

Minggu-14

Database Maintenance

TIM PENGAJAR SMBD

Outline



Memastikan integritas basis data



Memelihara indeks.



Melakukan pemeliharaan database rutin

2

Latar Belakang

- Agar engine database memberikan outcome terbaik maka diperlukan perawatan rutin terjadwal

1. Memastikan Integritas Database

- Database yang korup memang sangat jarang terjadi. Namun demikian, dalam maintenance sangat penting dilakukan pengecekan untuk memastikan tidak ada database yang korup.
- Database yang korup harus mampu segera dideteksi, sebelum terjadi masalah yang lebih serius
- Perintah DBCC CHECKDB dapat digunakan untuk mendeteksi database korup. Dan dalam situasi tertentu dapat digunakan untuk memperbaiki database yang korup

perintah DBCC CHECKDB

- Mengecek integritas database fisikal maupun logikal
 - Alokasi semua page di database
 - Konsistensi tabel dan index
 - Konsistensi katalog database
 - Konsistensi level LINK untuk obyek FILESTREAM
 - Obyek service broker
- Memiliki beberapa opsi untuk repair
 - Beberapa opsi memungkinkan terjadi kehilangan data
- Perlu rutin dijalankan
 - Harus sinkron dengan strategi backup, agar dapat me-recover database corrupt

Beberapa fungsi lain dalam DBCC yang dapat dijalankan terpisah jika diinginkan

Option	Description
DBCC CHECKALLOC	Checks the consistency of disk space allocation structures for a specified database.
DBCC CHECKTABLE	Checks the pages associated with a specified table and the pointers between pages that are associated with the table. DBCC CHECKDB executes DBCC CHECKTABLE for every table in the database.
DBCC CHECKCATALOG	Checks the database catalog by performing logical consistency checks on the metadata tables in the database. These metadata tables are used to hold information that describes both system and user tables and other database objects. DBCC CHECKCATALOG does not check user tables.

DBCC CHECKDB Options

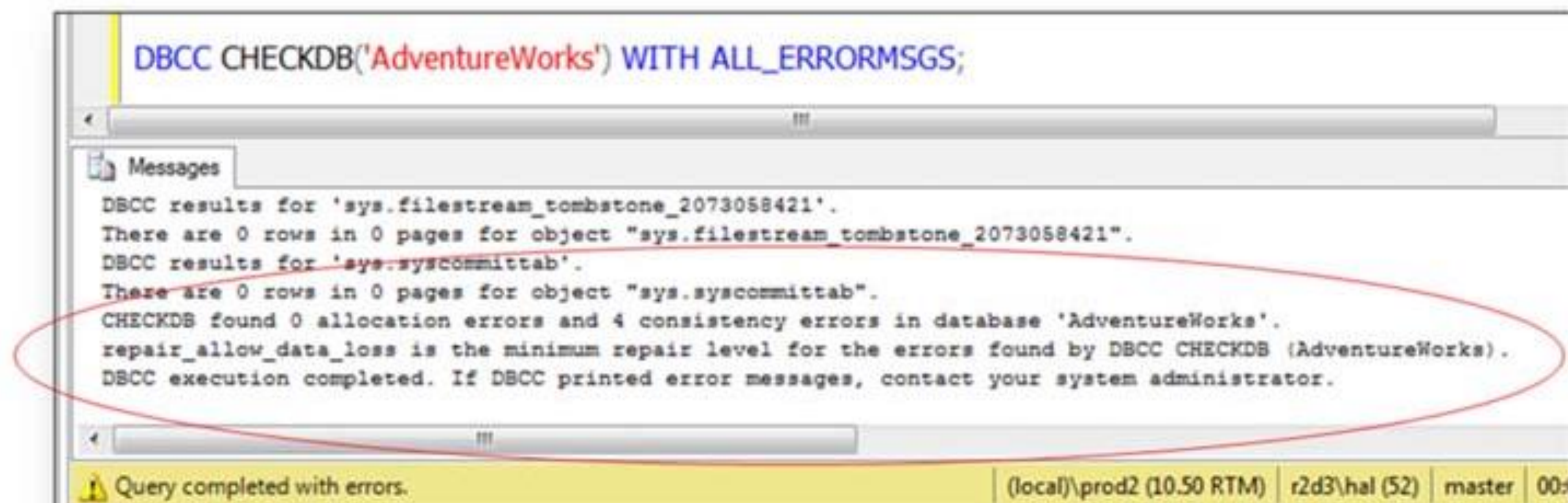
Option	Description
PHYSICAL_ONLY	Only checks the physical integrity with less overhead
NOINDEX	Does not perform logical checks on nonclustered indexes
EXTENDED_LOGICAL_CHECKS	Performs additional logical checks of indexed views, spatial and XML indexes
TABLOCK	Uses locks instead of database snapshots
ALL_ERRORMSGSGS	Returns all error messages instead of the default action that returns the first 200
NO_INFOMSGSGS	Returns only error messages and no informational messages
ESTIMATEONLY	Estimates the amount of tempdb space that is required for execution

- PHYSICAL_ONLY. sering digunakan pada sistem yang sedang berjalan(production) karena secara substansial mengurangi waktu yang diperlukan untuk menjalankan DBCC CHECKDB pada database besar
- NOINDEX. tidak melakukan pemeriksaan intensif terhadap indeks non-Clustered untuk tabel user. Dapat mengurangi waktu eksekusi keseluruhan tetapi tidak mempengaruhi tabel sistem
- EXTENDED_LOGICAL_CHECKS hanya dapat dilakukan ketika *database* berada dalam tingkat kompatibilitas *database* 100 (SQL Server 2008) atau lebih tinggi.
- TABLOCK digunakan agar DBCC CHECKDB dapat melakukan *table lock* pada setiap tabel sambil melakukan *consistency check*
- ALL_ERRORMSGSGS dan NO_INFOMSGSGS hanya memengaruhi output dari perintah tetapi bukan operasi yang dilakukan oleh perintah
- ESTIMATEONLY memperkirakan kebutuhan ruang dalam tempdb

Opsi perbaikan DBCC CHECKDB

Selain memberikan rincian kesalahan, output DBCC CHECKDB juga menunjukkan opsi perbaikan yang mungkin diperlukan untuk memperbaiki masalah.

- DBCC CHECKDB menawarkan dua opsi perbaikan
 - a) **REPAIR_REBUILD** membangun kembali indeks. *Page* data yang rusak dihapus. Opsi ini untuk *corruption* ringan yang tidak mengakibatkan kehilangan data.
 - b) **REPAIR_ALLOW_DATA_LOSS** hampir selalu mengakibatkan kehilangan data. Melakukan dealokasi *page* yang rusak/*corrupt* dan *page* lain yang merujuk ke *page* yang rusak. Kehilangan data yang signifikan bisa saja terjadi



```
DBCC CHECKDB('AdventureWorks') WITH ALL_ERRORMSG;

Messages
DBCC results for 'sys.filestream_tombstone_2073058421'.
There are 0 rows in 0 pages for object "sys.filestream_tombstone_2073058421".
DBCC results for 'sys.syscommittab'.
There are 0 rows in 0 pages for object "sys.syscommittab".
CHECKDB found 0 allocation errors and 4 consistency errors in database 'AdventureWorks'.
repair_allow_data_loss is the minimum repair level for the errors found by DBCC CHECKDB (AdventureWorks).
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

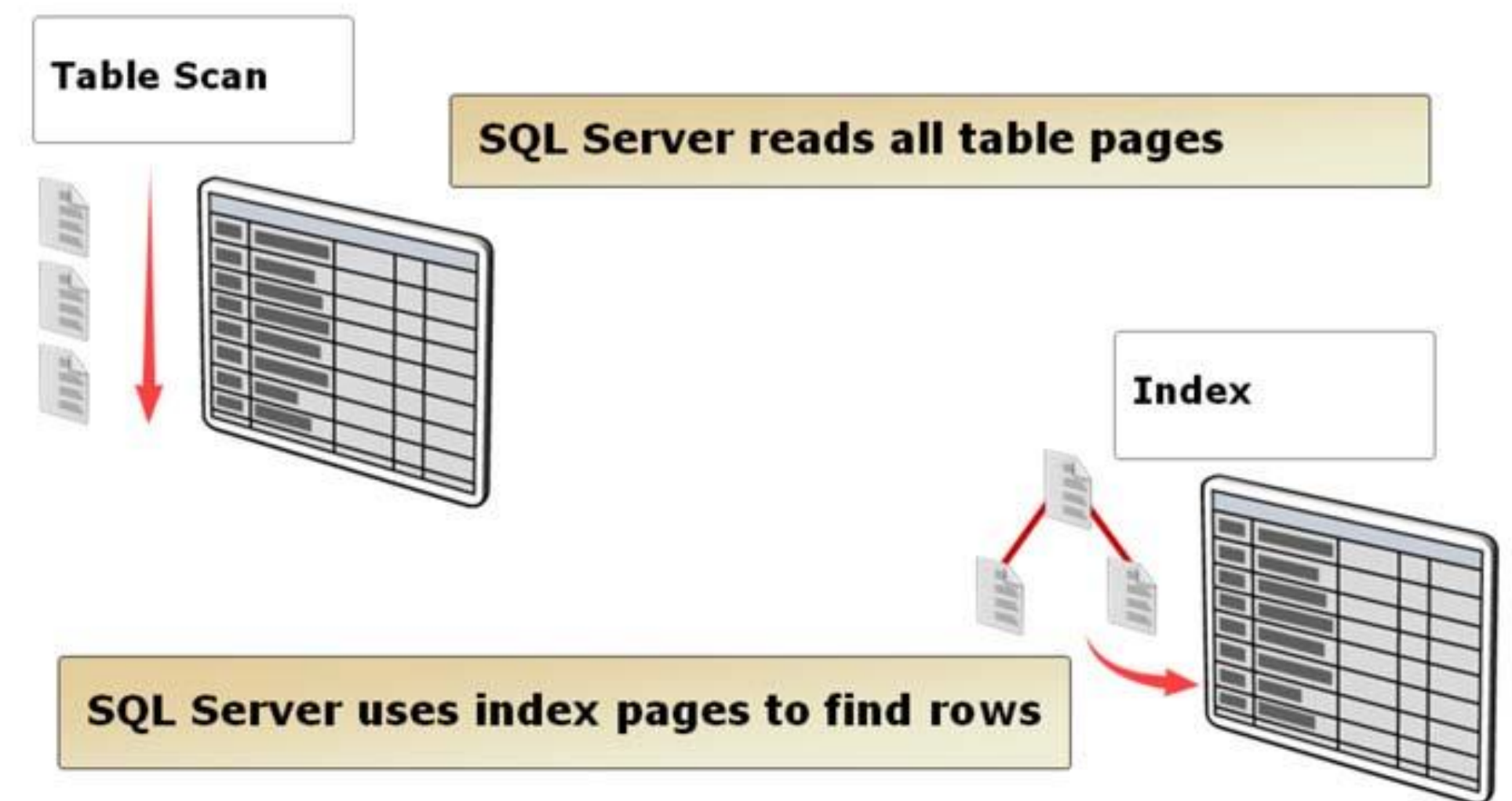
Query completed with errors. (local)\prod2 (10.50 RTM) | r2d3\hal (52) | master | 00:
```

2. Memelihara indeks

- Perlu memelihara indeks untuk mengoptimalkan performa database.
- Seiring berjalannya waktu, indeks dapat terfragmentasi. Akibatnya, performa database dapat menurun.
- Untuk mengembalikan performa perlu dilakukan defragmentasi indeks atau membangun ulang indeks

Indeks mempengaruhi performa

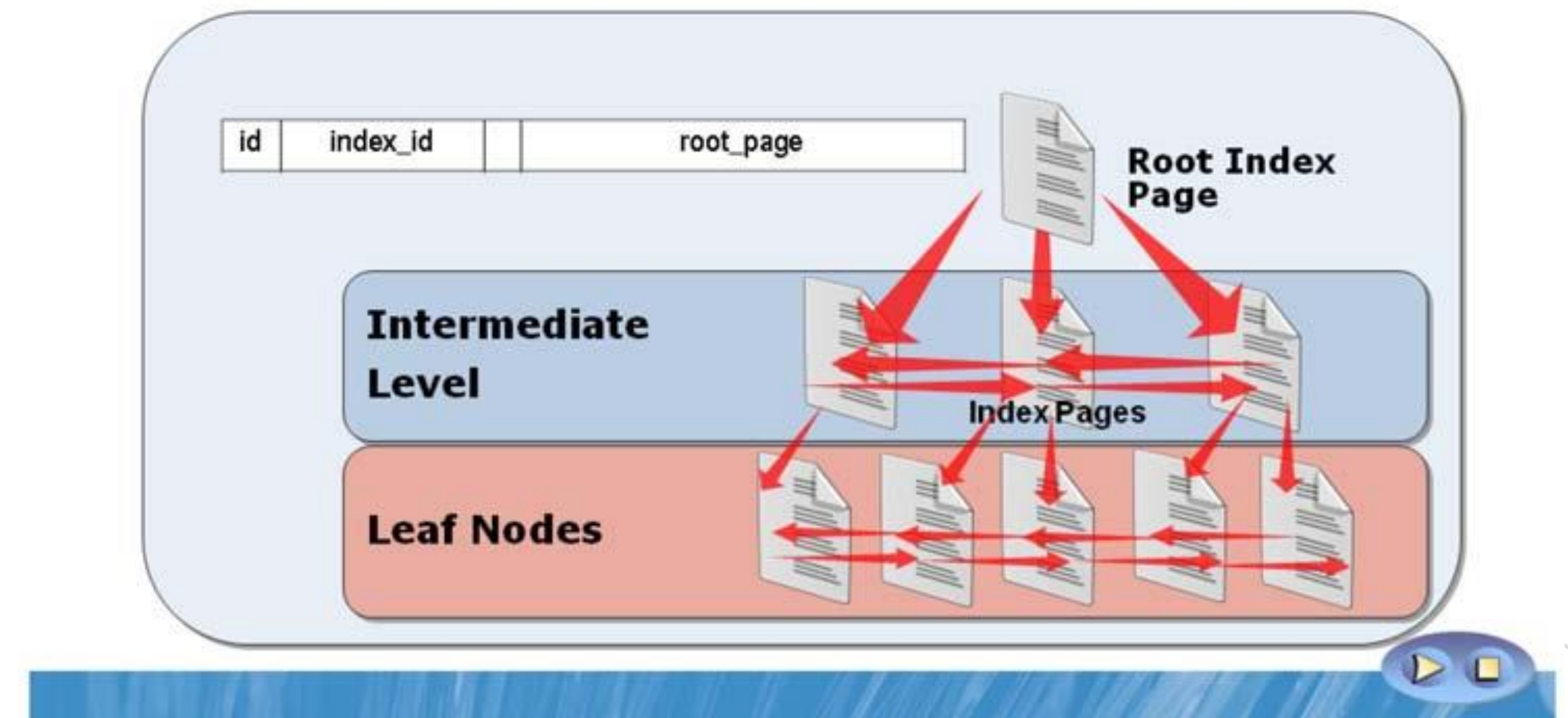
- Dua cara Sql server mengakses data dari sebuah table yaitu: dengan cara membaca semua halaman atau cukup melalui indeks untuk menemukan baris yang diinginkan.
- Menggunakan indeks dapat mengurangi effort dalam mencari baris data.
- Indeks dapat meningkatkan performa ketika melakukan searching, sorting, serta join dalam menjalankan query
- Namun indeks dapat mempengaruhi performa aktivitas modifikasi(update) data, serta butuh ekstra space



Tipe Indeks SQL Server

- Indeks Clustered
 - Terdapat pengurutan row dalam page. Pengurutannya berdasarkan sebuah kolom atau lebih(disebut: clustering key) yang ditentukan.
 - Sebuah tabel hanya dapat memiliki sebuah indeks clustered
- Indeks NonClustered
 - tidak mempengaruhi tata letak data dalam tabel seperti yang ada pada indeks clustered

- Indexes are organised as B-Trees
- Clustered index has data pages in the leaf level
- Nonclustered index has pointer to data rows in leaf level

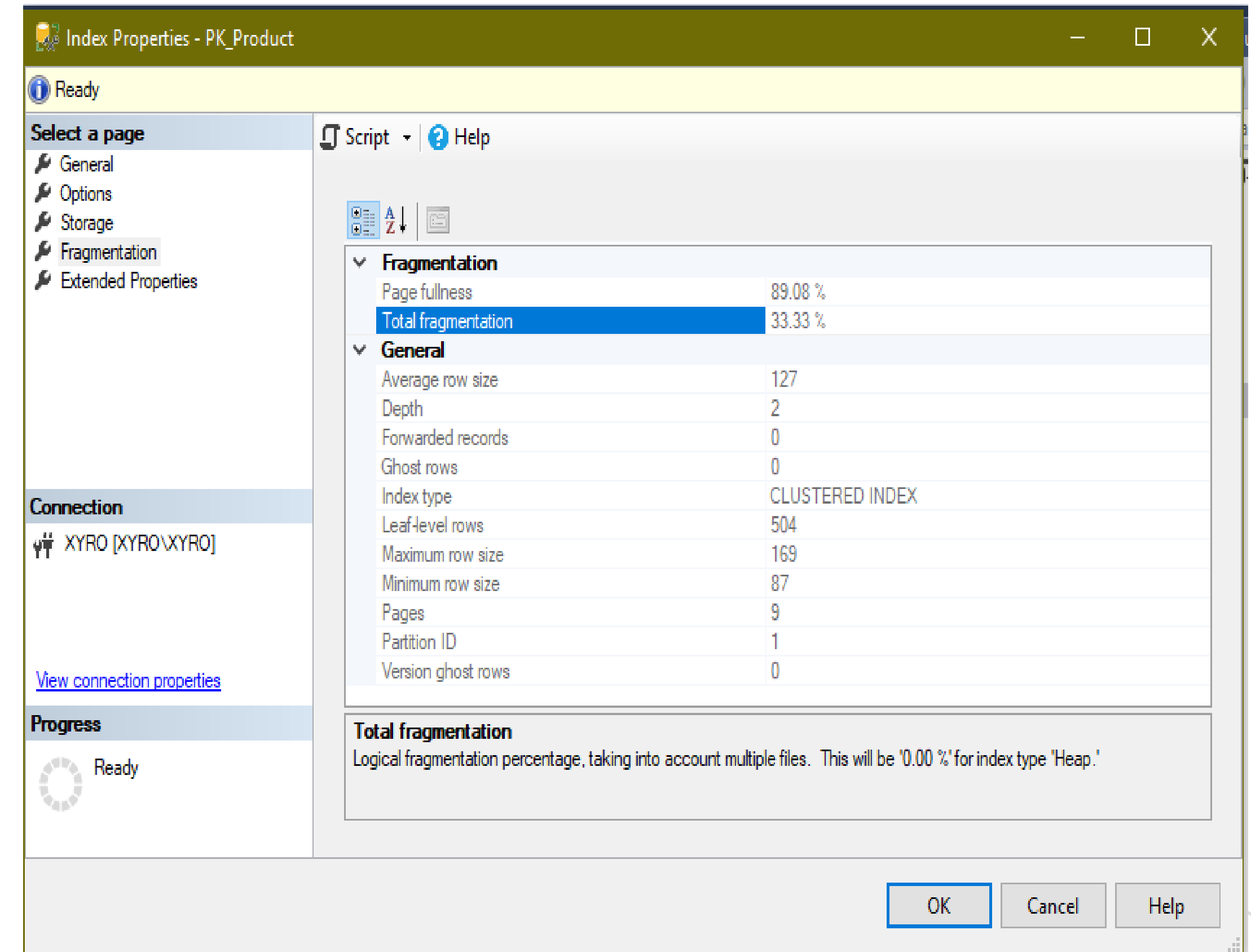


Fragmentasi Indeks

- Fragmentasi indeks merupakan penggunaan page yang tidak efisien dalam indeks. Fragmentasi terjadi seiring waktu terjadi perubahan data.
- Untuk operasi membaca data, indeks dapat berkinerja dengan sangat baik ketika setiap page indeks terisi penuh. Akan tetapi, aktifitas modifikasi pada data dalam indeks menyebabkan perlu untuk memecah page indeks sehingga terdapat ruang kosong
- **Fragmentasi Internal.** terjadi ketika page tidak menahan data sebanyak yang mampu mereka tahan. Terjadi ketika page terpecah saat penyisipan (insert) juga dapat terjadi saat operasi pembaruan (update) yang menyebabkan baris dipindahkan ke page lain. Sehingga, ruang kosong tersisa di dalam page
- **Fragmentasi eksternal.** terjadi ketika page yang diurutkan secara logis tidak disimpan dalam nomor page yang terurut. Jika page indeks baru perlu dialokasikan, maka page tersebut akan dimasukkan secara logis ke lokasi yang benar dalam daftar page tetapi bisa juga ditempatkan di akhir indeks.

Mendeteksi Fragmentasi

- Untuk mendeteksi fragmentasi, SQL Server menyediakan kolom `avg_fragmentation_in_percent` dari `sys.dm_db_index_physical_stats` pada dynamic management view
- SQL Server Management Studio juga menyediakan rincian indeks fragmentasi di page properti untuk setiap indeks



FILLFACTOR dan PAD_INDEX

- Digunakan untuk mendapatkan free space dalam page indeks. Dalam situasi tertentu meningkatkan performa INSERT serta UPDATE, namun mempengaruhi performa SELECT.
- Dampak penurunan performa akibat pemisahan page dapat dikurangi dengan mengosongkan ruang kosong di setiap halaman saat membuat indeks, termasuk indeks clustered.
- Dengan menentukan FILLFACTOR, misalkan FILLFACTOR = 70, artinya diisi 70% dan dikosongkan 30%
- **PAD_INDEX**. jika diaktifkan, mengalokasikan free space di level non-leaf pada indeks

- Free space can be left in indexes, including clustered indexes
 - FILLFACTOR (leaf level only)
 - PAD_INDEX (intermediate and root levels also)

```
ALTER TABLE Person.Contact
ADD CONSTRAINT PK_Contact_ContactID
PRIMARY KEY CLUSTERED
(
    ContactID ASC
) WITH (PAD_INDEX = OFF, FILLFACTOR = 70);
GO
```


Pemeliharaan Indeks

- Ada 2 opsi untuk menghilangkan fragmentasi pada indeks clustered maupun nonclustered: **Rebuild** dan **Reorganize**
- **Rebuild.** menghapus dan menciptakan kembali indeks untuk menghilangkan fragmentasi. Menggunakan kembali kembali disk space dengan memadatkan page berdasarkan fill factor yang ditentukan atau yang ada sebelumnya, kemudian menata ulang baris indeks di page yang berdekatan
- **Reorganize.** memerlukan sumber daya sistem yang lebih minimal. Reorganize indeks juga memadatkan page indeks. Pemadatan didasarkan pada nilai fill factor yang ada. Dimungkinkan untuk menghentikan proses Reorganize tanpa kehilangan pekerjaan yang telah dilakukan. Artinya dapat melakukan reorganize parsial pada indeks yang besar setiap hari.

- **Rebuild**

- Rebuilds the whole index
- Needs free space in database
- Performed as a single transaction with potential requirement for a large amount of transaction log space

- **Reorganize**

- Sorts the pages and is always online
- Less transaction log usage
- Can be interrupted but still retain work performed to that point

```
ALTER INDEX CL_LogTime ON dbo.LogTime  
REBUILD;
```

```
ALTER INDEX ALL ON dbo.LogTime  
REORGANIZE;
```

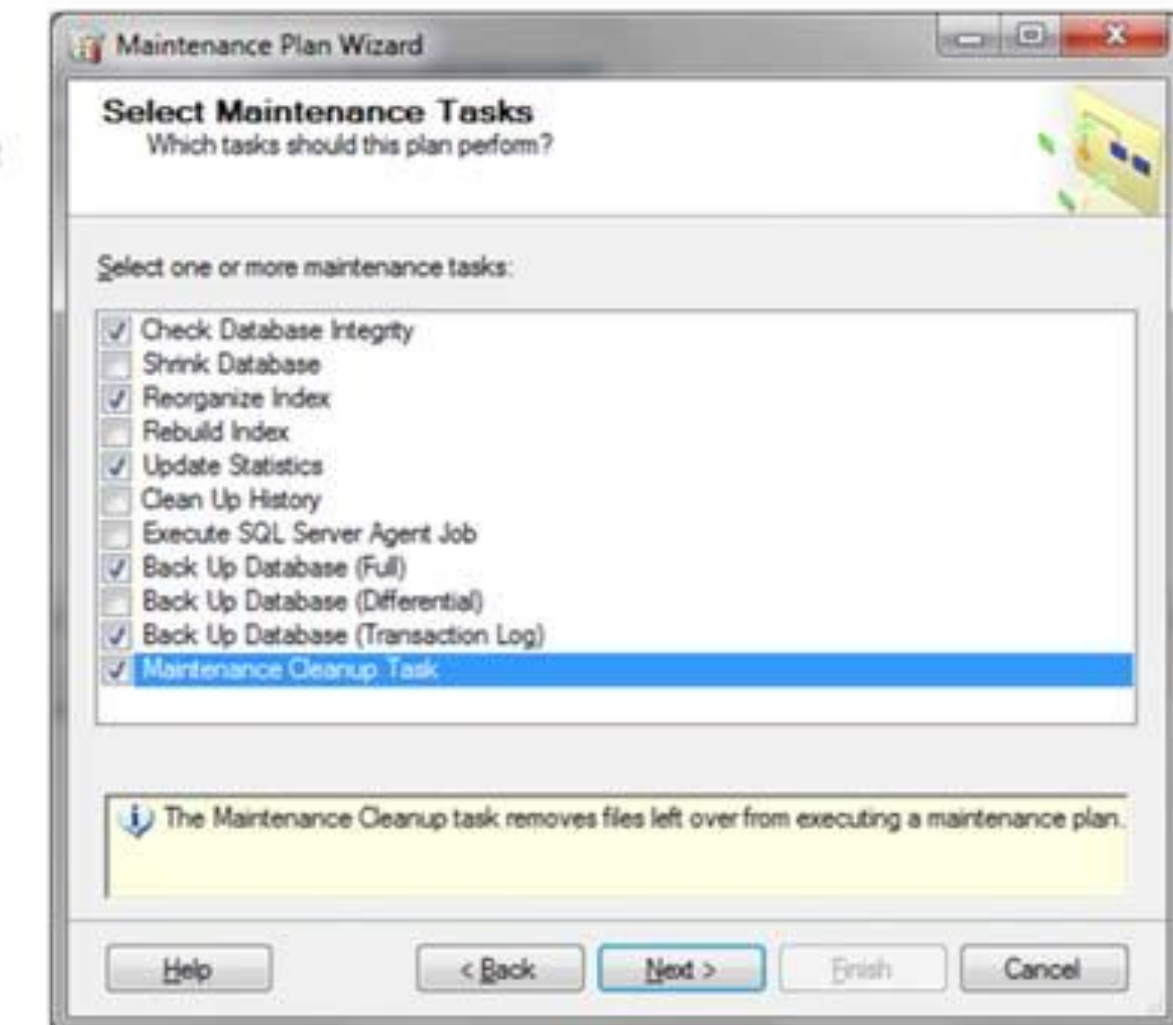
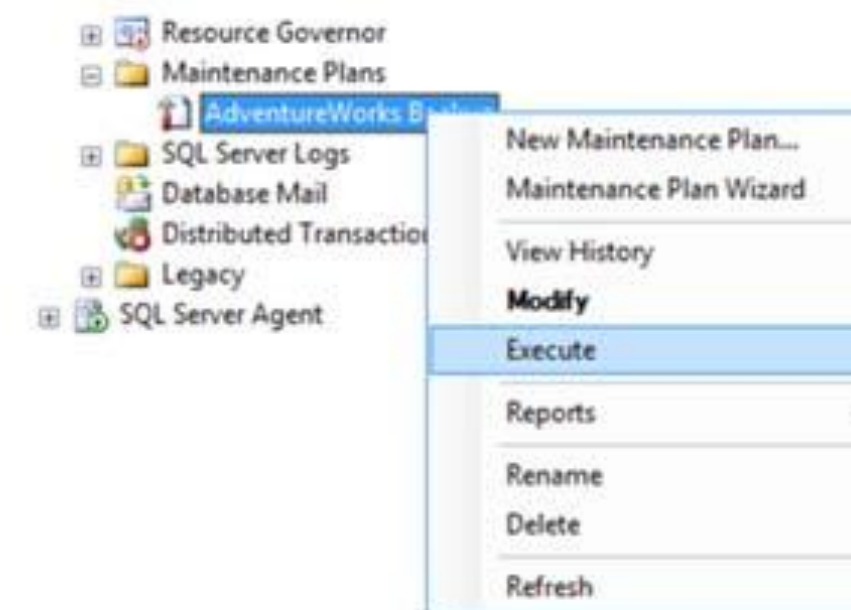

3. Otomasi Pemeliharaan Database Rutin

- Sebelumnya, ditunjukkan beberapa maintenance basis data secara manual yang dijalankan secara teratur.
- *Database Maintenance Plan Wizard* dapat digunakan untuk membuat *jobs* yang melakukan tugas-tugas maintenance database
- Output wizard dapat menjadi titik awal untuk membuat rencana maintenace, atau kita dapat membuat rencana maintenace dari nol

SQL Server Database Maintenance Plan

- Database Maintenance Plan Wizard dapat men-create SQL Server Agents yang melakukan tugas pemeliharaan basis data rutin dan menjadwalkan pekerjaan tersebut. untuk memastikan bahwa database telah dicadangkan secara teratur, berkinerja baik, dan memeriksa apakah terdapat ketidakkonsistenan

- Help database administrators to schedule core maintenance tasks
- Create one or more SQL Server Agent jobs
- Use SSIS to perform tasks
- Can be created manually rather than via the wizard



SQL Server Database Maintenance Plans

- Beberapa tugas maintenance database yang dapat dikerjakan antara lain:
 - Membuat cadangan file *database* dan log transaksi. *Database* dan *back up log* dapat disimpan untuk jangka waktu tertentu dan kemudian dihapus secara otomatis.
 - Menjalankan pekerjaan SQL Server Agent yang melakukan berbagai tindakan.
 - Memadatkan file data dengan menghapus halaman *database* kosong.
 - Melakukan pemeriksaan konsistensi internal data dan halaman data dalam *database* untuk memastikan bahwa masalah sistem atau perangkat lunak tidak merusak data.
 - Mengatur kembali informasi pada halaman data dan indeks halaman dengan *rebuild* indeks.
 - Memperbarui statistik indeks untuk memastikan *query optimizer* memiliki informasi terkini tentang distribusi nilai data dalam tabel.

Memonitor Database Maintenance Plans

- SQL Server database maintenance plans diimplementasikan menggunakan SQL Server Agent jobs yang menjalankan paket SQL Server Integration Services (SSIS), sehingga dapat dimonitor menggunakan Job Activity Monitor di SSMS
- Hasil yang digenerate oleh maintenance task ditulis kedalam tabel maintenance plan `dbo.sysmaintplan_log` and `dbo.sysmaintplan_log_detail` dalam database msdb. Melihatnya, dapat diquery langsung atau melalui log file viewer

- Real time monitoring through Job Activity Monitor
- Execution results stored in msdb and can also be
 - Written to a text file
 - Sent to an Operator
- Cleanup tasks are used to implement retention

Log File Viewer - (local)\prod2

Select logs

- ☐ Database Mail
- ☐ Job History
- ☒ Maintenance Plans
 - ☒ AdventureWorks Backup
 - ☐ Remote Maintenance Plans
- ☐ SQL Server Agent

Log file summary: No filter applied

Date	Plan Name	SubPlan N...	Task Name
13.01.2011 13:41:42	AdventureWorks Backup	Subplan_1	
13.01.2011 13:4...			Back Up Da
13.01.2011 13:38:07	AdventureWorks Backup	Subplan_1	

- Any question?

