

Microsoft SQL Server 2008

資料庫進階管理

台灣微軟 特約顧問

許致學



Systems Engineering
Systems Administration
Database Administration
Database Development



Microsoft®

SQL Server™ 2008 資料庫進階管理

- 企業級管理
- 高效能和高延展性
- 高可用度
- 企業級安全

企業級管理

★ SQL Server 2008 新功能

- ★ 中央管理伺服器
- ★ 以原則為基礎的管理
- Management Studio 管理報表
- ★ 資源管理員
- ★ 資料收集器

Microsoft®
SQL Server™ 2008

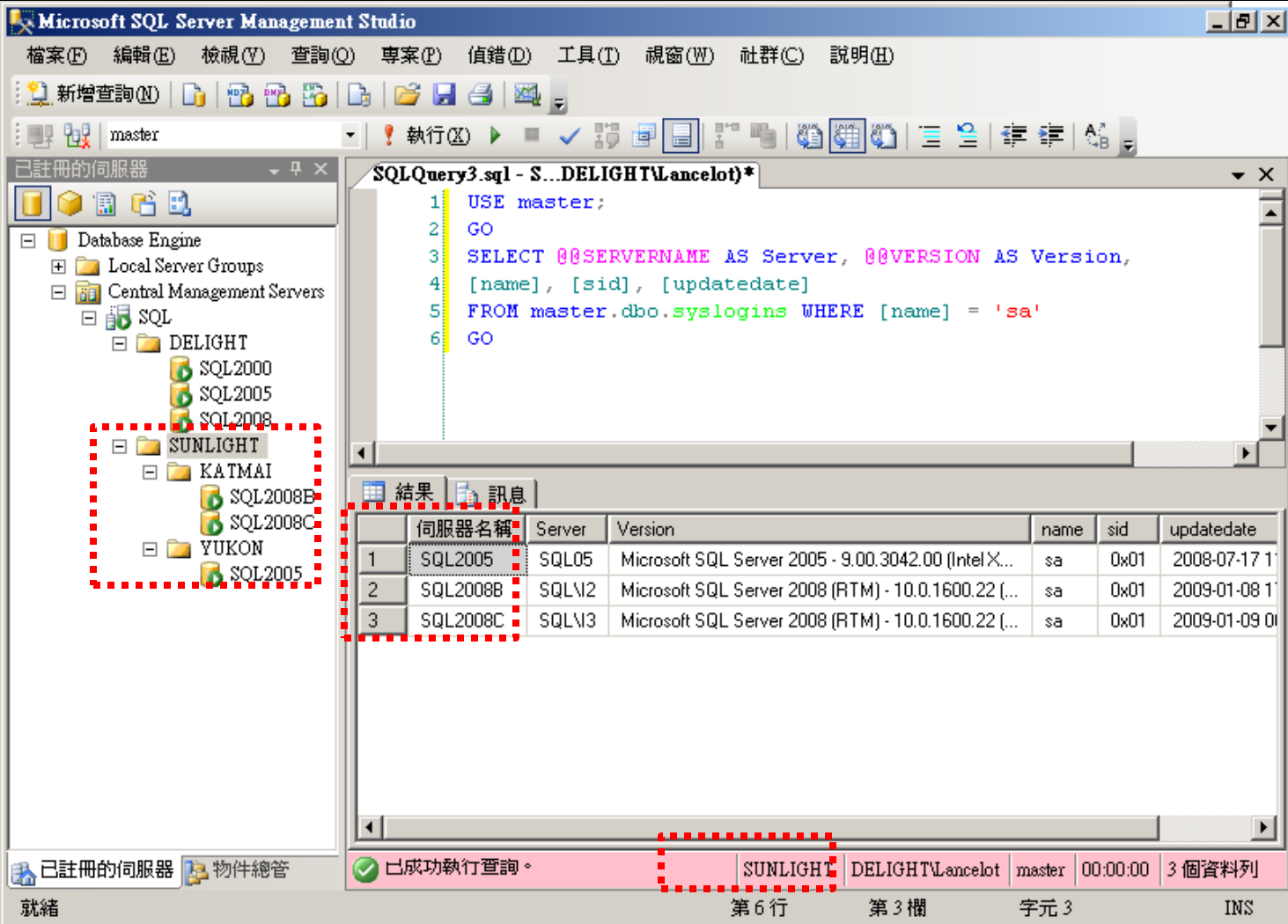
中央管理伺服器

- 中央管理伺服器
 - Central Management Servers
- 伺服器群組
- 多伺服器查詢
- 原則管理



階層式伺服器群組

執行多伺服器查詢



Microsoft SQL Server Management Studio

檔案(F) 編輯(E) 檢視(V) 查詢(Q) 專案(P) 偵錯(D) 工具(T) 視窗(W) 社群(C) 說明(H)

新增查詢(N)

master

已註冊的伺服器

- Database Engine
 - Local Server Groups
 - Central Management Servers
 - SQL
 - DELIGHT
 - SQL2000
 - SQL2005
 - SQL2008
 - SUNLIGHT
 - KATMAI
 - SQL2008B
 - SQL2008C
 - YUKON
 - SQL2005

SQLQuery3.sql - S...DELIGHT\lancelot)*

```
1 USE master;  
2 GO  
3 SELECT @@SERVERNAME AS Server, @@VERSION AS Version,  
4 [name], [sid], [updatedate]  
5 FROM master.dbo.syslogins WHERE [name] = 'sa'  
6 GO
```

結果 訊息

	伺服器名稱	Server	Version	name	sid	updatedate
1	SQL2005	SQL05	Microsoft SQL Server 2005 - 9.00.3042.00 (Intel X...	sa	0x01	2008-07-17 1
2	SQL2008B	SQLN2	Microsoft SQL Server 2008 (RTM) - 10.0.1600.22 (...)	sa	0x01	2009-01-08 1
3	SQL2008C	SQLN3	Microsoft SQL Server 2008 (RTM) - 10.0.1600.22 (...)	sa	0x01	2009-01-09 0

已註冊的伺服器 物件總管

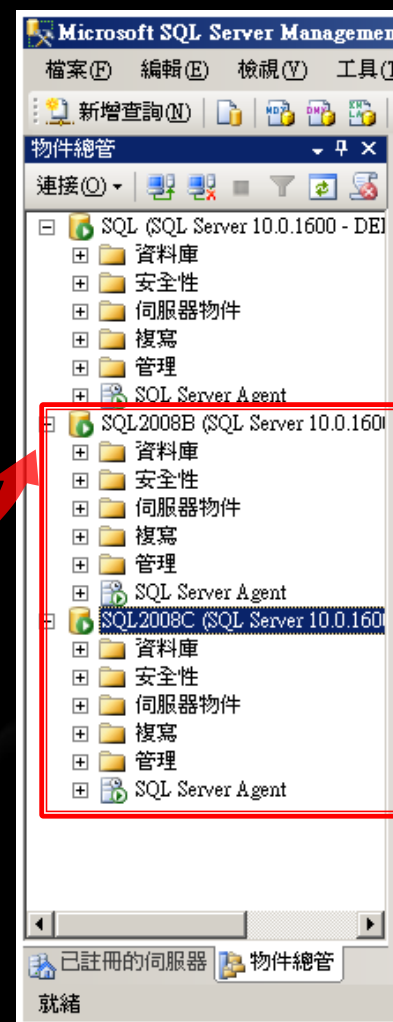
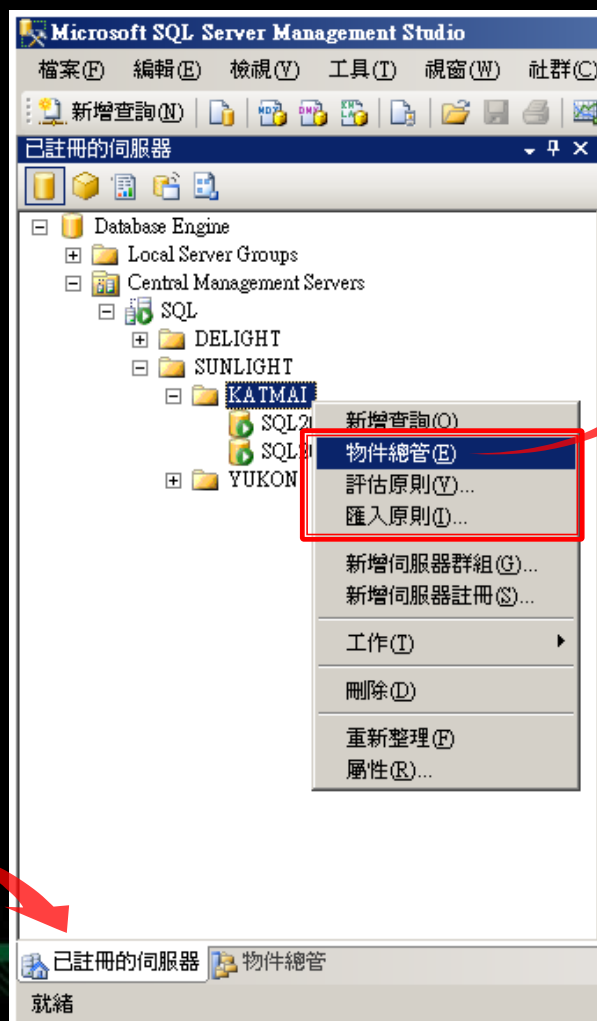
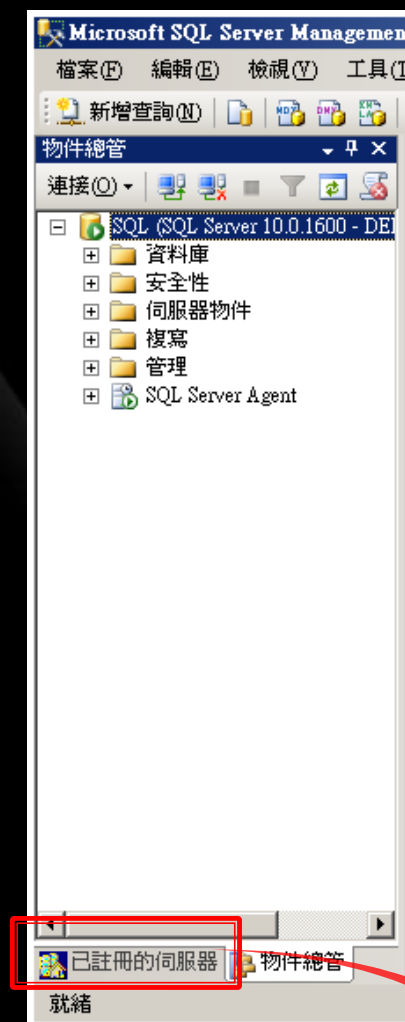
就緒

已成功執行查詢。

SUNLIGHT DELIGHT\lancelot master 00:00:00 3 個資料列

第 6 行 第 3 欄 字元 3 INS

自動連線多個伺服器



多伺服器查詢

一次查詢多個伺服器的 sa 登入帳戶最後一次更改密碼的日期

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the '已註冊的伺服器' (Registered Servers) tree with 'DELIGHT' selected. The central query editor shows a T-SQL query to retrieve the last password change date for the 'sa' login across three servers. The bottom pane shows the results of the query in a table format.

SQLQuery1.sql - DE...DELIGHT\lancelot*)*

```
1 USE master
2 GO
3 SELECT @@SERVERNAME AS Server, @@VERSION AS Version,
4 [name], [sid], [updateddate]
5 FROM master.dbo.syslogins WHERE [name] = 'sa'
6 GO
```

結果

	伺服器名稱	Server	Version	name	sid	updateddate
1	SQL2005	SQL05	Microsoft SQL Server 2005 - 9.00.3042.00 (Intel X...	sa	0x01	2008-07-17 1
2	SQL2000	SQL01	Microsoft SQL Server 2000 - 8.00.760 (Intel X86) ...	sa	0x01	2008-07-31 1
3	SQL2008	SQL\12	Microsoft SQL Server 2008 (RTM) - 10.0.1600.22 (...)	sa	0x01	2009-01-08 1

已註冊的伺服器 物件總管

就緒

已成功執行查詢。 DELIGHT DELIGHT\lancelot master 00:00:00 3 個資料列

第 6 行 第 3 欄 字元 3 INS

本機伺服器群組

編輯伺服器註冊屬性

一般 連接屬性

登入

輸入伺服器名稱，或從下拉式清單中選擇。

伺服器類型(T): Database Engine

伺服器名稱(S): SQLN3

驗證(A): SQL Server 驗證

登入(L): sa

密碼(P): *****

☒ 記住密碼(M)

已註冊的伺服器

您可以使用新名稱名稱。

已註冊的伺服器4

sqln3

已註冊的伺服器描述(D):

測試(E) 儲存(Y) 取消 說明

RegSrvr.xml 註冊資訊：SQL Server 驗證

```
<RegisteredServers:ConnectionStringWithEncryptedPassword
  type="string">server=SQL\I3;uid=sa;password=AQAAANCMnd8BFdERjHoAwE/CI+sBAAAFq2P1BikC
  size=4096;multipleactiveresultsets=false</RegisteredServers:ConnectionStringWithEncryptedPassword>
<RegisteredServers:CredentialPersistenceType
```

RegSrvr.xml 註冊資訊：Windows 驗證

```
<RegisteredServers:ConnectionStringWithEncryptedPassword
  type="string">server=SQL;trusted_connection=true;pooling=false;multipleactiveresultsets=false</
<RegisteredServers:CredentialPersistenceType
  type="CredentialPersistenceType">PersistLoginName</RegisteredServers:CredentialPersistenceType>
```


中央管理伺服器 vs. 本機伺服器群組

中央管理伺服器	本機伺服器群組
須先註冊一個SQL Server 2008的中央管理伺服器	可以直接註冊SQL Server 2005和SQL Server 2000
註冊的資訊儲存在系統資料庫msdb	註冊的資訊儲存在RegSrvr.xml檔案
僅支援Windows驗證	以支援SQL Server驗證與Windows驗證

RegSrvr.xml檔案在作業系統Windows XP和Windows Server 2003預設的路徑是：C:\Documents and Settings\<UserName>\Application Data\Microsoft\Microsoft SQL Server\100\Tools\Shell；而作業系統Windows Vista和Windows Server 2008預設的路徑是：C:\Users\<UserName>\Application Data\Microsoft\Microsoft SQL Server\100\Tools\Shell

DEMO

中央管理伺服器

以原則為基礎的管理

原則自動套用至
整個企業

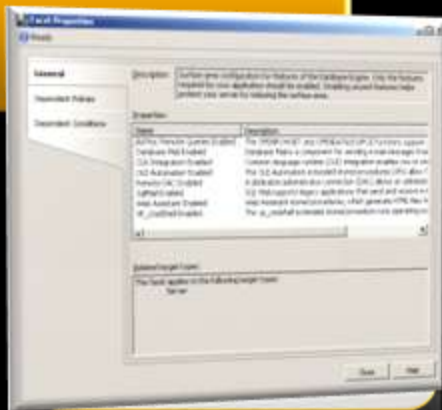


集中定義原則



以原則為基礎的管理

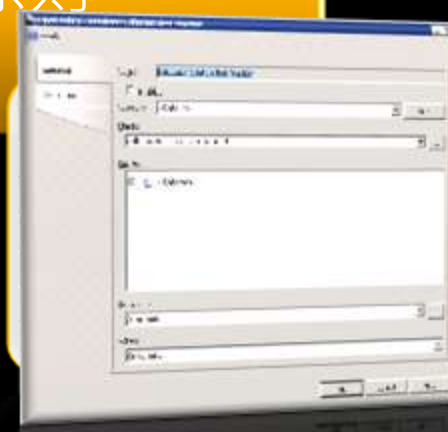
Facets



條件



原則



受管理的目標



類別目錄

原則的評估模式

手動

視需要

- 直接指定時才會進行評估

自動

變更時：避免

- 發生相關變更時，使用 **DDL觸發程序** 來防止原則被違反

變更時：僅限記錄

- 發生相關變更時，使用 **事件通知** 來評估原則，並且記錄原則被違反

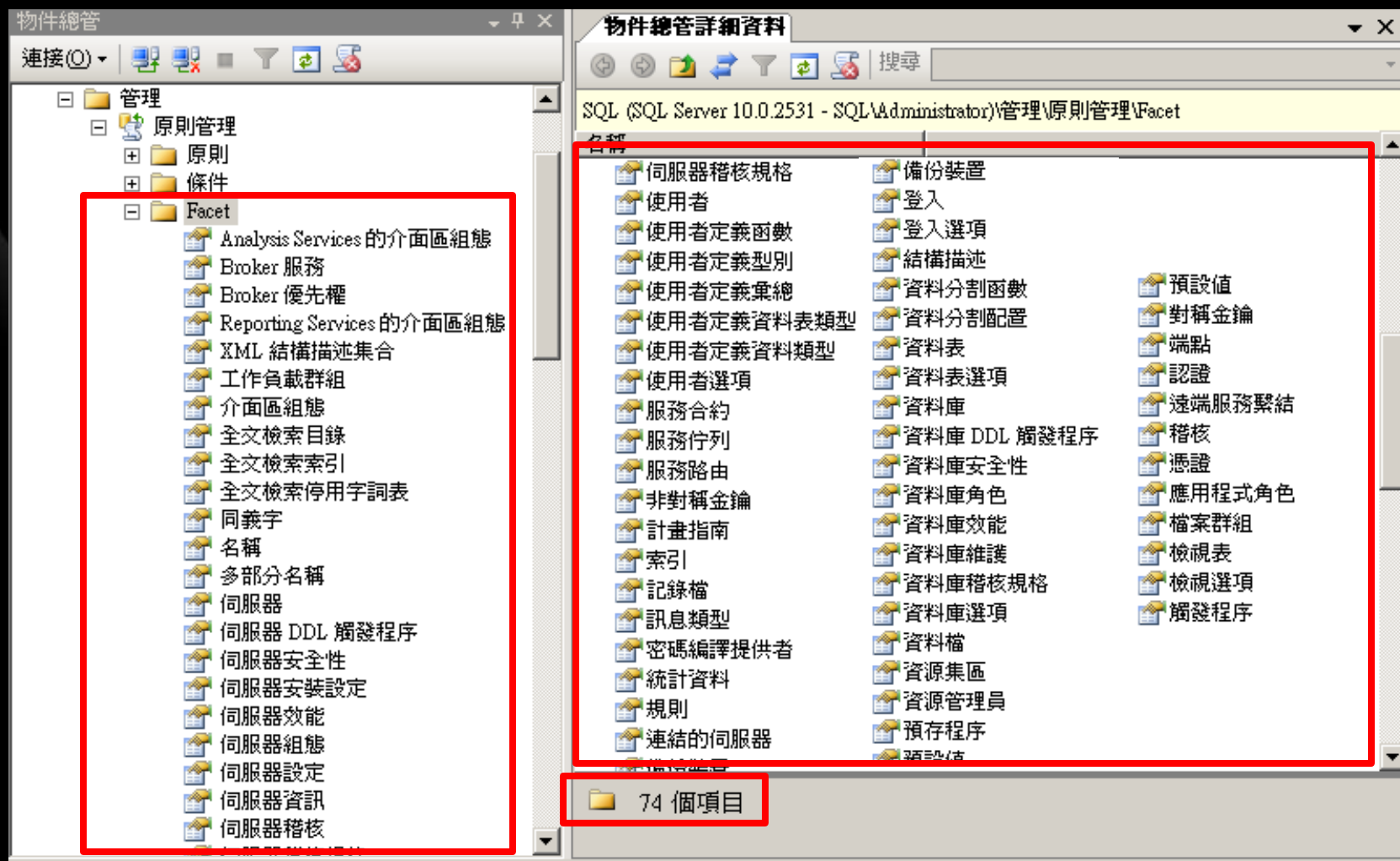
安排程時間

- 使用SQL Server Agent作業排程定期評估原則，並且記錄原則被違反

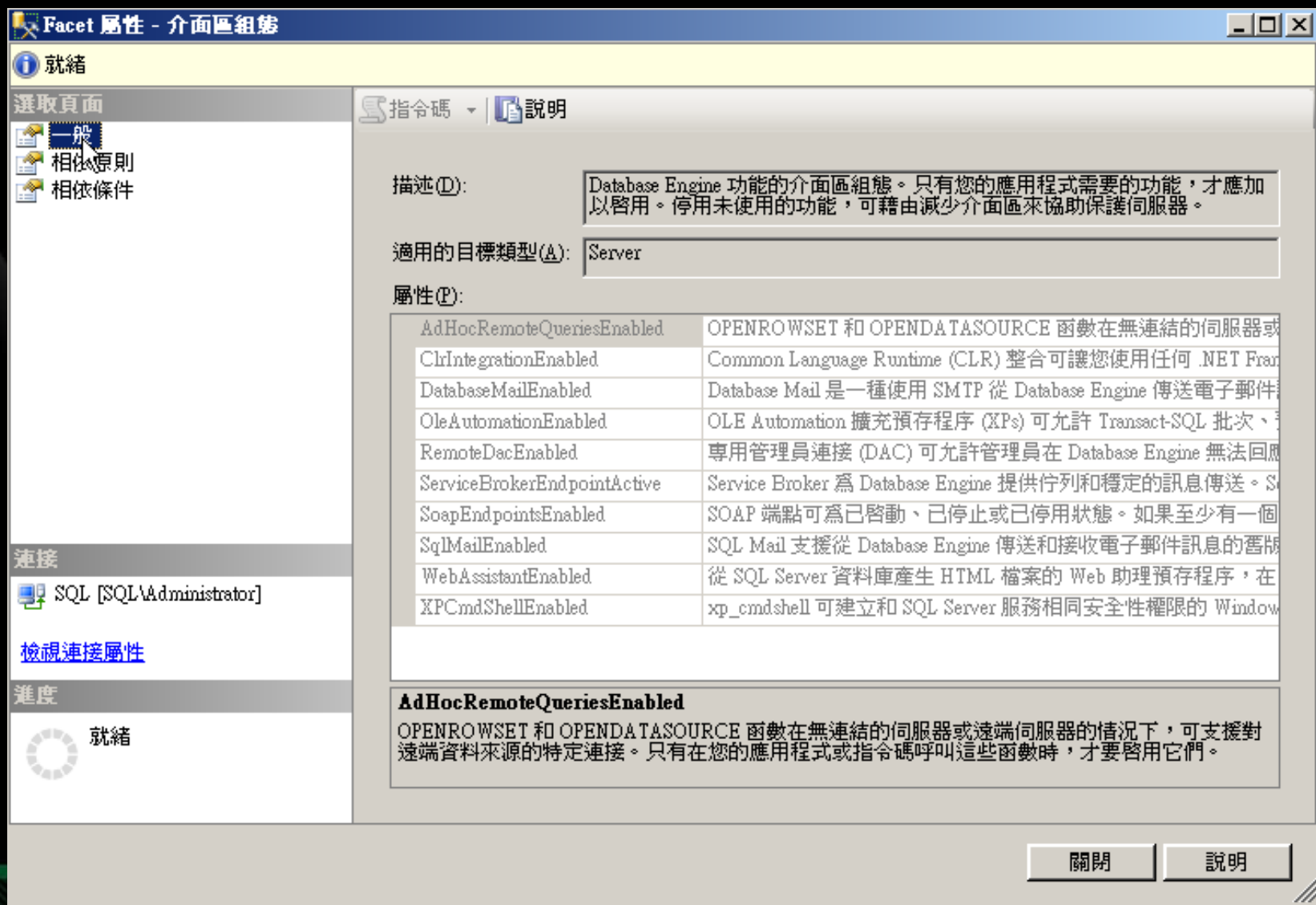
可支援前版

僅支援 SQL Server 2008

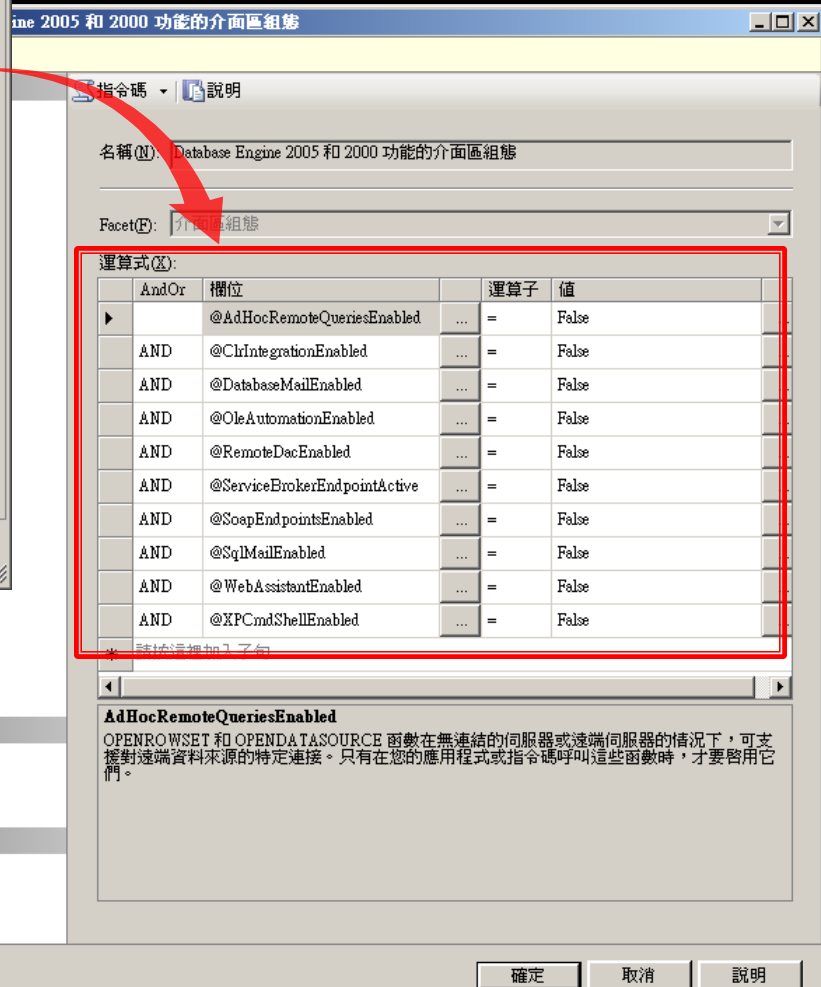
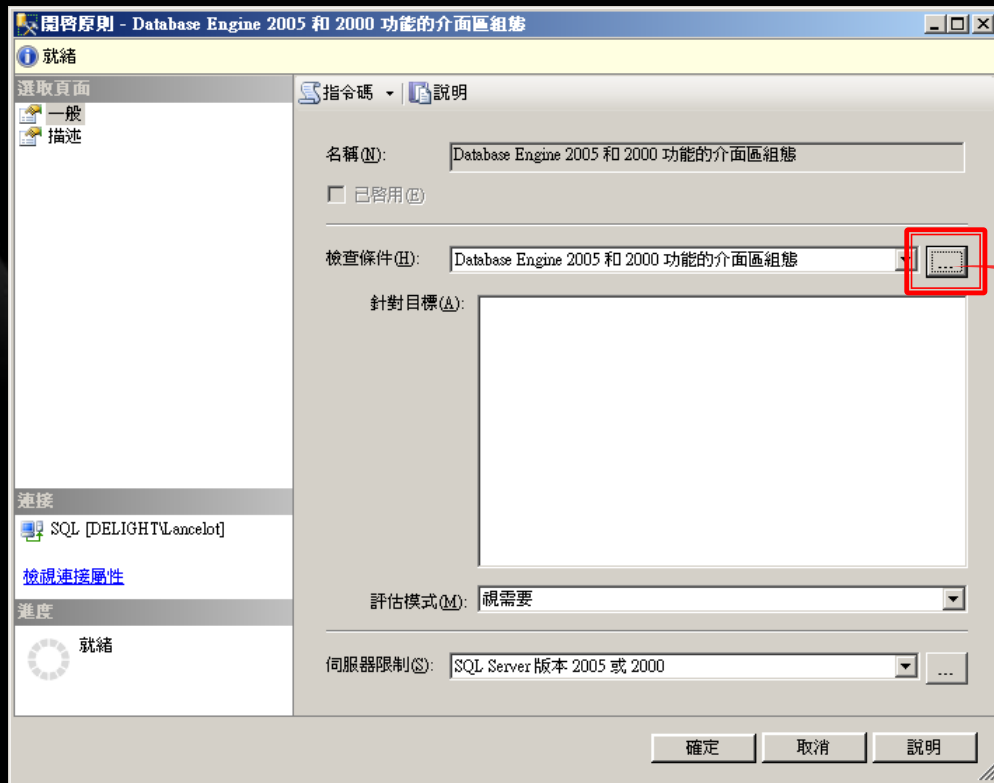
Facet - 組成原則的邏輯屬性



Facet 屬性 - 介面區組態



原則與條件



檢視原則評估結果

評估原則 - 資料庫PolicyTest結構描述安全性原則

就緒

選取頁面

- 原則選取
- 評估結果

指令碼 說明

結果:
選取要在下表中查看詳細資料的原則(A)

原則	訊息
資料庫PolicyTest結構描述安全性原則	

目標詳細資料(I):

伺服器	目標	詳細資料
SQL	SQLSERVER:SQL\SQLDEFAULT\T\Databases\PolicyTest\StoredProc...	檢視...
SQL	SQLSERVER:SQL\SQLDEFAULT\T\Databases\PolicyTest\Tables\dbo.T2	檢視...
SQL	SQLSERVER:SQL\SQLDEFAULT\T\Databases\PolicyTest\Tables\Producti...	檢視...
SQL	SQLSERVER:SQL\SQLDEFAULT\T\Databases\PolicyTest\UserDefined Fu...	檢視...

連接

SQL [DELIGHT\ancelot]

檢視連接屬性

進度

完成

結果詳細檢視

AndOr	結果	欄位	運算子	預期的值	實際值
	✗	@Schema	!=	'dbo'	'dbo'

原則描述:

檢查資料庫PolicyTest的資料表、檢視表、預存程序、使用者自訂函數是否使用結構描述名稱dbo

其他說明: 資料庫物件的結構描述名稱不可使用 dbo: <http://msdn.microsoft.com/zh-tw/library/ms190387.aspx>

關閉 說明

物件總管

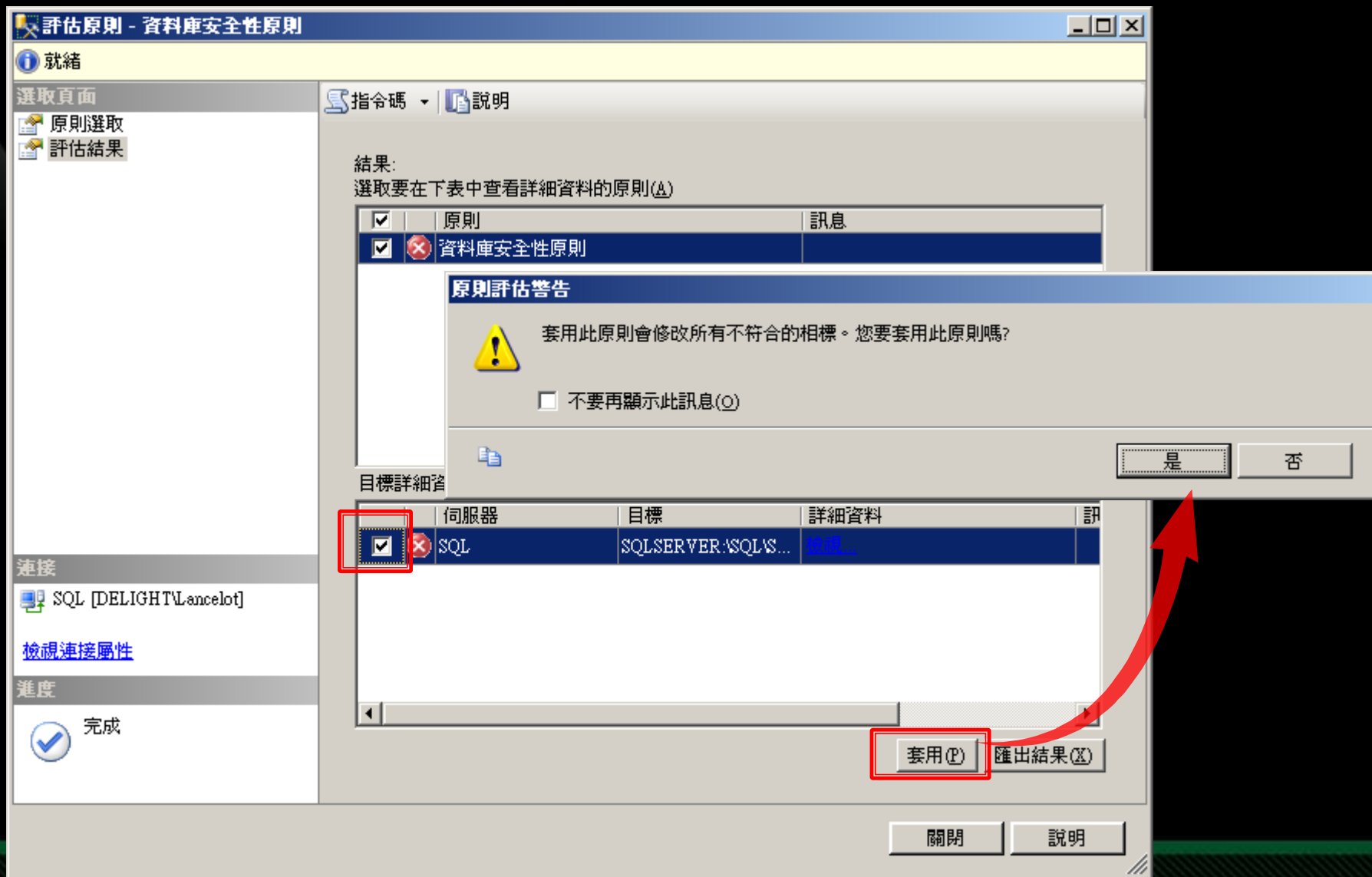
連接(O)

SQL (SQL Server 10.0.1600 - DEL)

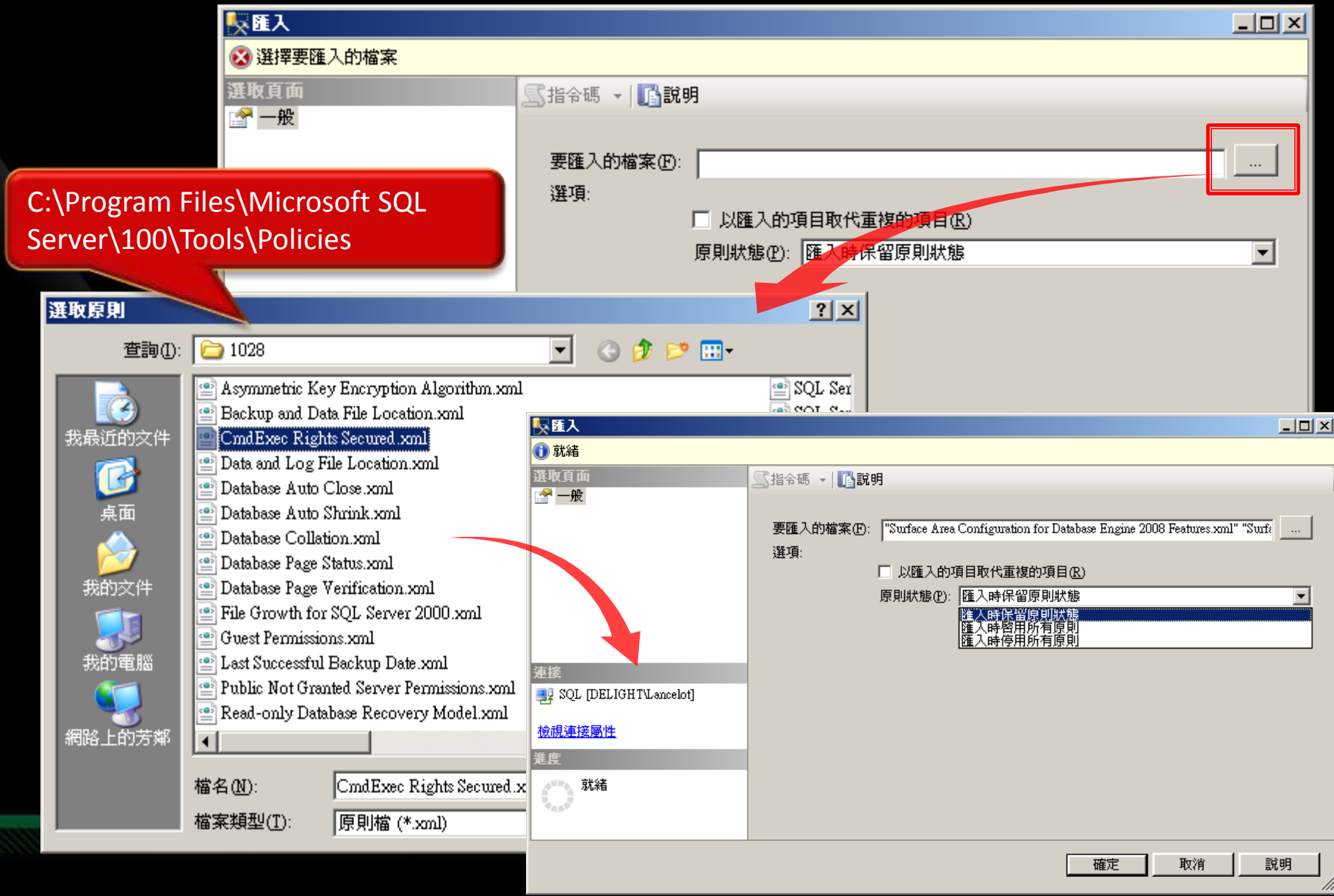
- 資料庫
 - 系統資料庫
 - 資料庫快照集
 - Northwind
 - PolicyTest
 - 資料庫圖表
 - 資料表
 - 系統資料表
 - dbo.T2
 - Production.T1
 - 檢視
 - 同義字
 - 可程式性
 - Service Broker
 - 儲存體
 - 安全性
 - ReportServer
 - ReportServerTempDB
- 安全性
- 伺服器物件
- 複寫
- 管理

已註冊的伺服器 物件總管

原則評估後進行套用



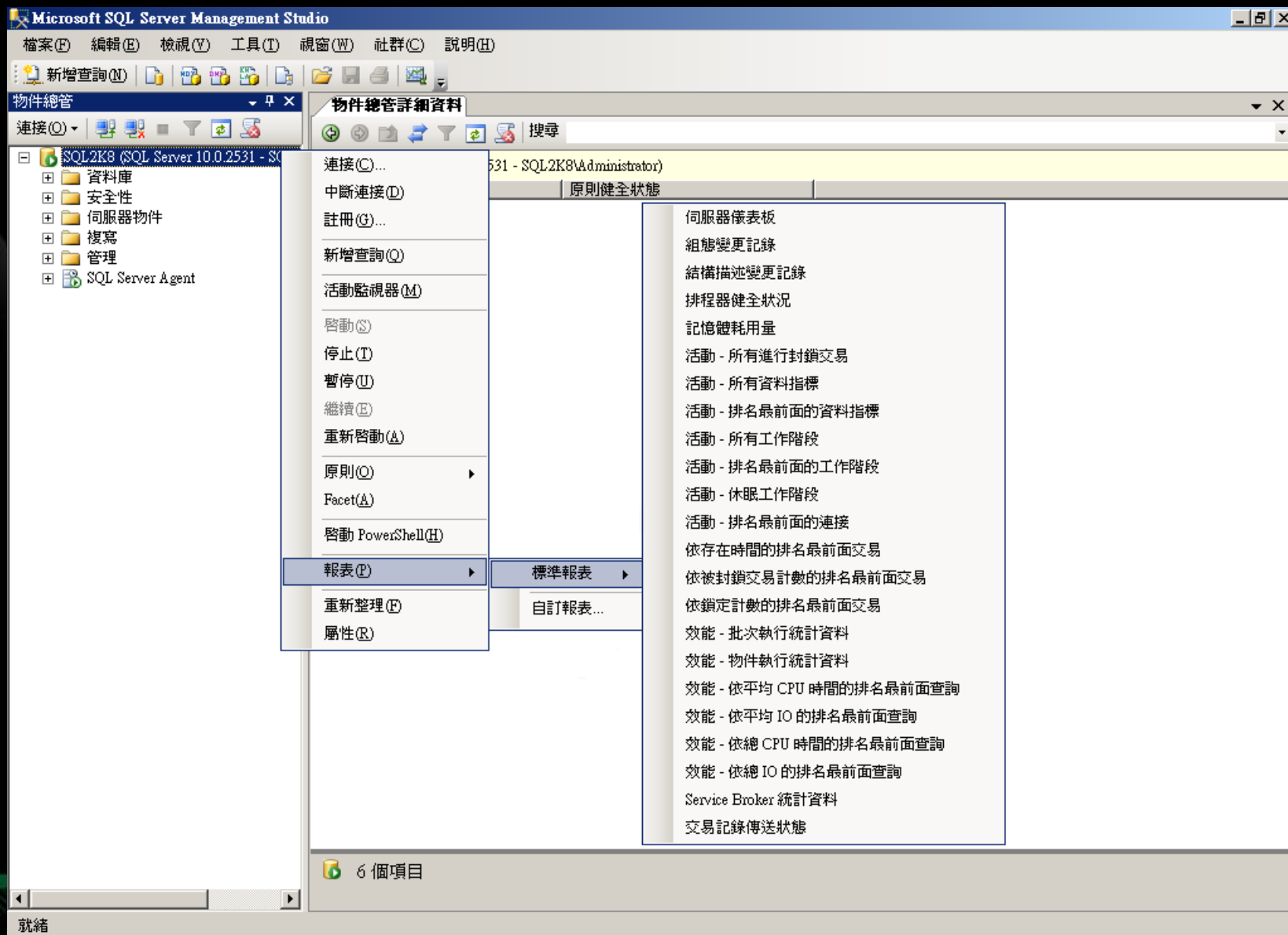
匯入內建的原則檔案 (Best Practices)



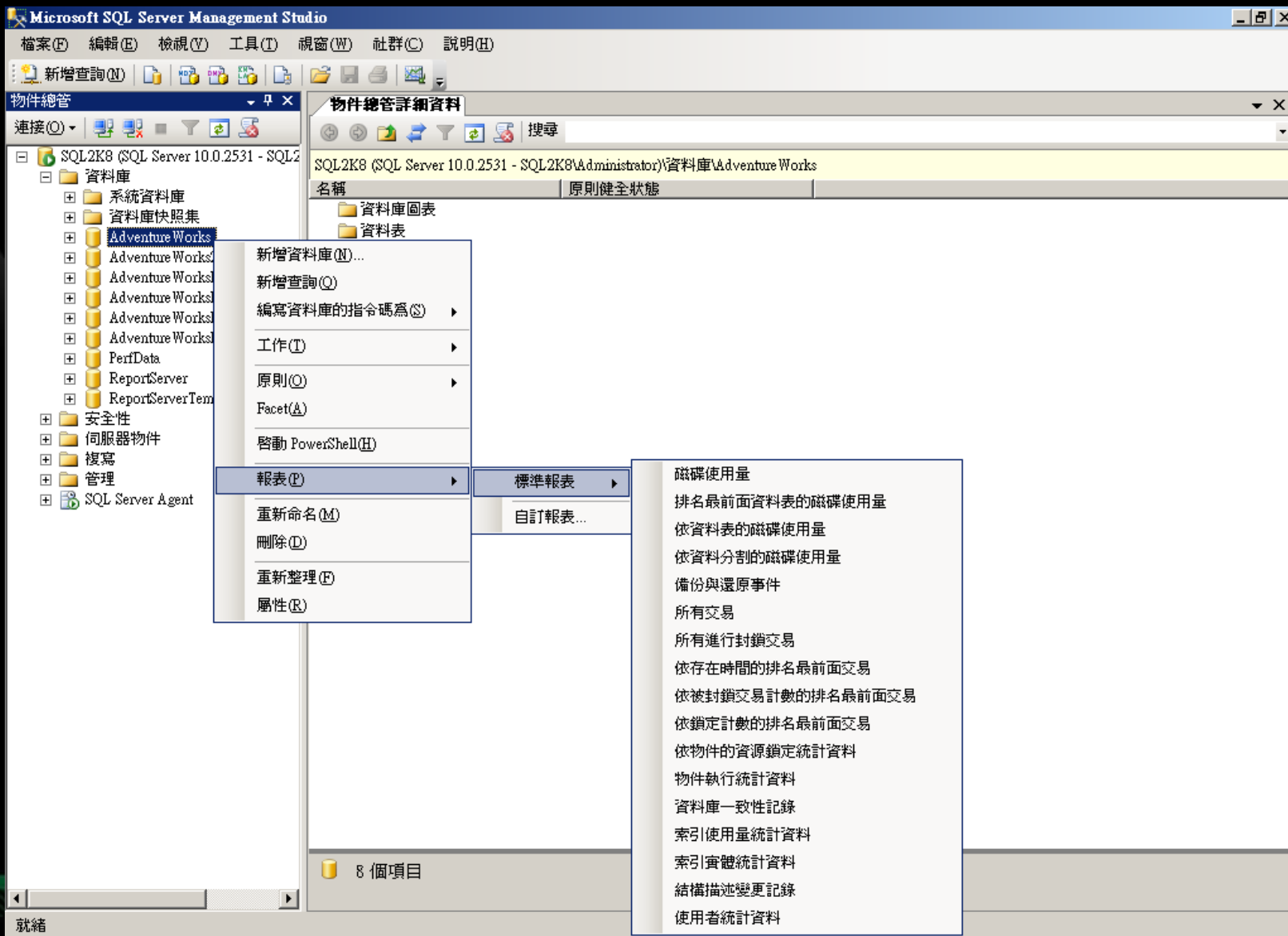
DEMO

以原則為基礎的管理

Management Studio 報表 (伺服器)



Management Studio 報表 (資料庫)



Performance Dashboard Reports

- 一組遵循 Reporting Services 定義的客製化報表
- 整合 SQL Server Management Studio
 - Report Viewer 控制項
- 針對 SQL Server 2005 SP2 之後的效能監控
- 藉由存取 SQL Server 2005 所提供的系統物件，呈現 SQL Server 2005 執行個體當下運作的情形，藉以分析效能問題

Performance Dashboard Reports

● SQL Server 2005 免費下載

- <http://www.microsoft.com/downloads/details.aspx?familyid=1d3a4a0d-7e0c-4730-8204-e419218c1efc&displaylang=en>

● SQL Server 2008 需稍加修改才可使用

- 下載 PerfDashboardReports.zip
 - <http://blogs.technet.com/rob/archive/2009/02/18/performance-dashboard-reports-for-sql-server-2008.aspx>
- 更改預設安裝路徑
 - C:\Program Files\Microsoft SQL Server\100\Tools\PerformanceDashboard
 - C:\Program Files (x86)\Microsoft SQL Server\100\Tools\PerformanceDashboard
- 將下列二個檔案取代
 - Performance_dashboard_main.rdl 和 Setup.sql
 - 執行 Setup.sql 前，建議修改 `datediff(ms, ...)` 變更為 `datediff(s, ...)`

Performance Dashboard Reports 之應用

硬體資源的情況

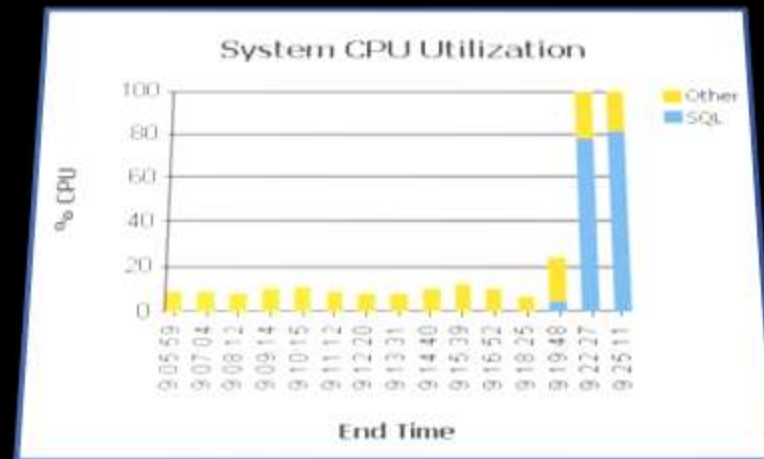
- 檢視指令所使用 CPU 和 磁碟 I/O

封鎖

- 當下被迫等待的指令

索引

- 建議須新增的索引以提升效能



的

Recent CPU Consumption

Report Time: 06/02/2007 9:28:43 AM

Summary: Shows the top 10 requests that have the highest aggregate CPU usage. This is based on CPU consumed by current requests and estimated CPU usage for each request under execution. This does not include CPU for any requests which have already logged out.

Session ID	Active Requests	Estimated Request CPU (ms)	Session CPU (ms)	Login Time	Last Request Start Time	Last Request End Time	Program Name	Login Name
11	5	82,246,000	82,246,000	06/02/2007 9:22:41 AM	06/02/2007 9:22:59 AM	06/02/2007 9:22:41 AM		WMS0000001\Administrator
	Request ID	Estimated Request CPU	Request CPU (ms)	Request Start Time	Request Status	Comment		
10	5	76,000,000	76,000,000	06/02/2007 9:23:04 AM	06/02/2007 9:23:04 AM	06/02/2007 9:23:04 AM	Microsoft SQL Server Management Studio - 8026	WMS0000001\Administrator
10	5	6,7,000,000	6,7,000,000	06/02/2007 9:22:20 AM	06/02/2007 9:22:20 AM	06/02/2007 9:22:20 AM	Microsoft SQL Server Management Studio - 8026	WMS0000001\Administrator

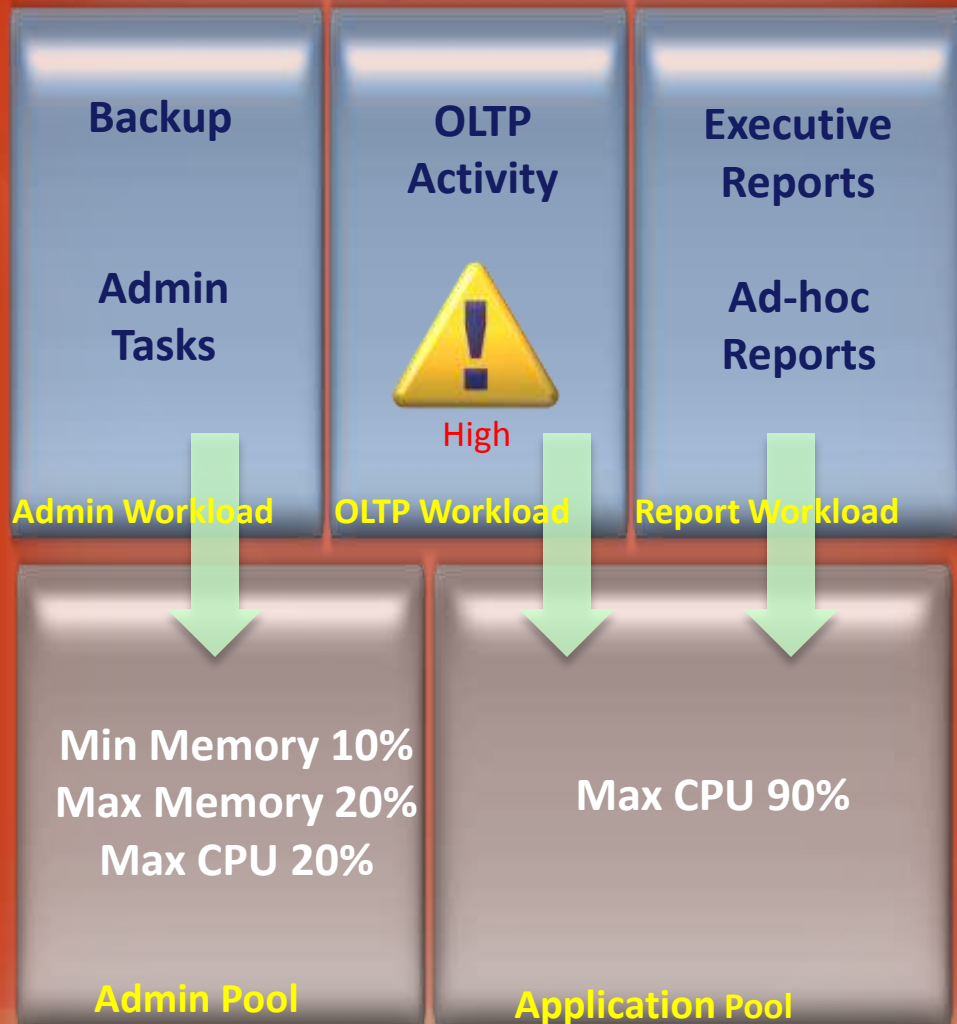
The above table lists the top 10 requests that have the highest aggregate CPU usage that are known to have been executed during the time window.

Query Number	Query	Executes	Plan Generation	Plan Cached	Last Executed	CPU (ms)		
						Total	Plan	Last
1	SELECT * FROM [WMS0000001].[dbo].[Inventory] WHERE [Location] = 'A' AND [Status] = 'A' AND [Date] <= '2007-06-01' ORDER BY [Location], [Status], [Date]	1	06/02/2007 9:47:01 AM	06/02/2007 9:46:59 AM	911,163	285,502		
2	SELECT * FROM [WMS0000001].[dbo].[Inventory] WHERE [Location] = 'A' AND [Status] = 'A' AND [Date] <= '2007-06-01' ORDER BY [Location], [Status], [Date]	1	06/02/2007 9:41:27 AM	06/02/2007 9:40:41 AM	329,532	144,146		

3 資源管理員

企業版

SQL Server



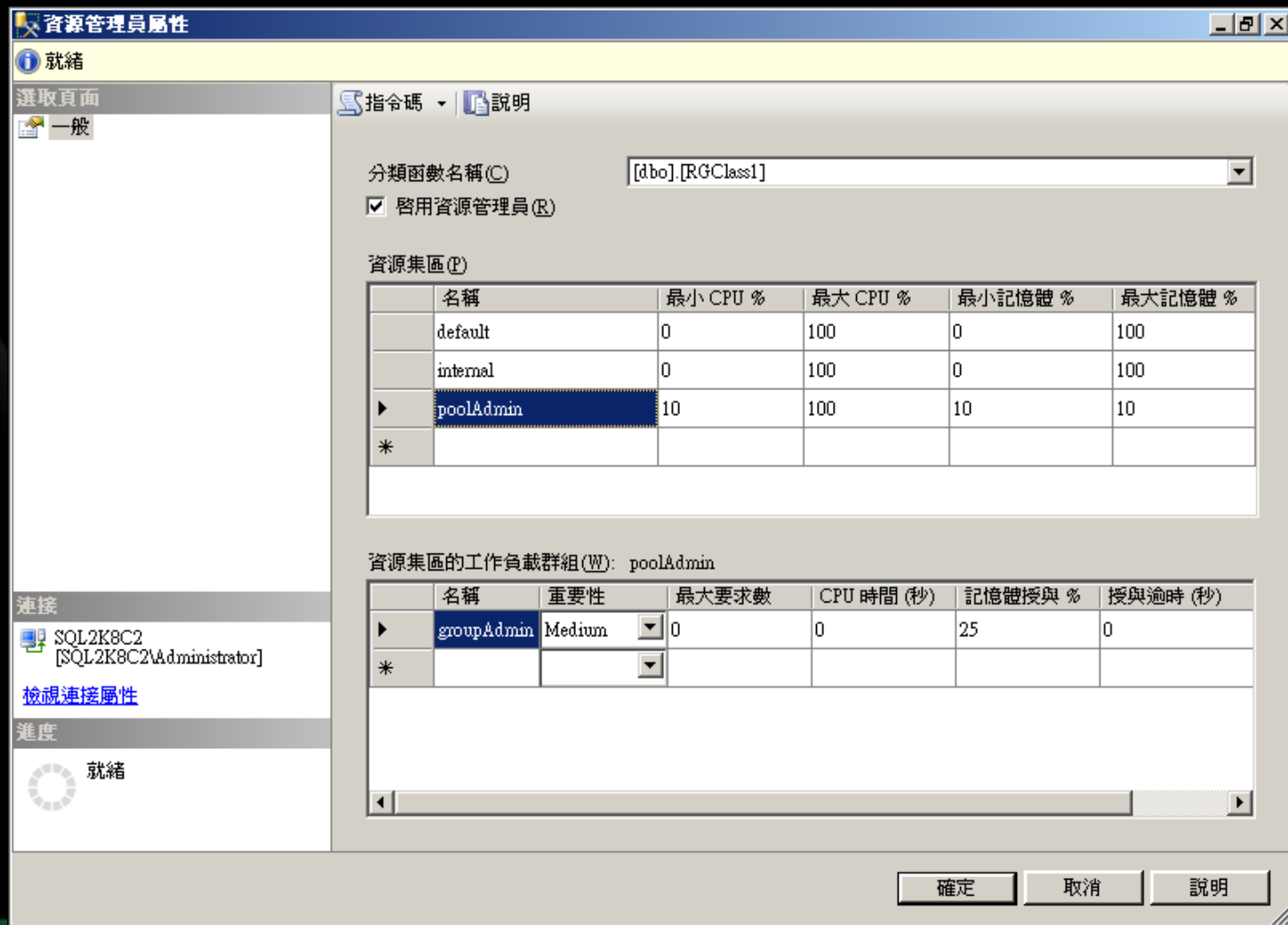
可依條件區分工作負荷

- 應用程式
- 登入帳號
- 資料庫
- ...

工作負荷限制條件

- 記憶體授與 %
- CPU 時間
- 授與逾時
- 最大要求數

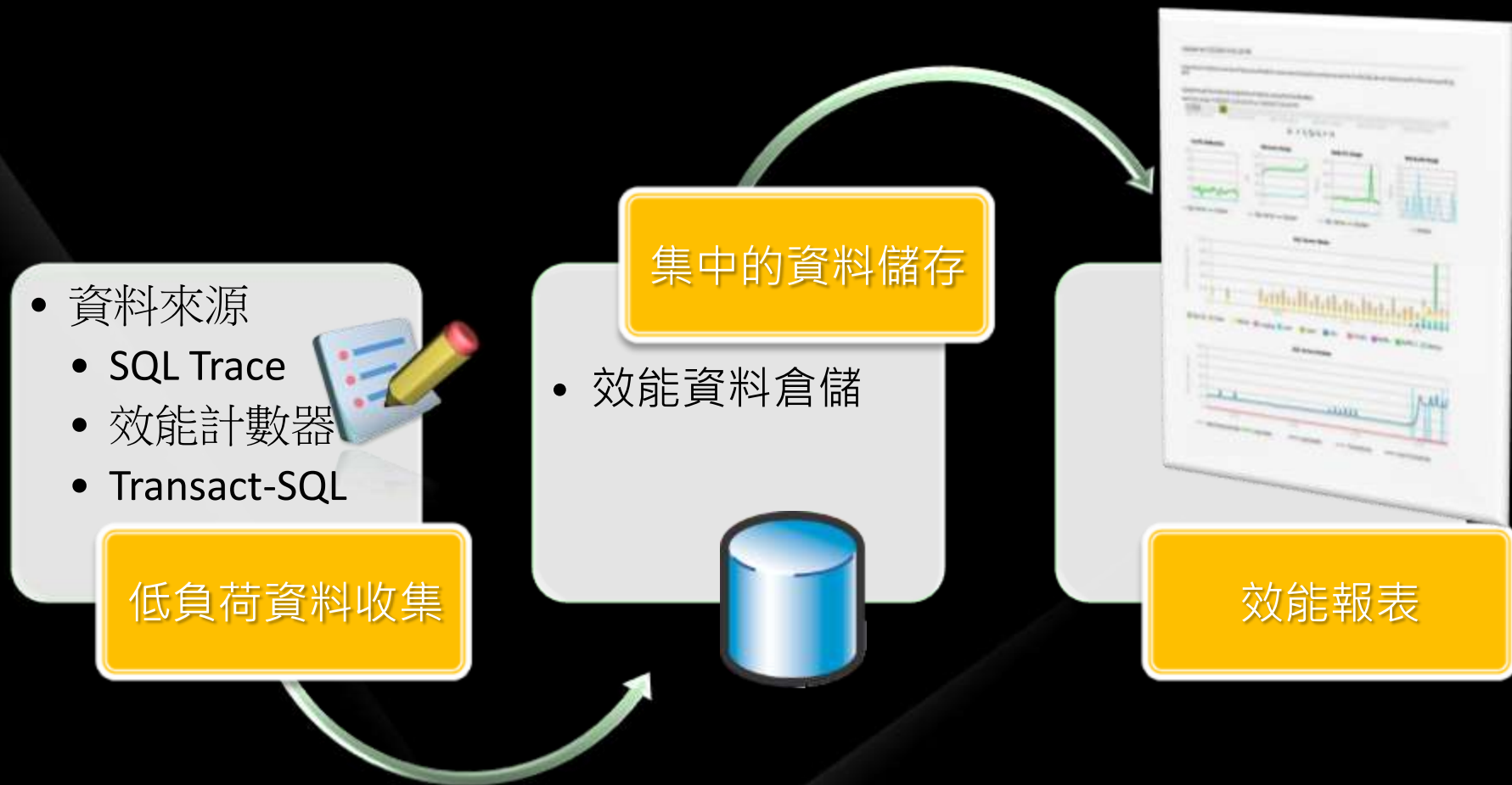
資源管理員



DEMO

資源管理員

資料收集器



設定管理資料倉儲精靈

設定管理資料倉儲精靈

選取組態工作
選取工作以設定管理資料倉儲或資料收集組。

選取要執行的工作：

☒ 建立或升級管理資料倉儲(C)
建立或升級用於儲存資料收集組結果的管理資料倉儲。若要設定此資料收集，請再執行一次這個精靈，並且選取 [設定資料收集]。

☐ 設定資料收集(S)

設定此執行個體來開始收集資料到現有的管理資料倉儲。

說明(H) < 上一步(B) 下一步(N) > 完成(F) >>

設定管理資料倉儲精靈

設定管理資料倉儲儲存體
選擇要用做管理資料倉儲的資料庫，以及在設定資料收集時目標伺服器上的快取目錄位置。

選取要主控管理資料倉儲的伺服器及資料庫。

伺服器名稱(E): SQL

資料庫名稱(D): PerfData 新增(W)

說明(H) < 上一步(B) 下一步(N) > 完成(F) >> 取消

設定管理資料倉儲精靈

設定管理資料倉儲精靈

選取組態工作
選取工作以設定管理資料倉儲或資料收集組。

選取要執行的工作：

☐ 建立或升級管理資料倉儲(C)
建立或升級用於儲存資料收集組結果的管理資料倉儲。若要設定此資料收集，請再執行一次這個精靈，並且選取 [設定資料收集]。

☒ 設定資料收集(S)

設定此執行個體來開始收集資料到現有的管理資料倉儲。

說明(H) < 上一步(B) 下一步(N) > 完成(F) >>

設定管理資料倉儲精靈

設定管理資料倉儲儲存體
選擇要用做管理資料倉儲的資料庫，以及在設定資料收集時目標伺服器上的快取目錄位置。

選取要當做管理資料倉儲主機的伺服器及資料庫。

伺服器名稱(E): SQL ...

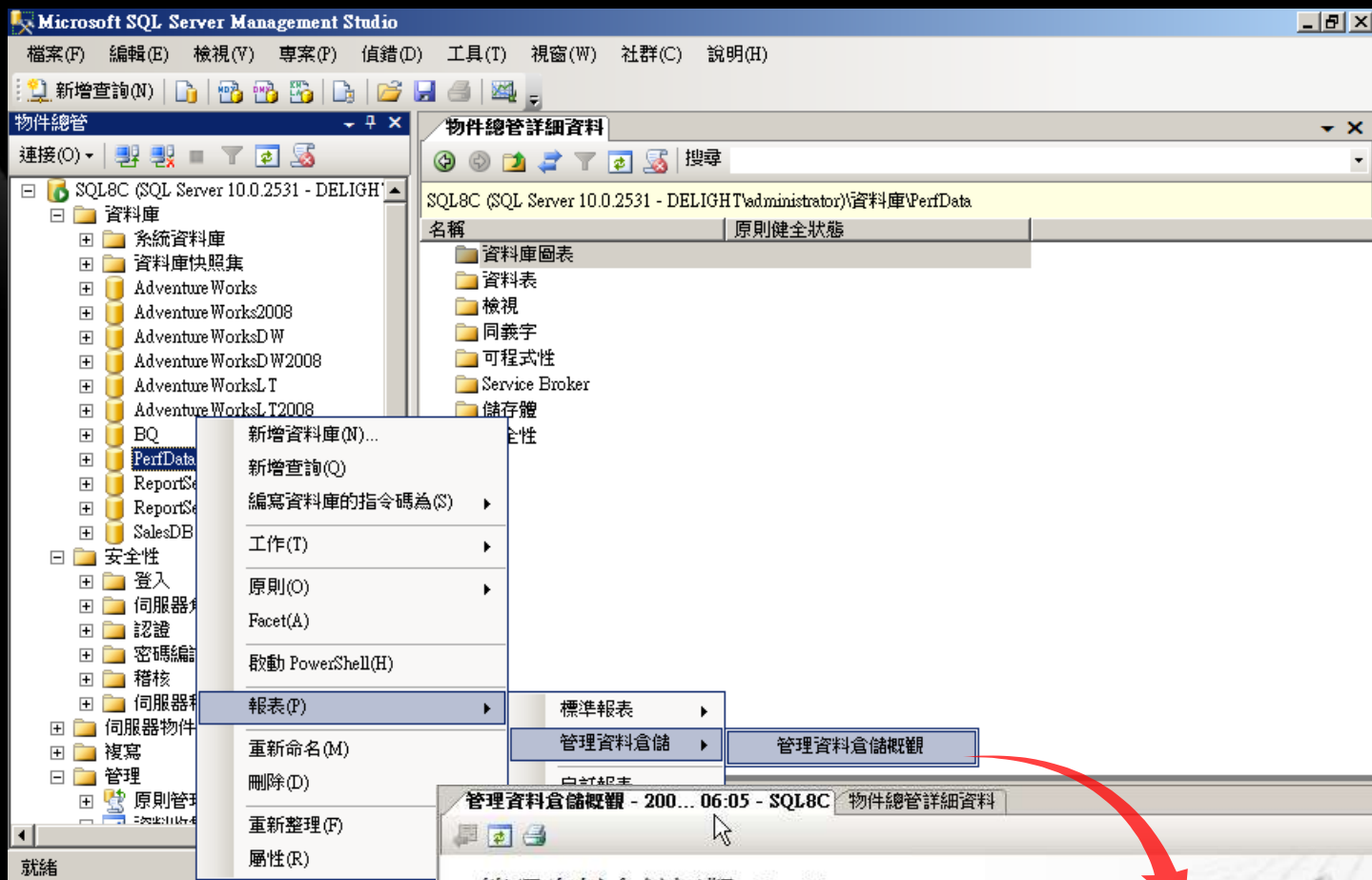
資料庫名稱(D): PerfData

輸入要將收集到之資料上載至管理資料倉儲之前，在本機快取的位置。空白的值將會使用收集器處理序的 TEMP 目錄。

快取目錄(C): C:\PerfLogs ...

說明(H) < 上一步(B) 下一步(N) > 完成(F) >> 取消

檢視管理資料倉儲報表



管理資料倉儲概觀 - 200... 06:05 - SQL8C

物件總管詳細資料

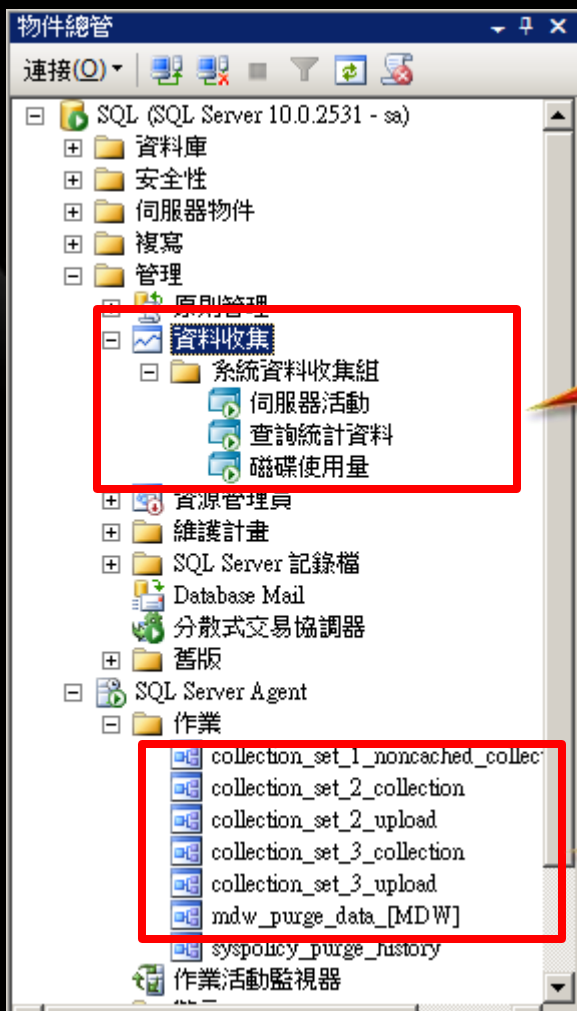
管理資料倉儲概觀: PerfData

在 SQL8C 的 2009/10/13 下午 06:05:53

此報表提供將資料記錄在此管理資料倉儲中之資料收集器執行個體的概觀。請按一下上次快照上載時間的超連結，顯示相關系統收集組的報表。

系統收集組的上次快照上載時間			
執行個體名稱 :	伺服器活動 :	查詢統計資料 :	磁碟使用量 :
SQL8C	2009/10/13 下午 06:00:45	2009/10/13 下午 06:00:45	2009/10/13 下午 06:03:11

資料收集器自動化作業



系統資料收集組：

1. 伺服器活動
2. 查詢統計資料
3. 磁碟使用量

資料收集、
載入、清除

DEMO

資料收集器

高效能和高延展性

★ SQL Server 2008 新功能

★ SQL Server 2008 增強功能

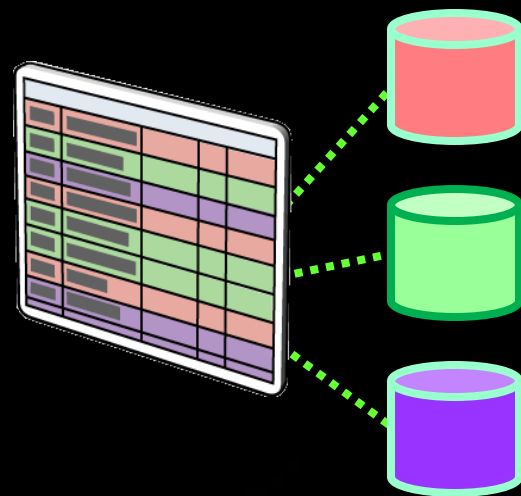
- ★ 資源管理員
- ★ 資料分割
- ★ 資料壓縮
- ★ 可擴充共用資料庫
- ★ 計畫指南

- ★ 疏鬆資料行
- ★ 篩選索引
- 索引檢視表
- 分散式分割檢視表

Microsoft®
SQL Server™ 2008

資料分割

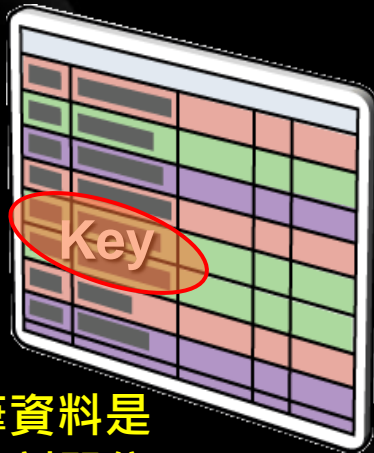
- 支援資料表與索引的分割
 - 儲存至不同的檔案群組
- 將單一資料表 / 索引區分成不同儲存體
 - 使用 Range 分割技術進行資料分組
 - 歷史資料與線上資料管理更容易
 - 提升資料庫備份與還原的效率
 - 提升資料刪除與大量資料載入的效率
 - 減少歷史資料索引離散與重整的狀況
 - 簡化應用程式開發與資料庫管理



資料分割函數和資料分割配置

資料分割配置指定於
不同的檔案群組

資料表

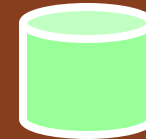


每一筆資料是
基本分割單位

資料分割的依據
如時間、序號...

資料分割
函數

資料分割配置



檔案群組 A



檔案群組 B



檔案群組 C

DEMO

資料分割

資料壓縮

企業版

資料壓縮精靈

壓縮類型

- Row 壓縮
- Page 壓縮

可壓縮的資料庫物件

- 整個資料表
- 非叢集索引
- 索引檢視表
- 分割資料表和所屬索引
 - 可以依每個分割指定壓縮類型

資料壓縮精靈 - wait_categories

選取壓縮類型
選取每個資料分割的壓縮類型。

☐ 所有資料分割使用相同的壓縮類型(U)

資料分割編號	壓縮類型	界限	資料列計數	目前的空間	要求的壓縮空間
1	None		17		

計算(C)

說明(H) < 上一步(B) 下一步(N) > 完成(F) >> 取消

資料壓縮

企業版

通常較適於線上交易
2% - 5% CPU
20% 壓縮比

Row

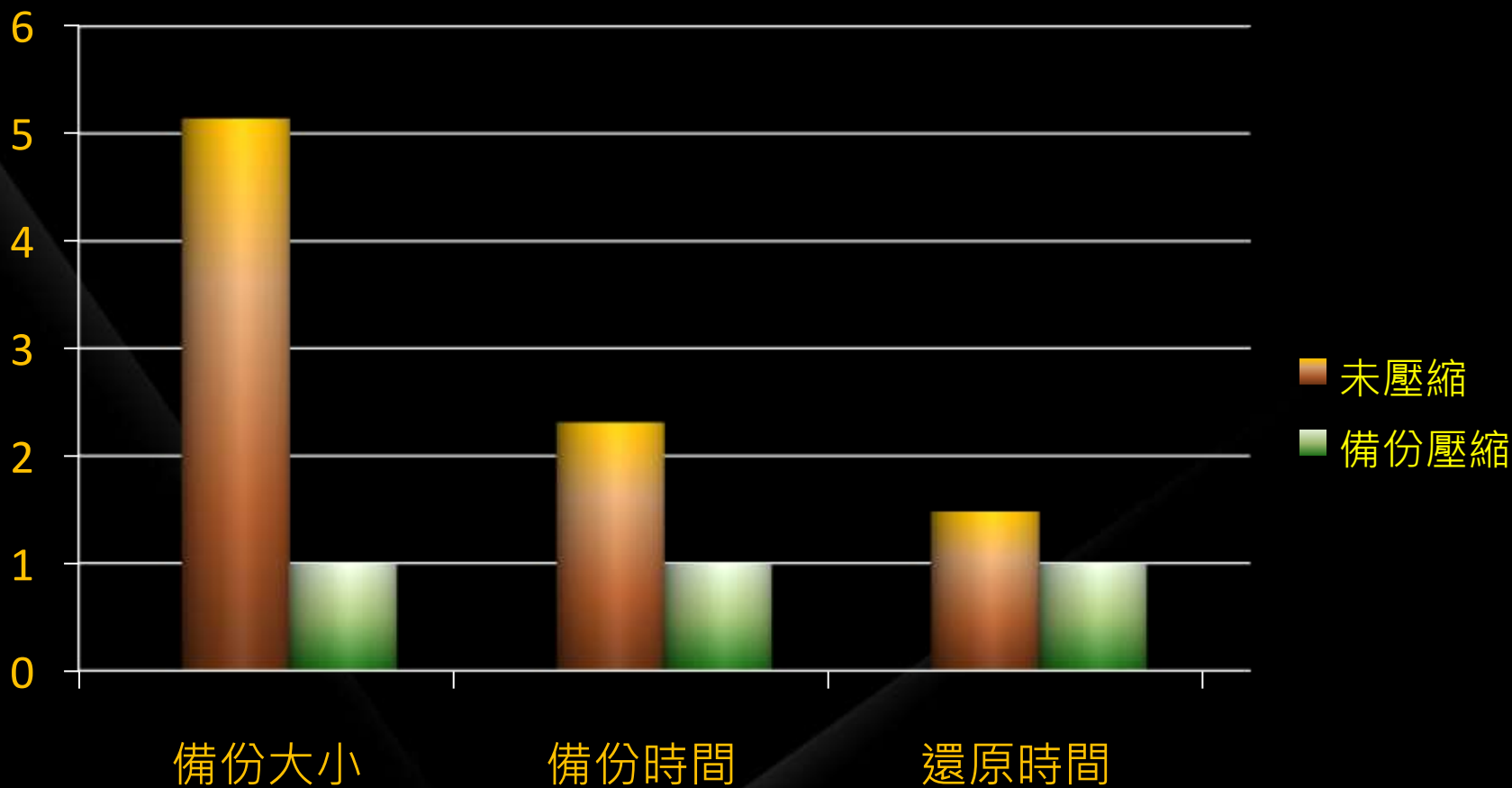
Page

提升記憶體使用率
資料在記憶體內
也保持壓縮狀態

通常較適於資料倉儲
10% - 15% CPU
60% - 70% 壓縮比

備份壓縮

企業版



DEMO

資料壓縮

可擴充共用資料庫

企業版

- 報表資料量龐大
- 資料倉儲系統

每部主機有各自獨立的處理器、記憶體和Tempdb



共用的唯讀檔案必須存放在SAN儲存設備，且共用磁碟區必須是唯讀

省下儲存設備費用、維護費用與電費

計畫指南

- 提供自主的T-SQL執行機制，協助最佳化查詢效能
- 可應用於資料庫升級後，進行查詢之效能微調
- 將T-SQL執行方式，以XML格式儲存在資料庫

`sp_create_plan_guide`

`sp_create_plan_guide_from_handle`

`sp_control_plan_guide`

`sys.plan_guides`

從查詢提示
建立計畫指南

從計畫快取
建立計畫指南

刪除、啟用、
停用計畫指南

系統檢視

DEMO

計畫指南

疏鬆資料行(Sparse Columns)

資料儲存空間更少



Null 值
儲存最佳化



疏鬆資料行(Sparse Columns)

優點

- + Null 值不佔用儲存空間
- + 節省儲存空間，提升效能
- + 查詢/DML 無需修改

缺點

- - 儲存非 Null 值需要更多的儲存空間

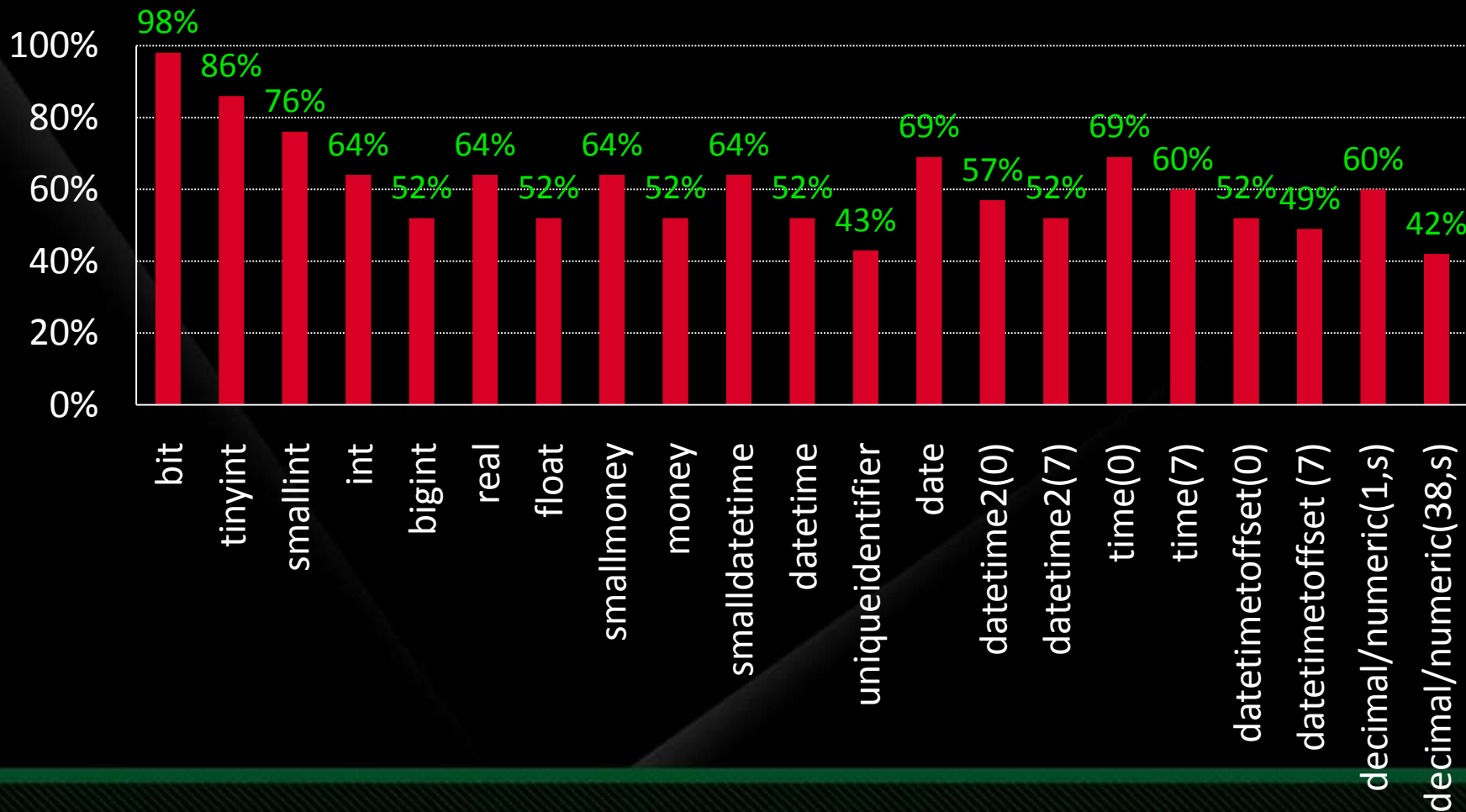
適用情境

- 當空間至少節省了 20% ~ 40%
- 寬型資料表(Wide Tables)，大部分皆為Null 值
 - 支援 30,000 個資料行
- 可搭配資料行集(Column Sets)和篩選索引一起使用

疏鬆資料行

以節省40%空間為例 (資料類型 vs. Null 百分比)

Null 百分比



DEMO

疏鬆資料行

覆蓋索引

- 利用非叢集索引滿足查詢的需求
 - 無需使用 Bookmark Lookup
- 類似更多資訊的名詞解釋架構，除了有名詞的頁次，還包括章、節、類型等資訊
 - 書本的名詞解釋通常只會指出名詞位在哪些頁次，無法回答位在哪個章、節，因此需要搭配書籤
 - WHERE 條件的欄位
 - SELECT 的欄位

```
SELECT Column2, Column3  
FROM TableA  
WHERE Column4 = ?
```

SQL Server 2000 覆蓋索引的限制

- 將 WHERE 條件的欄位與 SELECT 的欄位，皆包括在非叢集索引之中
 - WHERE 條件的欄位在前，再加 SELECT 的欄位
 - 欄位的數目如果太多，Leaf 與 Non-Leaf 皆有相同的欄位
 - B-Tree 過於龐大，導致效率不佳
 - 侷限於每個索引最多 16 個欄位和 900 Bytes 的限制

非叢集索引 + 內含資料行

CREATE INDEX 新增的 INCLUDE 子句

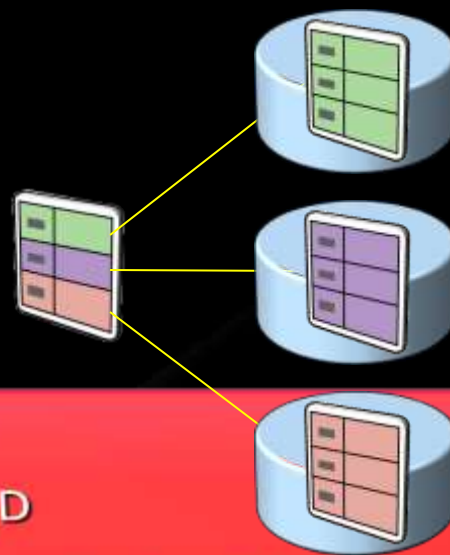
- 將非索引的欄位，儲存於索引的 leaf level
 - 無需依靠索引所屬的資料表，減少磁碟 I/O，使查詢速度更快

```
CREATE NONCLUSTERED INDEX IX_Address_PostalCode  
ON Person.Address (PostalCode)  
INCLUDE (AddressLine1, AddressLine2, City)
```

分割資料表 + 分割索引

● 分割索引 (Partitioned indexes)

- 如同分割資料表一般，將索引指定儲存於多個檔案群組



```
CREATE NONCLUSTERED INDEX  
    IX_TransactionHistory_ReferenceOrderID  
ON Production.TransactionHistory (ReferenceOrderID)  
ON TransactionsPS1 (TransactionDate);
```

DEMO

覆蓋索引

篩選索引(Filtered indexes)

使索引效率再提升

	database_id	db	object_id	objname	index_id	buffer_count
1	1	master	60	sysobjvalues	1	81
2	1	master	41	syscolpars	1	22
3	1	master	3	sysrscols	1	13
4	1	master	41	syscolpars	2	13
5	1	master	29	sysprivs	1	9
6	1	master	34	sysSchobjs	1	9
7	1	master	75	sysmultiobjrefs	1	6
8	1	master	1131151075	spt_values	2	6
9	1	master	34	sysSchobjs	3	6
10	1	master	34	sysSchobjs	2	5
11	1	master	1131151075	spt_values	1	5
12	1	master	55	sysiscols	1	5
13	1	master	7	sysallocunits	1	5
14	1	master	54	sysidxstats	1	4
15	1	master	27	sysowners	3	2
16	1	master	28	sysdbreg	3	2
17	1	master	7	sysallocunits	2	2
18	1	master	18	sysdbfrag	2	2
19	1	master	27	sysowners	2	2
20	1	master	28	sysdbreg	2	2
21	1	master	42	sysxlgn	3	2
22	1	master	46	syscerts	3	2
23	1	master	50	sysscalartypes	3	2
24	1	master	34	sysSchobjs	4	2
25	1	master	46	syscerts	4	2
26	1	master	42	sysxlgn	2	2
27	1	master	43	sysxsrvs	2	2



	database_id	db	object_id	objname	index_id	buffer_count
15	1	master	27	sysowners	3	2
16	1	master	28	sysdbreg	3	2
23	1	master	50	sysscalartypes	3	2
24	1	master	34	sysSchobjs	4	2
25	1	master	46	syscerts	4	2



篩選索引

ID	Type	Brand	Zoom	Resolution	WaistSize	Inseam	Price
1	Camera	Canon	3x	5 M			300	
2	Pants	Dockers			38	34	45	
3	Camera	Nikon	5x	10M			600	
4	Camera	Pentax	3x	3M			100	
5	Pants	Polo			30			
6	Pants	Dockers			40			
7							

建立篩選索引

當WHERE條件符合時，
Query Optimizer會
挑選合適的篩選索引

```
Create Index ZoomIdx on Products (Zoom) where Type = 'Camera';  
Create Index PlusSizeIdx on Products (WaistSize) where Type = 'Pant'  
and WaistSize > 36
```

```
-----  
Select ProductId, Type, Resolution, Zoom where Type = 'Camera'  
Select ProductId, Type, WaistSize, Inseam where Type = 'Pants' and  
WaistSize > 38
```

```
-----  
Insert into Products (ProductId, Type, Brand, WaistSize, Inseam)  
values (201, 'Pants', 'Polo', 30, 32);  
Insert into Products (ProductId, Type, Brand, WaistSize, Inseam)  
values (202, 'Pants', 'Polo', 38, 36);
```

只有在需要時，DML 運作會導致維護索引資料

DEMO

篩選索引

索引檢視表

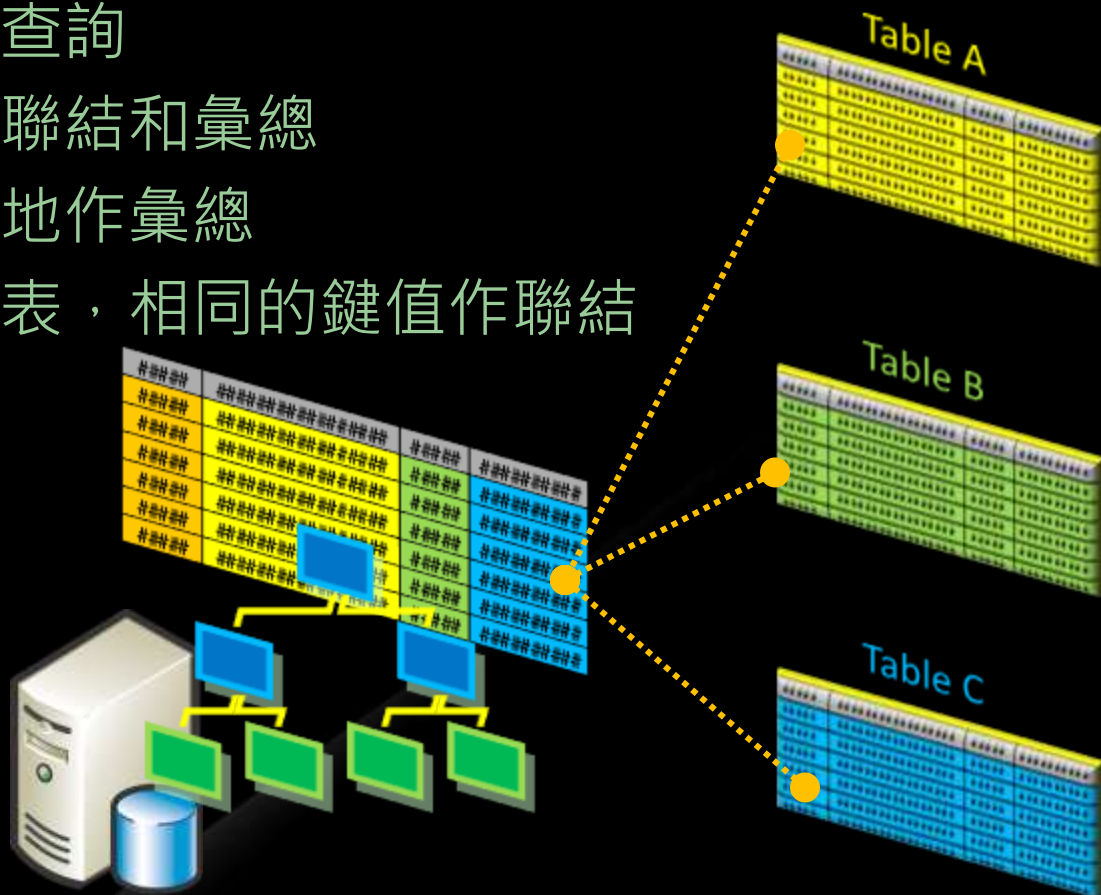
企業版

適用情境

- 重複同一種模式的查詢
- 對於大型資料表作聯結和彙總
- 對於某些欄位重複地作彙總
- 重覆對相同的資料表，相同的鍵值作聯結

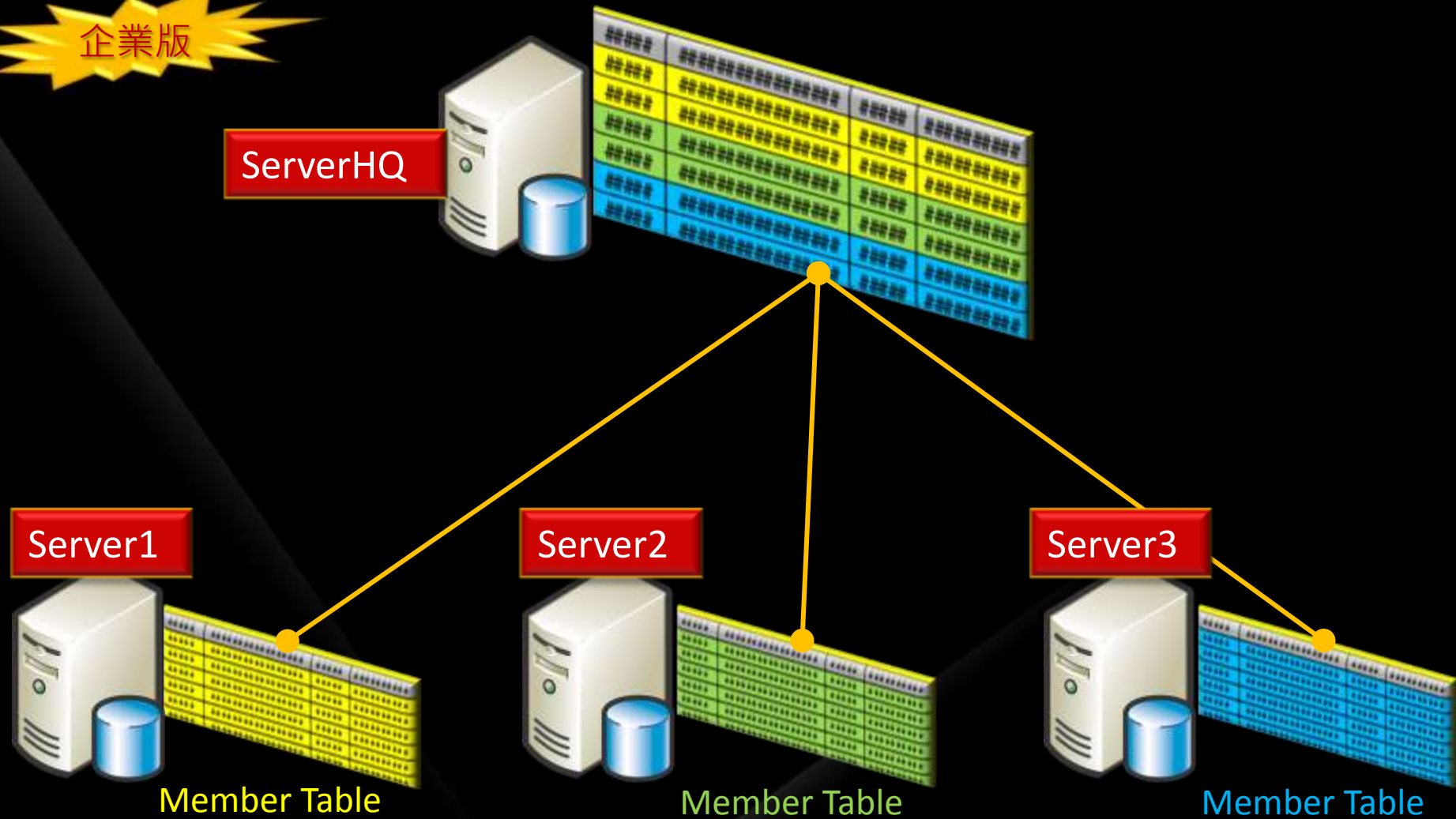
不適用情境

- 異動量大的資料表
- 索引檢視與原始資料表之資料量相差不多時



分散式分割檢視表

企業版



分散式分割檢視表

企業版

成員資料表

- CHECK條件、索引、定序
- 計算型欄位不可有索引、相同的主鍵定義與ANSI PADDING選項設定

為每個執行個體建立連結伺服器

- sp_addlinkedserver

建立分割檢視表

- 可搭配 INSTEAD OF 觸發程序

```
CREATE VIEW Sales.Orders AS
  SELECT * FROM Server1.SalesDB.Sales.Orders
UNION ALL
  SELECT * FROM Server2.SalesDB.Sales.Orders
UNION ALL
  SELECT * FROM Server3.SalesDB.Sales.Orders
```

DEMO

分散式分割檢視表

Q&A

高可用度

★ SQL Server 2008 新功能
★ SQL Server 2008 增強功能

- ★ 資料庫鏡像
- ★ 容錯叢集
- ★ 交易記錄檔傳送
- ★ 點對點交易式複寫
- 熱新增記憶體和處理器

- ★ 備份壓縮
- 線上還原
- 鏡像備份
- 資料庫快照
- 快速復原
- 線上索引

Microsoft®
SQL Server™ 2008

高可用度

- 確保在 可接受的條件下 持續運作，由系統的
服務等級規範決定
 - Service Level Agreements (SLA)
- 例：
 - 使用訂單系統時，必須於 1 分鐘以內 完成訂單
處理
 - 每個月最長的停機時間為 5 分鐘，且不得有任
何訂單資料遺失

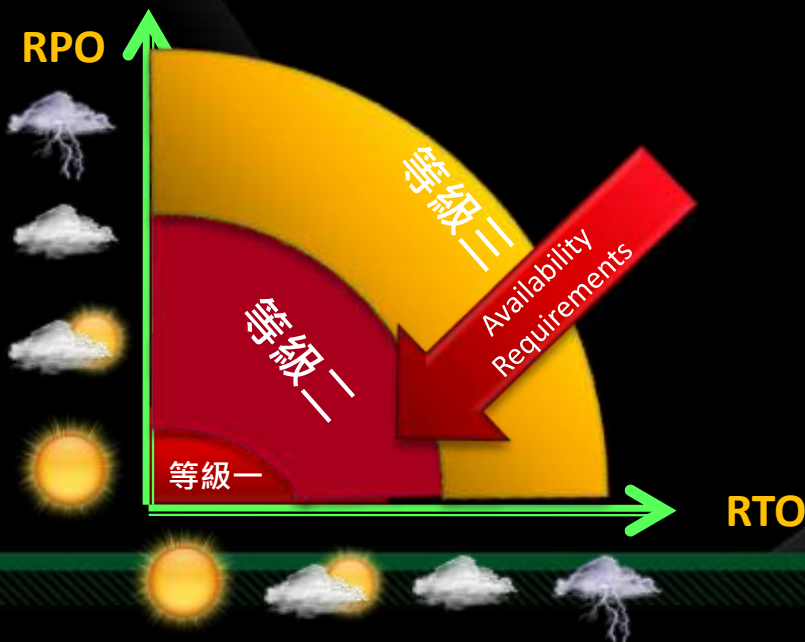
定義SLA的復原指標

📌 RTO: Recovery Time Objective

- 可以多快完成復原？
- (停機時間？)

📌 RPO: Recovery Point Objective

- 資料可以復原到哪個時間點？
- (多少的資料損失量？)



可用度等級	RTO	RPO
等級一	>99.99% (小於1小時)	小於5分鐘
等級二	99.9% - 99.99% (1至 8.5 小時)	5分鐘至8.5小時
等級三	(<99.9%) (數小時至數天)	數小時至數天

高可用性的防護

● 保護系統免於毀損

- 主機層級：硬體故障、服務中止、資料損壞 ...
- 機房層級：颱風、火災、地震 ...

非計畫性停機

● 復原錯誤狀況 (使用者或應用程式所造成)

- 不正確的資料異動、意外變更資料...

● 線上管理

- 軟體 / 硬體升級、索引重組...

計畫性停機

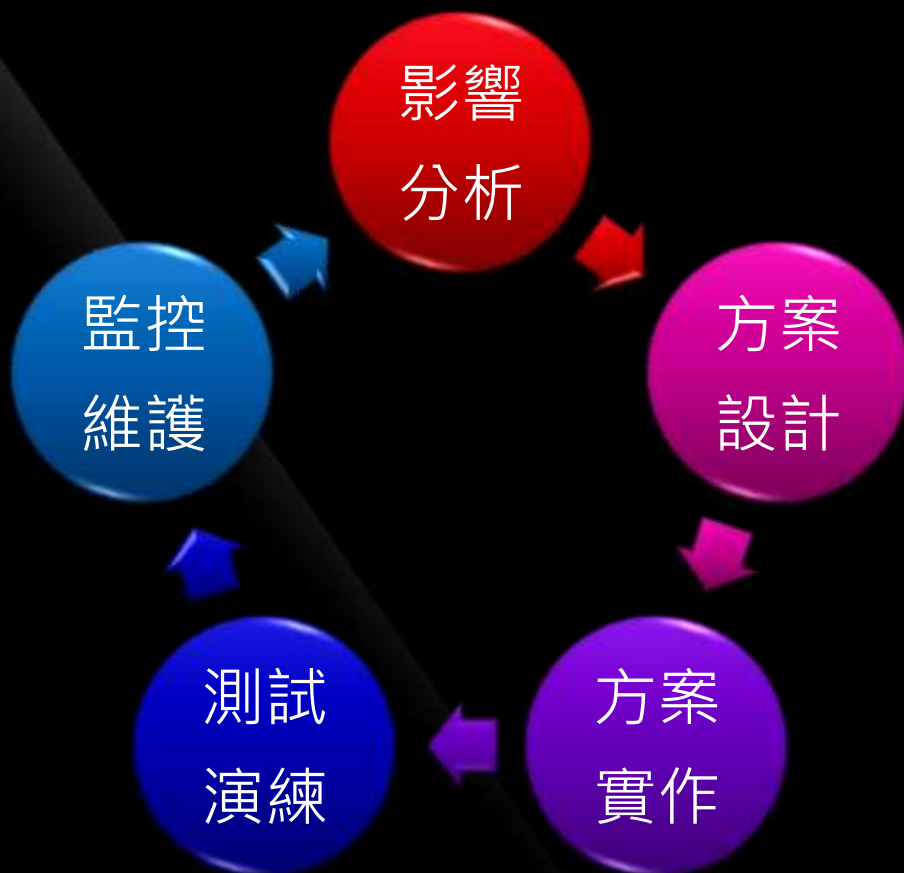
● 預期資源之使用情況

- 提升系統硬體資源，以支援更多同時使用者數、處理量

災難復原

- 設計一套復原的流程與作業程序，以因應天然災害或人為疏失造成的損壞，使商業運作可以盡快回復正常
 - 本地 vs. 異地
 - 容錯機制 vs. 備援機制
 - 災難復原演練
 - 確認可行性
 - 人員的熟悉度

商業持續運作規劃 (BCP)



● 影響分析

- 系統的關鍵功能
- 找出並確認可能發生的威脅
- 復原目標 (RTO + RPO)

● 方案設計

- 依據各種可能發生的威脅設計符合復原目標的方案
- 同時評估費用與人力等資源
- 費用效益分析

● 方案實作

- 實際部署高可用度方案

● 測試演練

- 測試高可用度方案是否符合復原目標

● 監控維護

- 日常監控
- 定期測試演練與檢視程序

SQL Server 高可用度規劃

影響分析

- 依資料庫區分影響的應用程式與服務
- 資料庫停機的因素分析
- 保護等級：本地 HA、異地 HA

方案設計

- 瞭解有哪些方案可供選擇？
- 瞭解方案的特性與費用？

方案實作

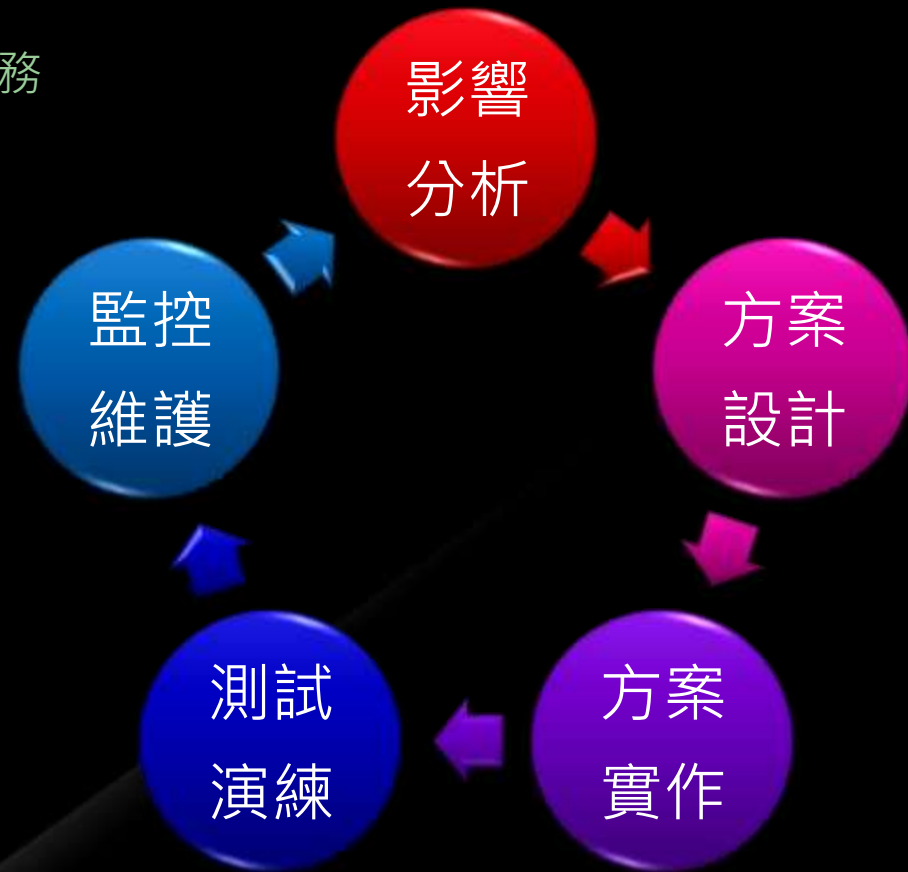
- 部署的步驟與 Best Practices？

測試演練

- 如何測試與演練高可用度方案？

監控維護

- 如何監控與維護高可用度方案？



可用度 RTO + RPO 比較

● 熱備援 (Hot Standby)

RPO = 0

- 備援伺服器亦擁有主要伺服器的資料

- 提供與主要伺服器交易一致的資料

RTO (秒)

- 能自動偵測主要伺服器故障狀況與自動接續服務

● 暖備援 (Warm Standby)

RPO >= 0

- 備援伺服器亦擁有主要伺服器的資料，不過...

- 不一定能提供與主要伺服器一致的資料

RTO (分)

- 可能需要手動偵測主要伺服器毀損或停機狀況與手動接續服務

● 冷備援 (Cold Standby)

RPO > 0

- 還原資料之備份

- 必須有正確的作業系統和相關軟體等的備份

RTO (時)

- 必須考慮繼續恢復正常服務的延遲狀況是否能接受

可用度 RTO + RPO 比較

天



分鐘



秒



無

可能的資料損失量(RPO)

資料庫快照

卸離 / 附加

冷備援

備份 / 還原

線上還原

交易記錄檔傳送

複寫

點對點

暖備援

資料庫鏡像

高效能

容錯叢集

資料庫鏡像

高可用

熱備援

停機時間 – 從異常偵測，移轉異常 / 修復異常到重新恢復正常運作 (RTO)

秒



分鐘



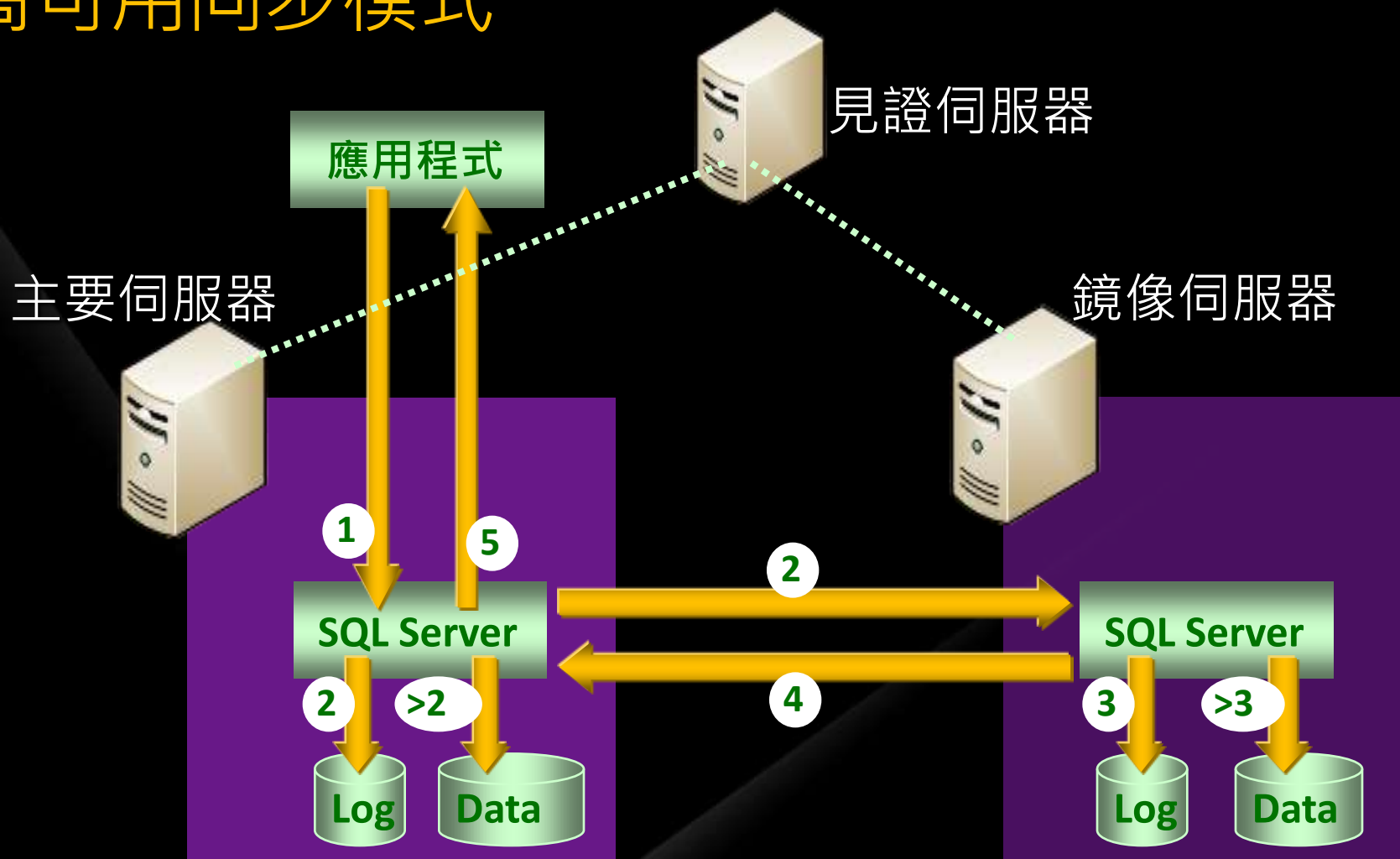
小時



天

資料庫鏡像

高可用同步模式



資料庫鏡像

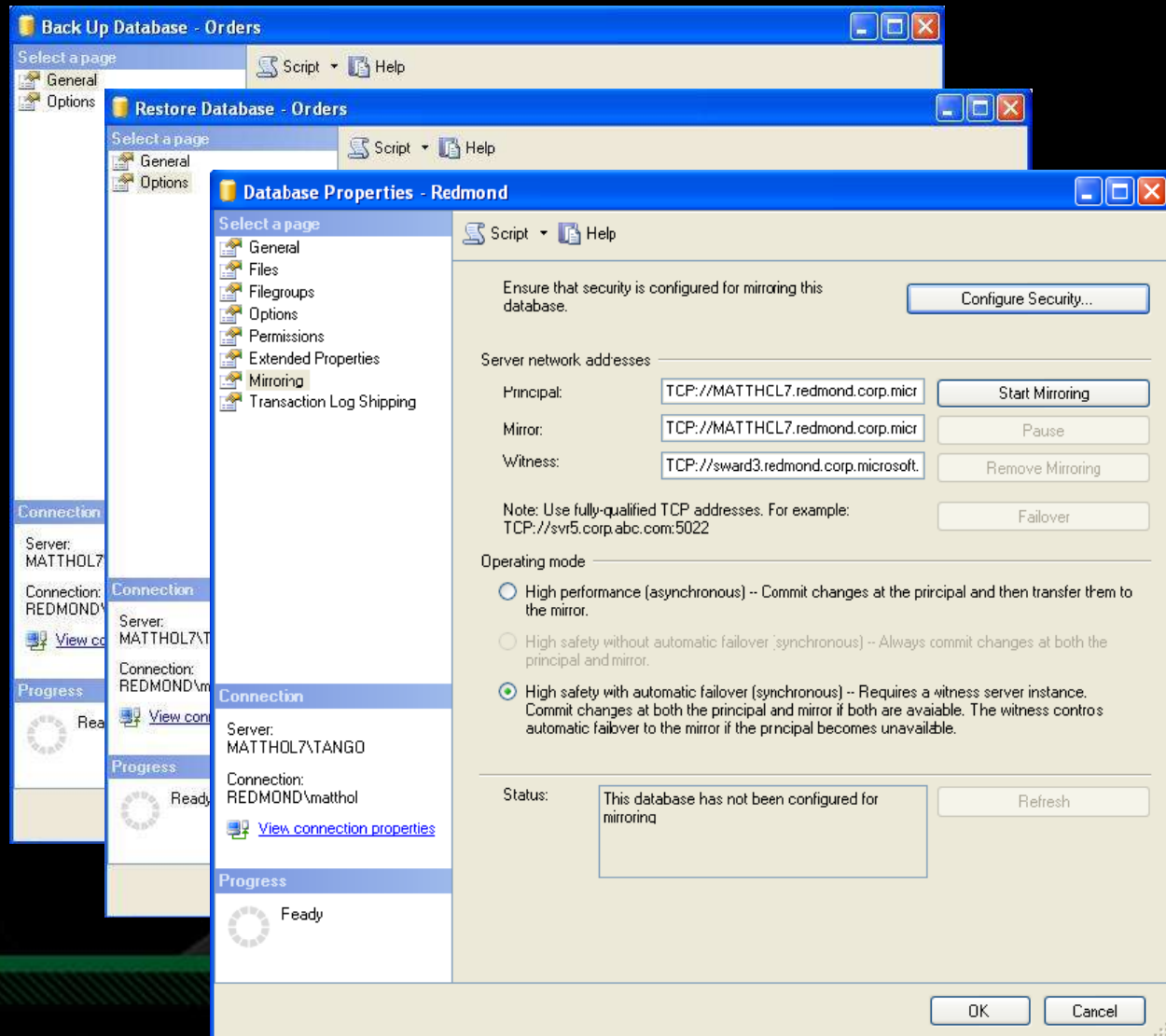
- 提供極快速的容錯移轉機制
 - 鏡像資料庫是主要資料庫的熱備援
- 容錯機制：
 - 以資料庫為單位
- 三種模式：
 - 高可用（同步）
 - 自動容錯移轉
 - 見證伺服器
 - 高保護（同步）
 - 高效能（非同步）

企業版

資料庫鏡像

4 步驟完成設定

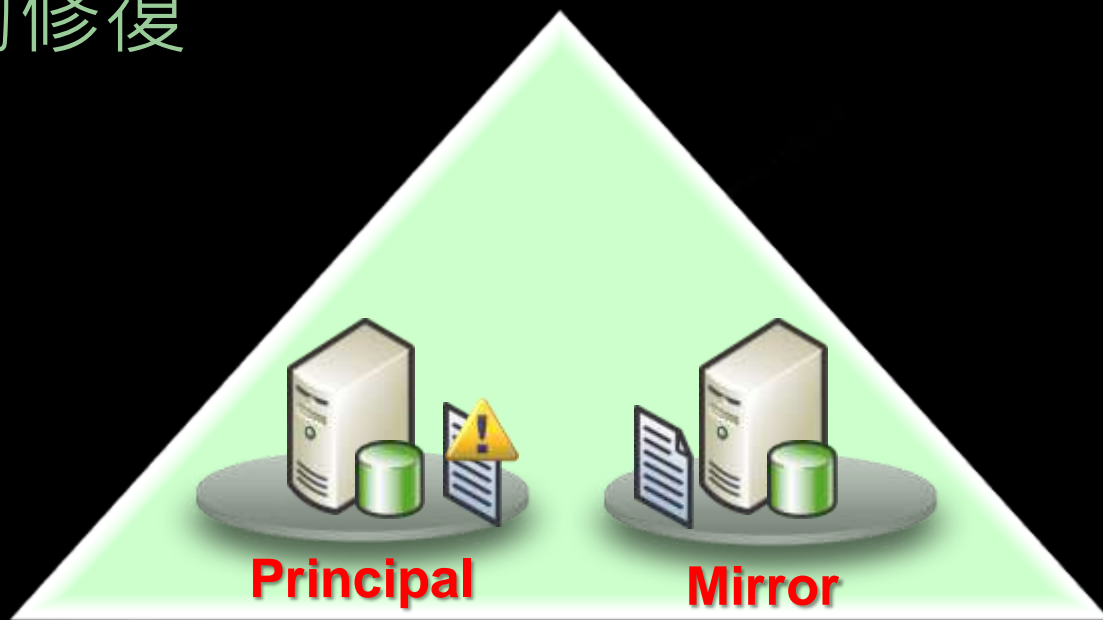
1. 在主要伺服器端執行資料庫完整備份
2. 在鏡像伺服器端還原資料庫並指定 `no_recovery` 選項
3. 設定鏡像傳輸時使用的端點
4. 指定資料庫鏡像之模式，啟動鏡像



資料庫鏡像

SQL Server 2008 增強功能

- 壓縮記錄檔傳送
- 減少手動移轉時所需的復原時間
- 資料頁毀損自動修復



資料庫鏡像

取捨之間

📍 資料庫層級的容錯移轉

- + 提供最快速的移轉 (RTO 最小)
- - 系統資料庫不會受到保護

📍 同步模式

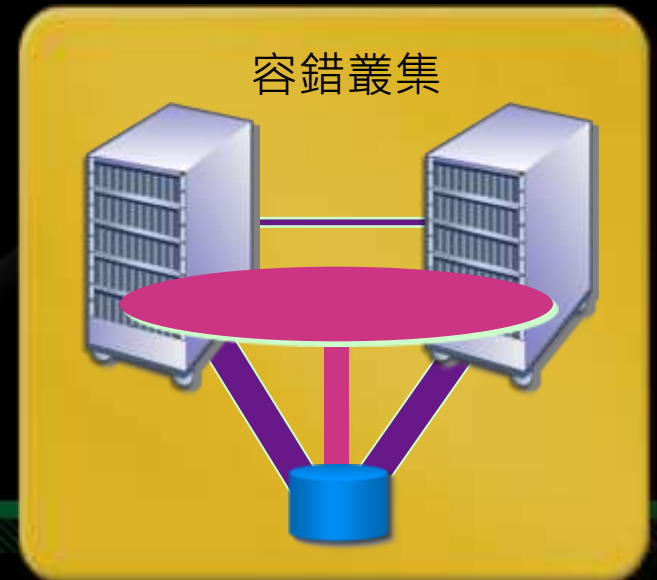
- + 不會有資料遺失 (RPO=0)
- - 效能會受到影響
- - 每個資料庫僅限一個鏡像資料庫
- - 當同一部主機有愈多的資料庫設定資料庫鏡像時，效能影響愈顯著
- - 鏡像資料庫無法直接存取，需搭配資料庫快照

DEMO

資料庫鏡像

容錯叢集

- 自動偵測錯誤並自動進行移轉
- 資料不會遺失，對負載也不會有直接的影響
- 可保護任何一點伺服器錯誤所造成的問題
- 在硬體支援下可支援異地容錯叢集
- 透過 SQL Server 快速復原技術可進行快速回復
- 資料只有一份



容錯叢集

- 建構在 Windows 容錯叢集之上，以執行個體為保護層級 (RTO)
 - 透過式移轉，包括：
 - 網路名稱 (執行個體)、網路位址 (IP)、儲存設備和 SQL Server Agent 服務
 - 用戶端無須更改連線方式
 - 異地容錯叢集需藉由硬體設備達成
- 要結合 SAN 的硬體容錯設計，以避免因 SAN 毀損而導致資料損失 (RPO)
 - 不像資料庫鏡像和交易記錄檔傳送有一份以上的資料

容錯叢集

SQL Server 2008 增強功能

● 更穩固和可靠的安裝

- 新的安裝精靈排除了可能發生的安裝與設定錯誤
- 整合 OS 和 SQL Server 的活動訊號檢查
- 版本升級與安裝修補程式

● Windows Server 2008 新功能

- 新的叢集驗證工具 (節點測試、網路測試、存放區測試)
- 支援 DHCP
- 完全整合的 IPv6，進行節點或活動訊號通訊
- 使用 DNS 不需要 WINS 和 NetBIOS 名稱解析廣播
- 最多支援 16 個伺服器節點

企業版

容錯叢集

取捨之間

● 執行個體層級的容錯移轉

- + 對用戶端而言，容錯叢集主機的移轉是透通式移轉
- + 整個執行個體的高可用性
- - 移轉時服務要重新啟動 (RTO較長)

● 資料無重複配置

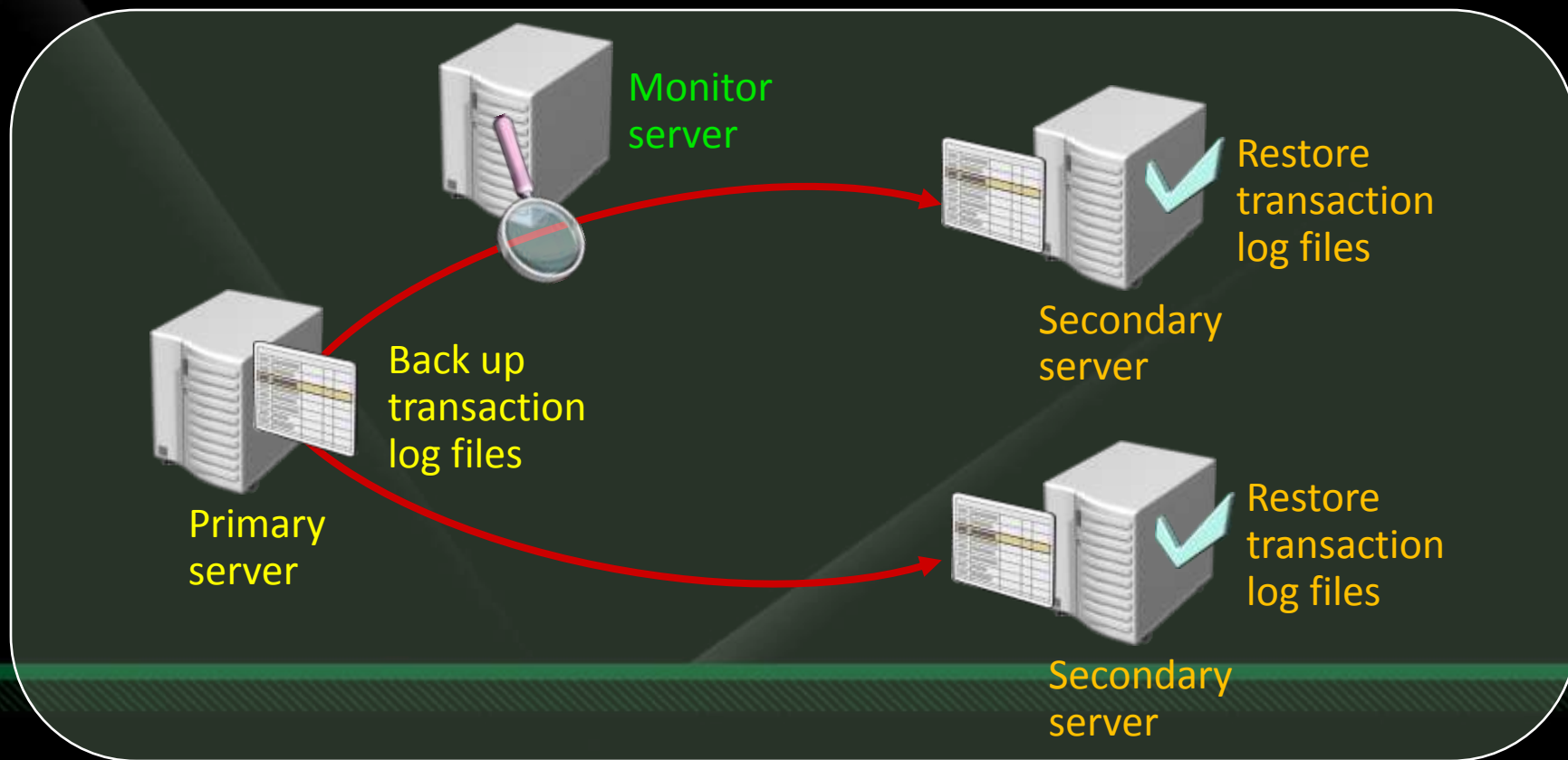
- + 無效能影響
- - 需要搭配 SAN 容錯機制，以避免資料損失(RPO=0)

● 異地備援之容錯叢集：VLAN + 儲存設備支援的複寫機制

- + SQL Server 無需特殊設定
- - 需要較複雜的硬體解決方案

交易記錄檔傳送

- 備援主機可供唯讀之資料來源
- 可應用於異地備援
- 注意：不要再做其他的交易記錄檔備份



交易記錄檔傳送

- 利用交易記錄檔備份和還原建構備援機制
 - 在主要執行個體上備份資料庫
 - 在備援執行個體上還原資料庫
- 可以將異動資料套用至多部備援執行個體
- 可刻意設計為不要立即還原
 - 利用延遲還原的設計，容易復原因使用者意外變更資料所造成的資料錯誤
- 沒有自動容錯移轉機制

交易記錄檔傳送 取捨之間

- 非同步的備援主機
 - + 最小的效能影響
 - - 可能會有資料遺失 ($RPO > 0$)
- 資料擷取 (備份) 和資料更新 (還原) 採用分開的機制
 - + 可以支援多個備援主機
 - - 需要管理者進行移轉 (RTO 較長)

DEMO

交易記錄檔傳送

交易式複寫

- 高效能、資料表層級的容錯
- 主要應用情境
 - 可以彈性設計復原架構
 - 不用以執行個體或是資料庫為保護單位，例如只針對最重要的資料表
 - 備援主機可以提供即時性報表的資料來源
 - 備援主機可以分散應用程式的查詢負荷 (向外擴充)
- 二種複寫類型可應用於高可用度和災難復原
 - 交易式複寫
 - 點對點複寫

複寫

取捨之間

- 資料表 (查詢) 層級的容錯機制
 - + 容易對應用程式架構做最佳化設計
 - + 可以從設計架構不同的資料庫進行複寫
 - - 效能影響
 - - 最小的移轉單位
- 複寫資料可提供讀取和變更
 - + 可發揮更多的硬體使用率
 - - 若允許變更 (訂閱者)，可能影響到資料的完整性
- 非同步
 - + 可以將效能影響降至最低，並且得到分散負荷的好處
 - - 需要管理者進行移轉 (RTO 較長)
 - - 有資料遺失的風險 (RPO > 0)

點對點複寫

企業版

SQL Server 2008 增強功能

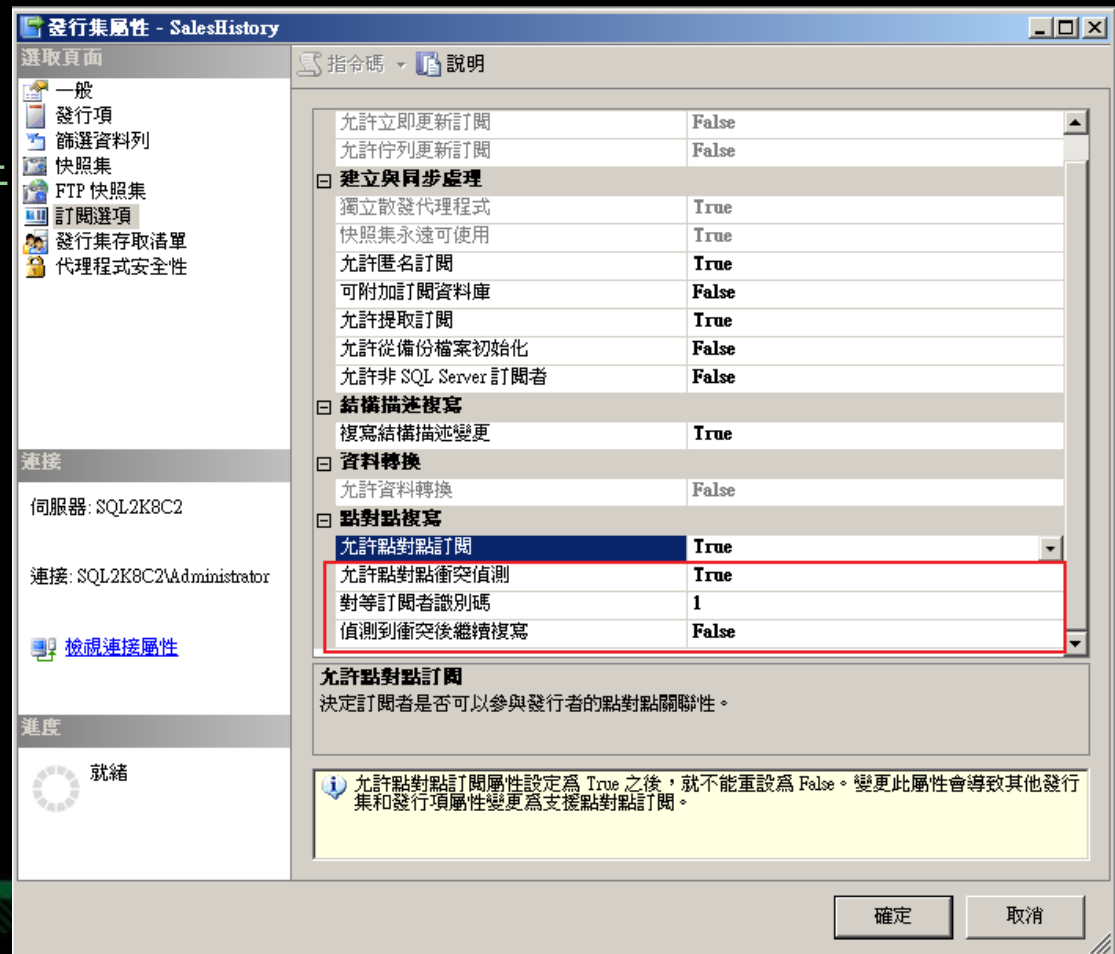
加入新結點時無須中斷原有的複寫架構

- 利用 log sequence number (LSN) 進行初始化

資料衝突偵測機制

Topology Viewer

- Visualize
- Create/Modify



DEMO

點對點複寫

熱新增記憶體和處理器

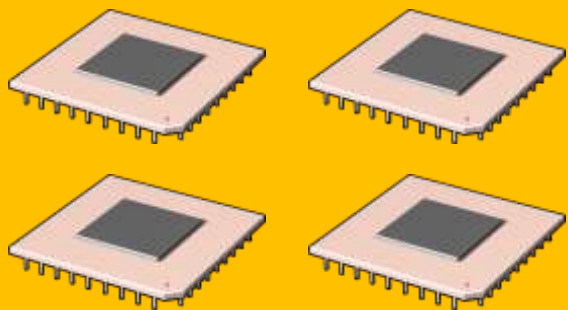
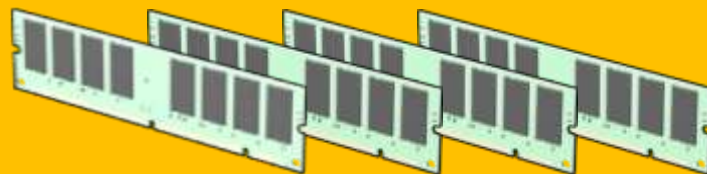
企業版

● 減少計畫性停機時間

● 熱新增記憶體

- 無須重新啟動伺服器即可增加實體記憶體
- Windows Server 2003/2008 (Enterprise/Datacenter)

硬體需支援



● 熱新增處理器

- 實體新增處理器、邏輯線上硬體分割、虛擬化層
- 64 位元 Windows Server 2008 (Enterprise/Datacenter)

備份壓縮

企業版



常見問題：

- 多少的壓縮比？
- 對伺服器硬體資源有何影響？

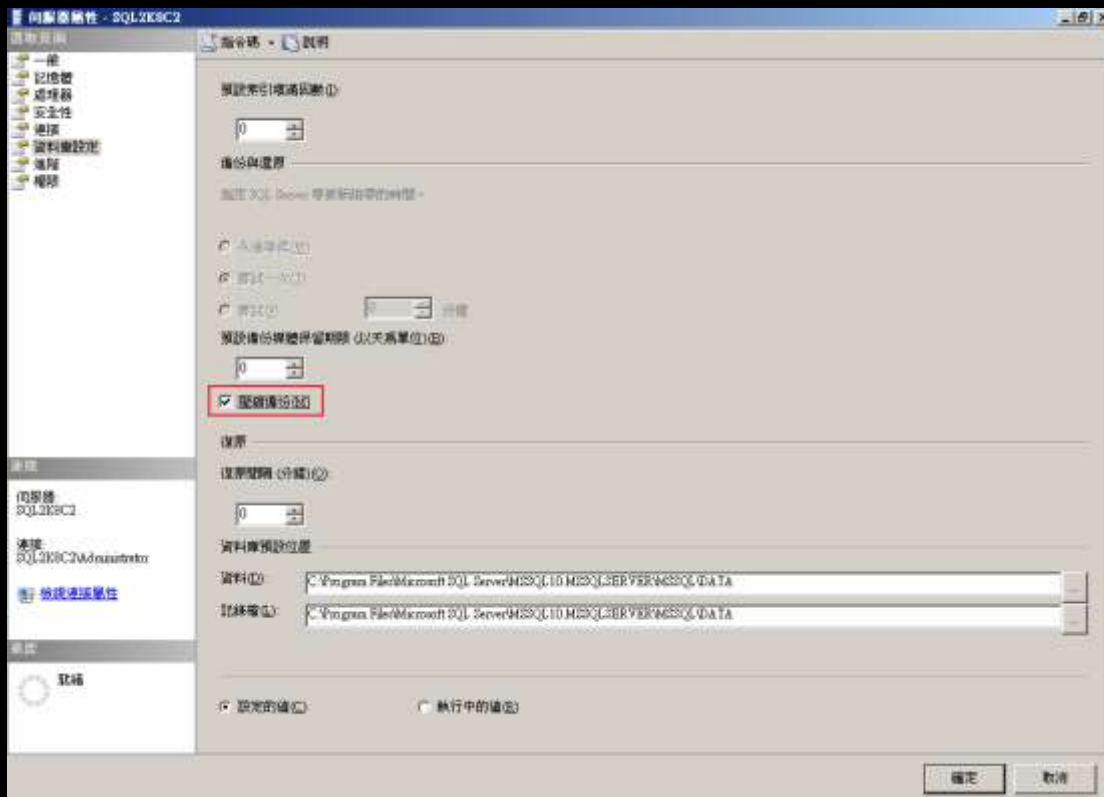


壓縮比會因下列因素而異：

- 資料庫的資料類型
- 儲存的資料是否已經壓縮過
- 資料或資料庫是否已經加密

備份壓縮

- ❏ 降低備份裝置 I/O
- ❏ 加快備份速度
- ❏ 交易紀錄檔傳送亦可使用備份壓縮



備份壓縮實測案例

縮短備份時間與減少儲存空間

	備份方式	備份檔案大小	備份時間
A	未壓縮，1 個備份檔案	32.8 GB * 1	1524 秒 (00:25:24)
B	未壓縮，4 個備份檔案	8.2 GB * 4	1020 秒 (00:17:00)
C	壓縮，1 個備份檔案	5.1 GB * 1	417 秒 (00:06:57)
D	壓縮，4 個備份檔案	1.3 GB * 4	392 秒 (00:06:32)

📌 執行時間

- B 較 A 節省 33% 的執行時間
- C 較 A 節省 73% 的執行時間
- D 較 A 節省 74% 的執行時間
- D 較 C 節省 6% 的執行時間

📌 壓縮比

- C 較 A 節省 84% 的儲存空間

PS.

1. 備份壓縮比例會因資料庫內的資料型態等因素而有所差異
2. 測試主機: HP ProLiant DL360 G4p
 - CPU: Xeon 3.0GHz Dual-Core * 2
 - Memory: 4GB
 - HDD: 73GB * 2 10,000 rpm (RAID 0)

DEMO

備份壓縮

線上還原

企業版

- 還原檔案或是還原分頁都是線上還原
 - 還原 Primary 檔案群組的任一檔案
 - 整個資料庫為離線狀態
 - 還原 Primary 檔案群組以外的任一檔案
 - 該檔案群組不可以使用
 - 其他已經還原的檔案群組都可以使用
 - 還原分頁時適用時機
 - page checksum 錯誤
 - torn pages 錯誤

線上還原機制

企業版

一直可以存取

Database

Primary 檔案群組



檔案群組 A



檔案群組 B



檔案群組 C



交易記錄檔之備份



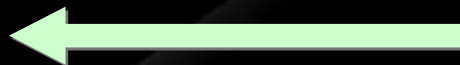
Primary 檔案
群組之備份

檔案群組 A
之備份

檔案群組 B
之備份

檔案群組 C
之備份

還原資料庫
交易記錄檔 備份

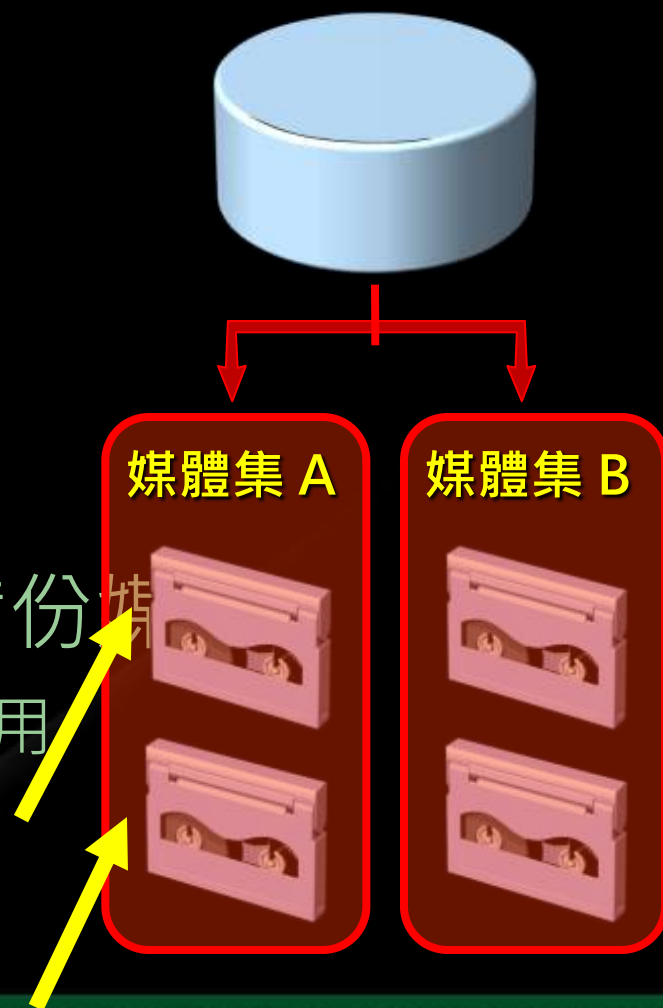


DEMO

線上還原

鏡像備份

- 備份媒體達到雙重保護
- 備份媒體必須是相同型態
 - 皆是磁碟或皆是磁帶
 - 允許最多 4 個鏡像
 - 效能不受影響
- 還原時僅需要其中的一個備份
 - 鏡像媒體集間的成員可以交替使用



鏡像備份之實作

● 利用鏡像備份將備份媒體集鏡像到網路

- 避免發生備份相同硬體遺失毀損

--加入備份的device到本機硬體

```
EXEC sp_addumpdevice 'disk', 'CTPdump1', 'e:\dump\dump1.bak';
```

```
EXEC sp_addumpdevice 'disk', 'CTPdump2', 'e:\dump\dump2.bak';
```

--加入備份的device到網路硬體

```
EXEC sp_addumpdevice 'disk', 'CTPdump3', '\\beta3a\c$\dump\dump3.bak';
```

```
EXEC sp_addumpdevice 'disk', 'CTPdump4', '\\beta3a\c$\dump\dump4.bak';
```

--透過媒體集將被份資料鏡像到另一個媒體集(網路)

```
BACKUP DATABASE northwind
```

```
TO CTPdump1,CTPdump2
```

```
MIRROR
```

```
TO CTPdump3,CTPdump4
```

```
WITH FORMAT, INIT, MEDIADESCRIPTION = N'媒體設備', MEDIANAME = N'MirrorBS'
```

Messages

Processed 440 pages for database 'northwind', file 'Northwind' on file 1.

Processed 1 pages for database 'northwind', file 'Northwind_log' on file 1.

BACKUP DATABASE successfully processed 441 pages in 3.417 seconds (1.057 MB/sec).

Address E:\dump

Name	Size	Type
dump1.bak	1,886 KB	BAK File
dump2.bak	1,821 KB	BAK File

Address \\beta3a\c\$\dump

Name	Size	Type	Date Modified	Attributes
dump3.bak	1,886 KB	BAK File	6/16/2005 4:22 PM	A
dump4.bak	1,821 KB	BAK File	6/16/2005 4:22 PM	A

鏡像備份之還原

可以透過鏡像備份進行還原

- 僅需要一組備份資料
- 不同鏡像媒體集可以對等互換

--回覆資料僅需要一份鏡像媒體集

```
RESTORE DATABASE northwind  
FROM CTPdump1,CTPdump2|
```

可以使用單一鏡像媒體集
[1,2] or [3,4]

Messages

Processed 440 pages for database 'northwind', file 'Northwind' on file 1.
Processed 1 pages for database 'northwind', file 'Northwind_log' on file 1.
RESTORE DATABASE successfully processed 441 pages in 1.212 seconds (2.980 MB/sec).

--回覆資料可以互換不同鏡像中的媒體

```
RESTORE DATABASE northwind  
FROM CTPdump1,CTPdump4
```

可以互換鏡像媒體集
[1,4] or [2,3]

Messages

Processed 440 pages for database 'northwind', file 'Northwind' on file 1.
Processed 1 pages for database 'northwind', file 'Northwind_log' on file 1.
RESTORE DATABASE successfully processed 441 pages in 1.564 seconds (2.309 MB/sec).

DEMO

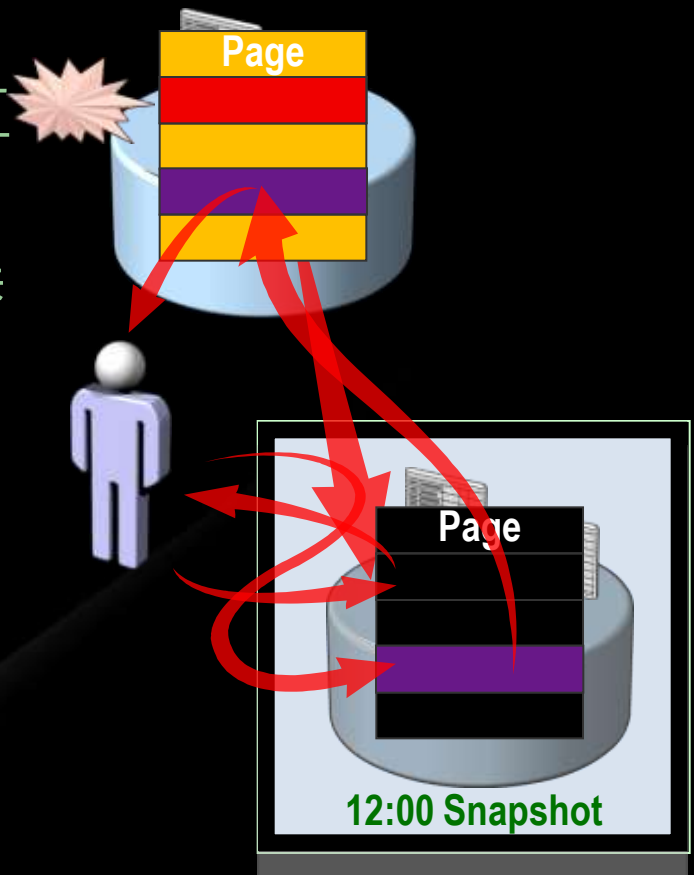
鏡像備份

資料庫快照

企業版

● 保留資料庫在特定時間點的資料狀態

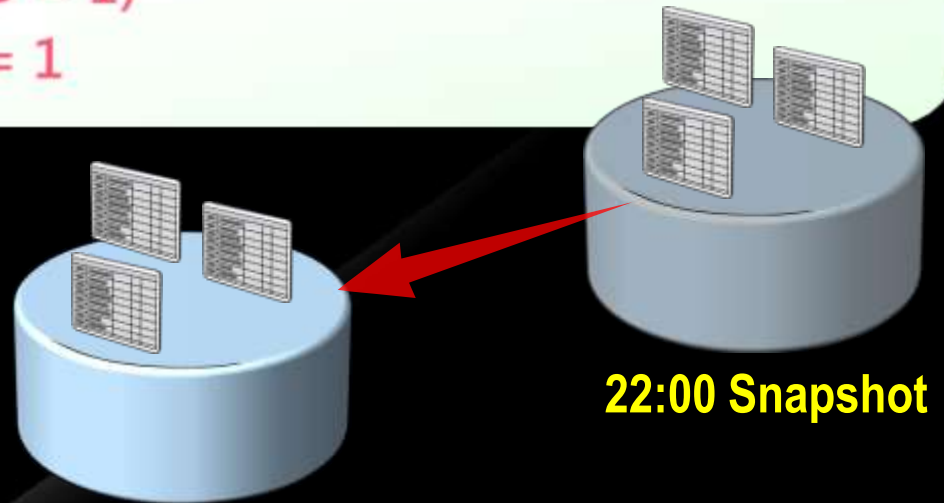
- 以 CREATE DATABASE 建立資料庫快照
- 採用 copy-on-write 處理異動資料
 - 系統不會直接將所有資料放置於快照區中
 - 異動前的資料置於新的快照區中
 - 原資料庫看到異動後的資料
 - 未更改的資料，都轉向原始資料庫進行資料提取



利用資料庫快照復原資料範例

● 範例一：復原被誤改的資料

```
USE AdventureWorks  
UPDATE HR.Department  
SET Name = (SELECT Name FROM  
AWsnapshot_2200.HR.Department  
WHERE DepartmentID = 1)  
WHERE DepartmentID = 1
```



利用資料庫快照復原資料範例

❏ 範例二：復原被誤刪的資料

```
USE AdventureWorks  
ALTER TABLE Production.WorkOrderRouting  
NOCHECK CONSTRAINT  
CK_WorkOrderRouting_ActualEndDate
```

```
INSERT INTO Production.WorkOrderRouting  
SELECT *  
FROM AWsnapshot_2200.Production.WorkOrderRouting
```

```
ALTER TABLE Production.WorkOrderRouting  
CHECK CONSTRAINT CK_WorkOrderRouting_ActualEndDate
```


利用資料庫快照復原資料範例

● 範例三：復原被誤刪的資料庫物件

1. 於資料庫快照產生資料庫物件與相關物件之指令檔
2. 於來源資料庫執行指令檔
3. 若有需要，可以利用範例二的方式復原資料

● 範例四：利用資料庫快照復原整個資料庫

```
RESTORE DATABASE Northwind  
FROM DATABASE_SNAPSHOT = 'AWsnapshot_2200'
```

須先刪除同一個資料庫的其他資料庫快照

較整個資料庫還原速度快

還原過程中，使用者無法存取這二個資料庫

DEMO

資料庫快照

快速復原

企業版

- SQL 2000 資料庫必須等復原過程全部完成後方可開始存取
 - Redo (Rolled forward) + Undo (Rollback)



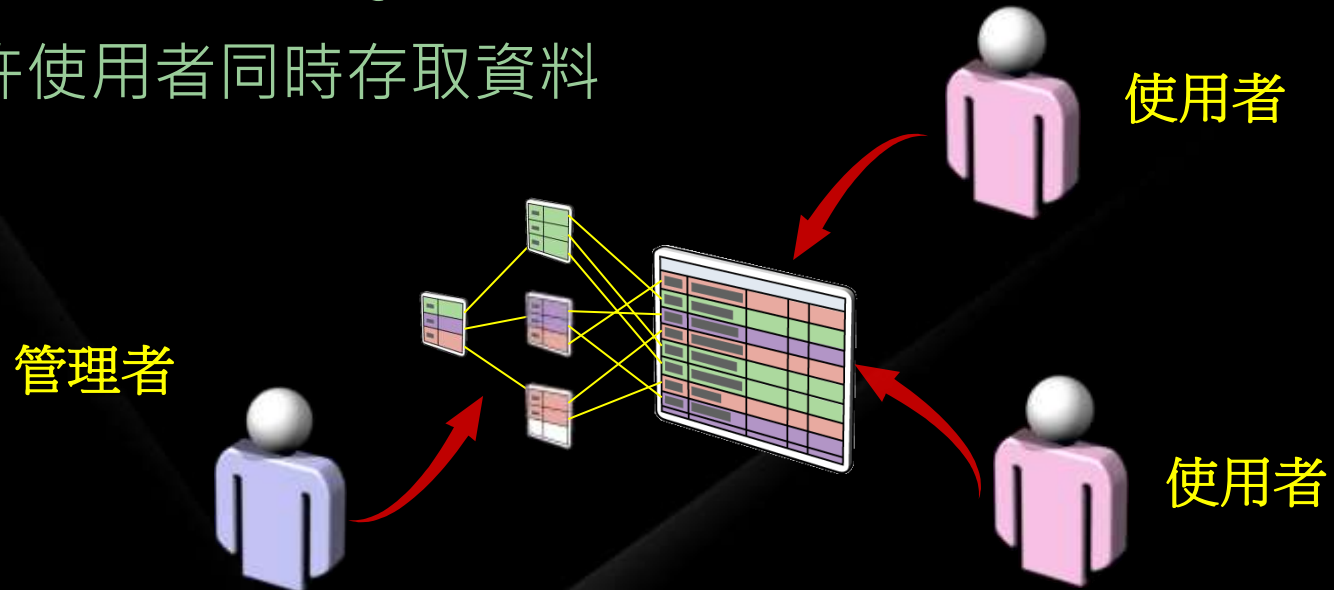
- SQL 2005 / SQL 2008 僅需完成 Redo，資料庫便可以存取



線上索引作業

● 索引之建立、重建、刪除皆可採用 Online 選項

- 不再採用以往 SQL Server 的獨占式鎖定
- 允許使用者同時存取資料



```
CREATE INDEX ...  
ON MyTable(...) WITH (ONLINE = ON)
```

DEMO

線上索引作業

綜合應用

資料庫鏡像

- 資料庫容錯 (本地 / 異地)

交易記錄檔傳送

- 資料庫容錯 (多個備援主機)
- 資料錯誤復原

複寫

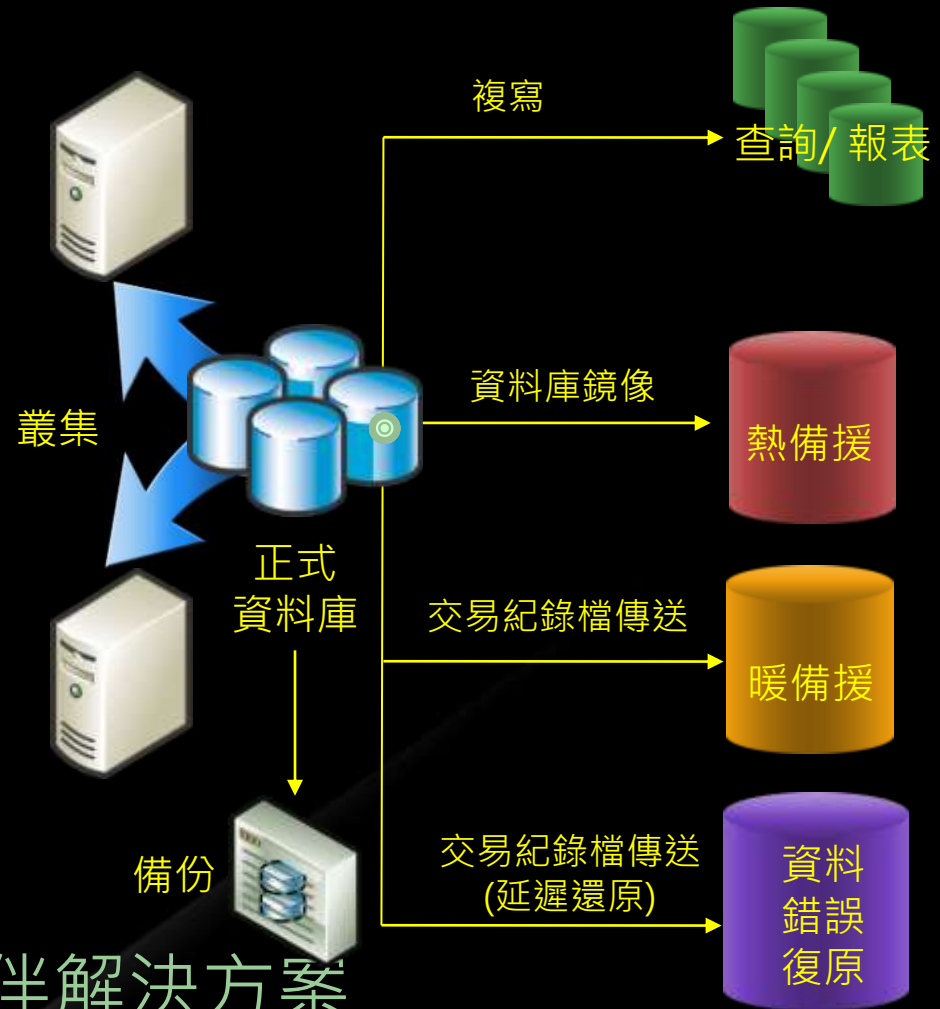
- 報表資料庫 (多主機)
- 查詢橫向擴充兼資料容錯

叢集

- 本地伺服器容錯

Always On 儲存設備夥伴解決方案

- 儲存設備高可用度 (異地)



Always On 方案技術特性比較

RPO

移轉機制

備援 使用

費用

高可用度 方案		無資料 損失 (RPO=0)	移轉單位			自動移 轉 (RTO)	讀取	多個 備援	寫入	硬體	程式效 能影響	管理 性
			Inst	DB	Tab							
交易記錄檔傳送		✗	✗	✓	✗	✗	✗*	✓	✗	低	低	低
資料庫鏡 像	同步	✓	✗	✓	✗	✓**	✗*	✗	✗	低	高	低
	非同步	✗	✗	✓	✗	✗	✗*	✗	✗	低	低	低
容錯叢集		✓	✓	✗	✗	✓	✗	✓	✗	高***	低***	低***
交易式複寫		✗	✗	✗	✓	✗	✓	✓	✗	低	低	高
點對點複寫		✗	✗	✗	✓	✗	✓	✓	✓	低	低	高

* 資料庫鏡像和交易記錄檔傳送可搭配資料庫快照或STANDBY 模式提供唯讀存取

** 資料庫鏡像提供最快速的容錯移轉

*** 需搭配 SAN 解決方案

企業級安全

★ SQL Server 2008 新功能

- ★ 透通資料加密
- ★ 可延伸金鑰管理
- ★ SQL Server 稽核
- C2 稽核模式
- Common Criteria 選項
- 登入稽核

Microsoft®
SQL Server™ 2008

資料加密

為何需要加密?

- 額外的安全性保護
- 因應法規或特需應用環境

SQL Server 2000

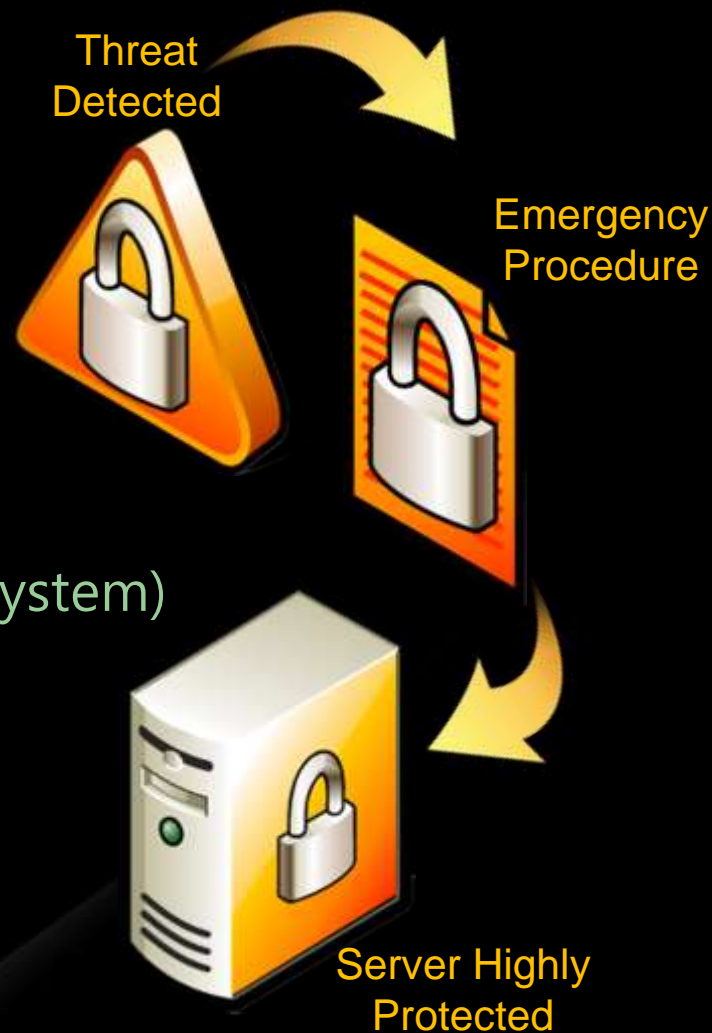
- 加密檔案系統 (EFS, Encrypted File System)
- 或者由其他廠商提供

SQL Server 2005

- 內建支援多種資料加密函數

SQL Server 2008

- 透通資料加密 (TDE, Transparent Data Encryption)
- 可延伸金鑰管理 (EKM, Extensible Key Management)



重要資料加密

📌 加密函數

- EncryptByAsmKey
- EncryptByCert
- EncryptByKey
- EncryptByPassPhrase
- HashBytes

📌 解密函數

- DecryptByAsmKey
- DecryptByCert
- DecryptByKey
- DecryptByPassPhrase

透過資料加密

企業版

SQL Server 2008

- 加密 / 解密在資料庫端
 - 用 Database Encryption Key (DEK)
- 應用程式無須再進行加密 / 解密
- DEK 用下列方式保護
 - 憑證
 - 硬體安全性模組 (HSM, Hardware Security Module)
- DEK 必須先解密才可以進行
 - 附加資料庫
 - 還原資料庫



透過資料加密應用情境

企業版



筆記型電腦安
裝SQL
Server，包括
重要的測試資
料

未經授權的使
用者取得備份
檔案或磁帶

符合安全規範
之需求，保護
機密資料

沒有先以憑證或 HSM 將 DEK
解密，資料庫就無法開啟

透通資料加密的使用考量

- 可與資料壓縮一併使用
- 不建議與備份壓縮一併使用
- 資料庫鏡像
 - 憑證要複製到鏡像伺服器上
- 當某一資料庫使用透通資料加密
 - Tempdb 資料庫也會加密，因此會對未啟用透通資料加密的其他資料庫有效能的影響
- 僅企業版支援

可延伸金鑰管理

企業版

- 使用硬體安全性模組
 - Hardware Security Module, HSM
- 廠商可針對金鑰組態、金鑰存取和HSM提供管理軟體
 - MSCAPI 提供者
- SQL EKM 金鑰提供代理功能存取 HSM 金鑰
- 使SQL Server可藉由協力廠商的模組元件支援
 - 進階加密功能與金鑰管理

可延伸金鑰管理

企業版

● 安全性

- 外部加密金鑰儲存(實體分隔資料和金鑰)
- 集中管理與儲存金鑰機制
- 額外的授權檢查
- 區分資料庫擁有者與資料擁有者

● 效能

- 硬體高效能的加密 / 解密

DEMO

透過資料加密

SQL Server 稽核

- 使用擴充的事件進行稽核
- 較SQL追蹤更有效率
- 輕鬆管理與佈署
 - T-SQL
 - SQL Server Management Studio
 - 記錄檔檢視器



建立 SQL Server 稽核之步驟

- 建立稽核物件，並定義稽核記錄存放的目標
 - 二進位檔案
 - Windows 應用程式事件記錄檔
 - Windows 安全性事件記錄檔
- 建立對應至稽核物件的伺服器稽核規格或資料庫稽核規格的動作群組或動作
- 啟用稽核規格
- 啟用稽核物件



新增稽核目的地

建立稽核

就緒

選取頁面

一般

指令碼 | 說明

稽核名稱(N): SQL Security Audit

佇列延遲 (以毫秒): 1000

☐ 於稽核記錄失敗時關閉伺服器(S)

稽核目的地(D): Security Log

檔案路徑(P): File
Security Log
Application Log

最大換用檔案(O): 2147483647

☒ 無限制(U)

檔案大小上限(Z): 0 MB GE TB(T)

☒ 無限制(L)

☐ 保留磁碟空間(R)

連接

SQL2K8C2
[SQL2K8C2\Administrator]

[檢視連接屬性](#)

進度

就緒

確定 取消 說明

新增伺服器稽核規格

建立伺服器稽核規格

就緒

選取頁面

一般

指令碼 說明

名稱(N): Server Audit

稽核(A): SQL Audit

動作:

	稽核動作類型	物件類別	物件結
1	DATABASE_OBJECT_PERMISSION_CHA...		
▶*2			
	DBCC GROUP		
	FAILED_LOGIN_GROUP		
	FULLTEXT_GROUP		
	LOGIN_CHANGE_PASSWORD_GROUP		
	LOGOUT_GROUP		
	SCHEMA_OBJECT_ACCESS_GROUP		
	SCHEMA_OBJECT_CHANGE_GROUP		
	SCHEMA_OBJECT_OWNERSHIP_CHANGE_C		

連接

SQL2K8C2
[SQL2K8C2\Administrator]

[檢視連接屬性](#)

進度

就緒

確定 取消 說明

新增資料庫稽核規格

建立資料庫稽核規格

就緒

選取頁面

一般

指令碼 說明

名稱(N): Database Audit

稽核(A): SQL Audit

動作:

	稽核動作類型	物件類別	物件結構描述
1	DATABASE_OBJECT_ACCESS_GROUP		
▶*2			
	DATABASE OWNERSHIP CHANGE GROUP		
	DATABASE PERMISSION CHANGE GROUP		
	DATABASE PRINCIPAL CHANGE GROUP		
	DATABASE PRINCIPAL IMPERSONATION		
	DATABASE ROLE_MEMBER_CHANGE GRO		
	DBCC GROUP		
	DELETE		
	EXECUTE		

連接

SQL2K8C2
[SQL2K8C2\Administrator]

[檢視連接屬性](#)

進度

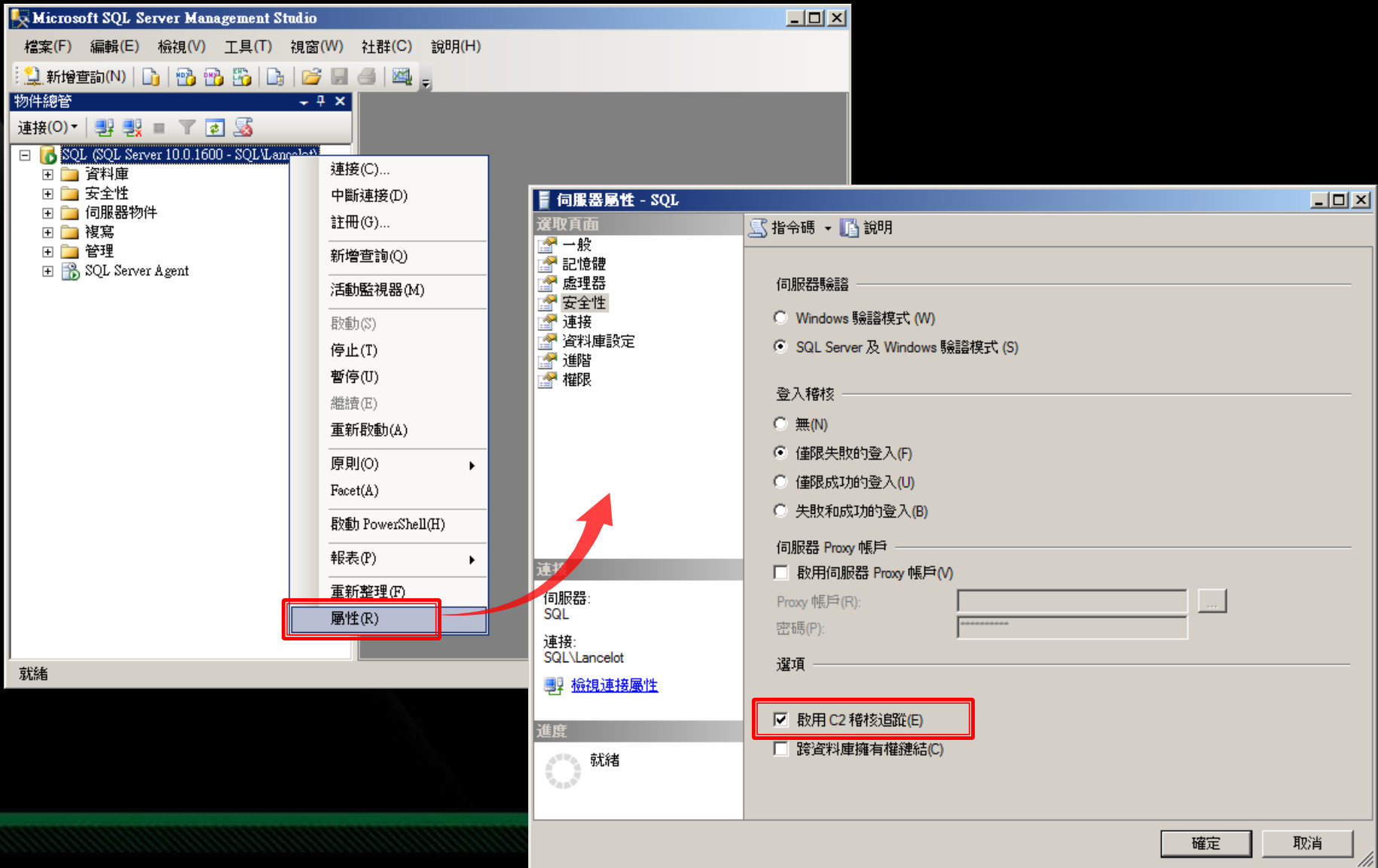
就緒

確定 取消 說明

DEMO

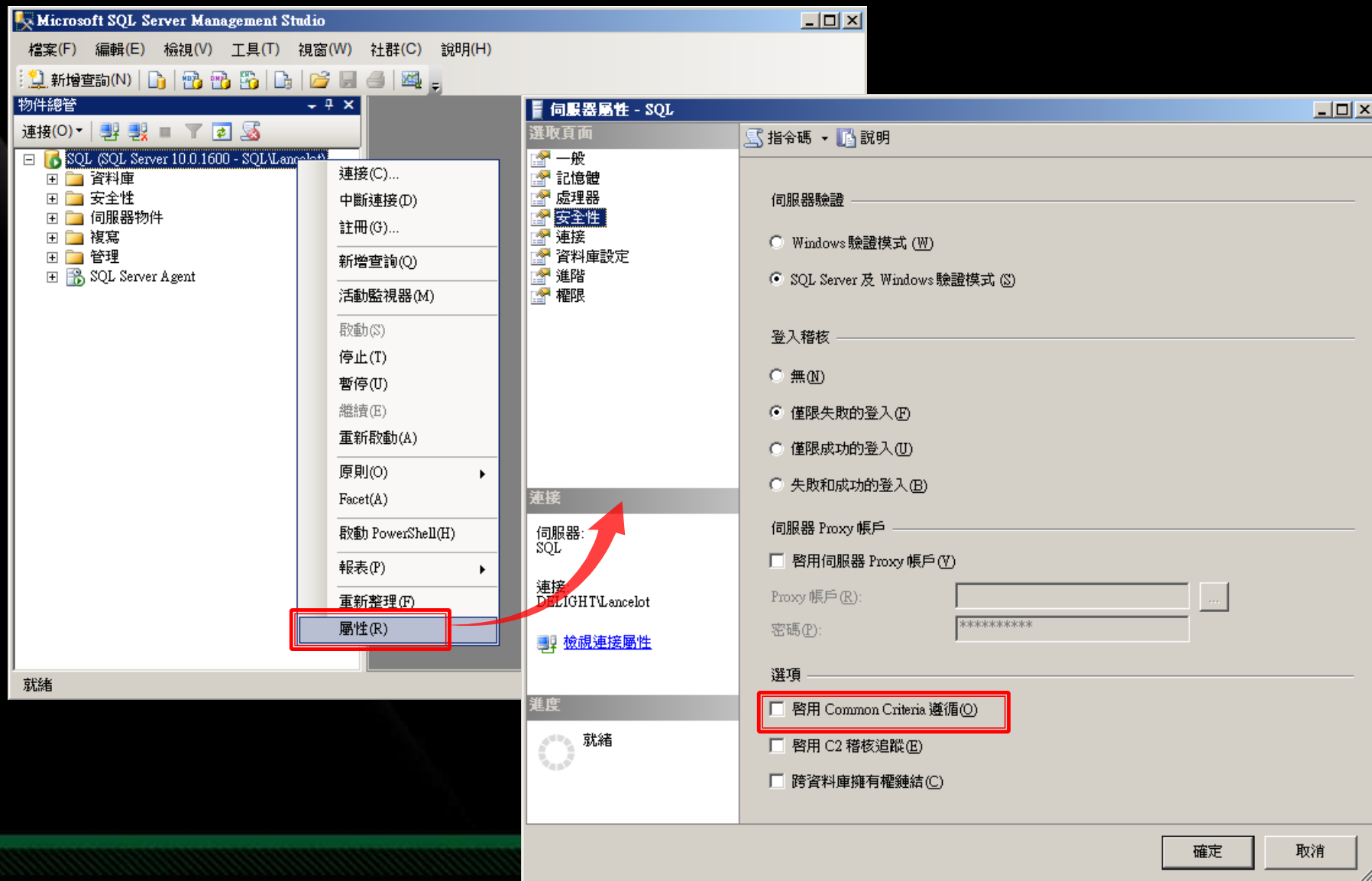
SQL Server 稽核

C2 稽核模式

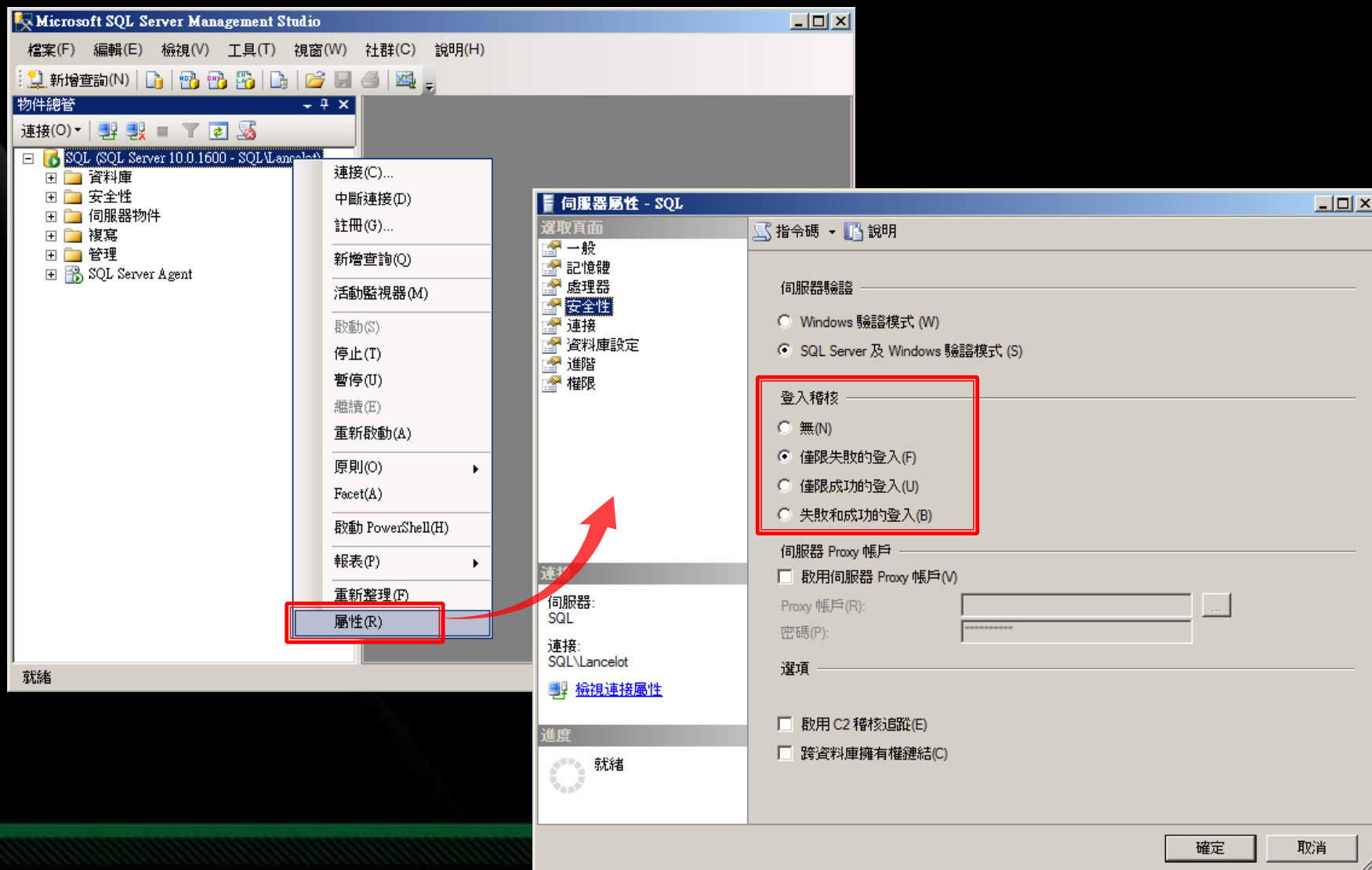


Common Criteria選項

企業版



登入稽核



SQL Server記錄檔檢視器

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. On the left, the 'SQL Server 記錄檔' (SQL Server Log) folder is selected under the 'SQL Server' instance. A context menu is open, showing options like '設定(C)...', '檢視(V)...', '報表(R)...', and '重新整理(F)'. The '檢視(V)...' option is highlighted, and a red arrow points from it to the 'SQL Server 記錄檔(S)' window.

The 'SQL Server 記錄檔(S)' window shows a list of log entries. The '記錄檔摘要: 未套用篩選(S)' (Log Summary: No filter applied) section displays a table of log entries. The table has columns for '日期' (Date), '來源' (Source), and '訊息' (Message). The entries are filtered to show login failures for user 'jack'.

日期	來源	訊息
2009/3/20 下午 04:55:29	登入	Login failed for user 'jack'. 原因: 密碼與提供的登入密碼不符。 [CLIENT: <local machine>]
2009/3/20 下午 04:55:29	登入	錯誤: 18456, 嚴重性: 14, 狀態: 8。
2009/3/20 下午 04:55:20	登入	Login failed for user 'mark'. 原因: 找不到符合所提供名稱的登入。 [CLIENT: <local machine>]
2009/3/20 下午 04:55:20	登入	錯誤: 18456, 嚴重性: 14, 狀態: 5。
2009/3/20 下午 04:54:55	登入	Login failed for user 'anne'. 原因: 密碼與提供的登入密碼不符。 [CLIENT: <local machine>]
2009/3/20 下午 04:54:55	登入	錯誤: 18456, 嚴重性: 14, 狀態: 8。
2009/3/20 下午 04:54:12	登入	Login failed for user 'anne'. 原因: 密碼與提供的登入密碼不符。 [CLIENT: <local machine>]
2009/3/20 下午 04:54:12	登入	錯誤: 18456, 嚴重性: 14, 狀態: 8。
2009/3/20 下午 04:54:07	登入	Login failed for user 'sa'. 原因: 嘗試評估密碼時發生錯誤。 [CLIENT: <local machine>]
2009/3/20 下午 04:54:07	登入	錯誤: 18456, 嚴重性: 14, 狀態: 7。
2009/3/20 下午 04:54:01	登入	Login failed for user 'sa'. 原因: 嘗試評估密碼時發生錯誤。 [CLIENT: <local machine>]
2009/3/20 下午 04:54:01	登入	錯誤: 18456, 嚴重性: 14, 狀態: 7。

The '選取的資料列詳細資料(D):' (Selected row details) section shows the details for the selected log entry:

日期	記錄檔	來源	訊息
2009/3/20 下午 04:55:29	SQL Server (目前的 - 2009/3/20 下午 04:55:00)	登入	Login failed for user 'jack'. 原因: 密碼與提供的登入密碼不符。 [CLIENT: <local machine>]

The status bar at the bottom of the log viewer shows '完成 (151 筆記錄)' (Completed (151 records)).

Microsoft®

SQL Server™ 2008 資料庫進階管理

- 企業級管理
- 高效能和高延展性
- 高可用度
- 企業級安全

Q&A

SQL Server 2008 升級保證班

- 台北,台中,高雄全面熱烈招生中 -

詳細課程內容: <http://www.microsoft.com/taiwan/sql2008/event/course/>

Microsoft SQL Server 2008 升級保證班一日營

還在忍受 SQL Server 2000 的龜速嗎?? 想改變! 升級就對了! 根據 Forrester 調查發現升級至 SQL Server 2008 為企業帶來的預期投資報酬率在 162% 至 181% 之間。而且在不更動硬體、資料庫或應用程式的情況下,系統查詢回應時間平均提升了 25%。全球最大筆記型電腦及其他運算裝置製造商之一的廣達電腦的 5TB 的 SAP 資料庫透過 SQL Server 2008 的資料壓縮功能,平均減少其資料庫儲存空間達 87% ([了解廣達電腦升級至 SQL Server 2008 之效益](#))。bwin.com 這家全球性的線上運動博弈服務公司採用了 SQL Server 2008 來處理它高達 100TB 的資料量 ([了解 BWin 在 SQL Server 2008 之成功案例](#))。

您是否擔心升級會遇到很多的問題,是否想把應用程式所需的修改與資料庫的升級計劃都事先找出來? 那麼【SQL Server 2008 升級保證班一日營】將會是一個很好的起點。

本次實作課程設計的內容,將協助測試您的 SQL Server 2000 及 SQL Server 2005 資料庫應用程式,是否可以相容於新版本的 SQL Server 2008,同時也會帶領您進行資料庫及相關服務升級之流程及步驟。


台灣微軟 SQL Server 升級小組邀請您


 [了解更多實實實用的升級資訊](#)

課程內容 (一日上機課程, 共 6 小時):

1. 如何進行升級準備及評估 (使用 SQL Server Upgrade Advisor 工具)
2. 測試 SQL Server 2000 應用程式升級至 SQL Server 2008 的執行狀況 (使用 SQL Server Upgrade Assistant 工具)
3. 開始升級 (資料庫 · Analysis Services · DTS · Reporting Services)
4. 國內其它客戶升級經驗分享

 優惠 TechNet學員
報名,特惠價只要
NT\$300元(原價
NT\$600元)

 請填好你手上報名
表交給工作人員即可

 至恆逸報到上課時
現場繳費即可