان هم که کار تصمیم‌گیری شما به پایان رسید، می‌توانید بر روی گزینه‌ی Data collection کلیک راست کرده و آن را غیرفعال نمایید.

برای مطالعه بیشتر

[SQL Server 2014 Field Benchmarking In-Memory OLTP and Buffer Pool Extension Features](#)

[New AMR Tool: Simplifying the Migration to In-Memory OLTP](#)

[A Tour of the Hekaton AMR Tool](#)

[SQL Server 2014 Memory Optimization Advisor](#)

[Getting started with the AMR tool for migration to SQL Server In-memory OLTP Tables](#)

[How to Use Microsoft's AMR Tool](#)

[SQL Server 2014's Analysis, Migrate, and Report Tool](#)