

第三章 關聯式資料庫管理系統：以*SQL Server 2008* 為例





本章內容

- 3.1 關聯式系統
- 3.2 關聯式資料庫管理系統的架構
- 3.3 關聯式系統中的查詢最佳化處理
- 3.4 系統目錄
- 3.5 使用 MS SQL Server 2008
- 3.6 資料庫的初始內容
- 3.7 卸離與回存資料庫



關聯式系統

- 關聯式系統所具有的特點：
 - 資料結構是以“關聯表”(Tables)呈現
 - 提供“關聯式代數/關聯式計算”相同表達能力的查詢介面
 - 關聯表運算的「封閉性」(Closure)
 - “一次處理一個集合”(Set-at-a-Time)
 - 使用“資料值”(Data Values)隱含地連接關聯表
 - 表格看成是由“記錄”(Records)所構成的集合



關聯式系統 (續)

- 關聯式系統所具有的特點：
 - 在邏輯上，資料是以表格方式呈現，實際儲存在磁碟上時卻可能是以循序檔案，磁碟索引結構 (如：B-trees, B⁺-trees)、雜湊表 (如：Extendible Hashing, Linear Hashing)、指標串列 (Pointer Chain)，或是經過壓縮儲存來為之
 - ANSI/SPARC 架構定義下的概念層與外部層
 - 表中的資料都是單一元素 (Atomic Values)，不可以是一個集合



MS SQL Server vs. Sybase

- MS SQL Server 最早是由 Sybase SQL Server 4.2.x 移植到個人電腦上的結果
- 兩者後來分道揚鑣
- MS SQL Server 6.0/6.5 → MS SQL Server 7.0 → MS SQL Server 2000 → MS SQL Server 2005 → MS SQL Server 2008
- Sybase SQL Server 也為了與 MS SQL Server 區別，於 1997 年改名 Sybase Adaptive Server Enterprise (Sybase ASE)



關聯式系統的共通名詞

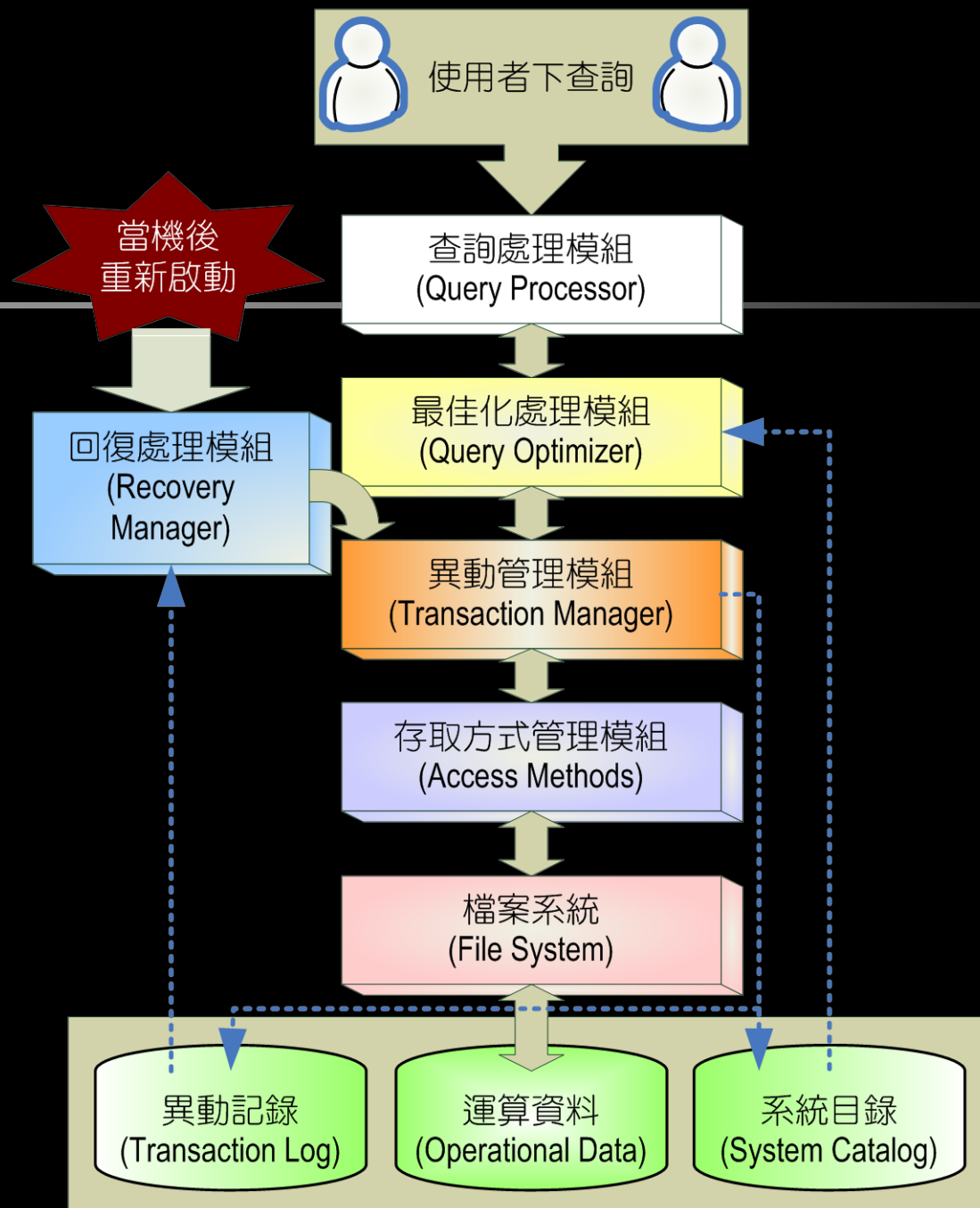
- 關聯表 (Relation) = 表格 (Table)
- 屬性 (Attributes) = 欄位 (Fields) = 欄 (Columns)
- 值組 (Tuples) = 記錄 (Records) = 列 (Rows)。
- 在學術界大多是用黃色字體的名詞來描述論文內容
(建議學生在寫論文或報告時採用黃色字體的名詞會比較正式)



本書所用之名詞對照表

原文	本書使用的名詞	坊間書籍的同義詞	SQL Server 2008 用詞
Relation	關聯表	關連表	關聯表
View	視界	景觀、概觀、視域	檢視
Domain	值域	定義域	值域
Clustering Index	叢聚索引	叢集索引	叢集索引
Stored Procedure	預儲程序	預存程序	預存程序
Trigger	觸發程序	觸發器	觸發程序
Primary Key	主鍵	主索引	索引鍵

關聯式資料庫管理系統架構





SQL Server 2008 的組成

- 主從式架構 (Client-Server Architecture)
- 多工 (Multi-Tasking)、多線程 (Multi-Threading)
- 可以穿過檔案系統，直接處理磁碟的輸出、入
- 提供查詢語言 Transact-SQL (符合 SQL-92, 以及部份 SQL 3 的功能)
 - 流程控制命令 (Control Flow Statement)
 - 使用者自訂型態 (User-Defined Data types)
 - 預儲程序 (Stored Procedures)、觸發程序 (Triggers)、使用者自訂函數等



SQL Server 2008 新功能

- 提供 DDL 觸發程序，設定物件結構被更改時的主動通知機制
- 可以透過 CTE (Common Table Expression) 功能，撰寫具遞迴 (Recursive) 能力的函數。
- 增加 XML 資料型態與相關資料的存取功能。
- 還提供了 XQuery 的語法，並與 SQL 指令結合，讓使用者得以直接存取 XML 文件中的元素與屬性資料。
- 提供 PIVOT/UNPIVOT 運算子：處理類似 Excel 中的樞紐分析表功能
- 結構化錯誤處理：透過 Try...Catch 邏輯區塊，捕捉錯誤代碼



SQL Server 相關輔助工具

- SQL Server 伺服器的組態管理工具：SQL Server 組態管理員 (Configuration Manager)。
- SQL Servers 管理介面：SQL Server Management Studio
- 監看、調整效能的工具：SQL Server Profiler、Database Engine Tuning Advisor。
- 直接下達 SQL 命令的介面：整合在 SQL Server Management Studio 中的 Query Analyzer。
- 產生報表：Report Builder 與 Reporting Service



SQL Server 相關輔助工具 (續)

- 資料庫處理函式庫，如：DB-Library 或 OLE DB Provider。
- 資料倉儲與資料採礦的建置介面：SQL Server Business Intelligence Development Studio。
- 大量資料匯入、匯出的功能—透過 SQL Server Business Intelligence Development Studio 中的 SQL Server Integration Service (SSIS) 功能來完成。
- 大量資料庫匯入、匯出工具指令：bcp (bulkcopy)



使用 bcp 的注意事項

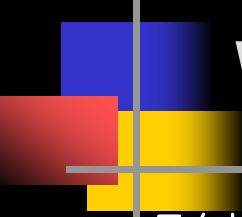
- 它是一個在作業系統層次下使用的命令
- 做舊資料轉換時非常有用
- 使用 bcp 時 SQL Server 並不會做異動記錄，使用 bcp 前最好先將該資料庫的 'select into/bulk copy' 選項設為 true，例如：

```
sp_dboption bob, 'select into/bulkcopy', true  
go
```



SQL Server 2008 的安全機制概念

- 「主體」(Principal)：要求使用 SQL Server 資源的個體、群組或程序。
- 「安全實體」(Securable)：SQL Server 使用存取權限加以保護的資源個體。
- 「結構描述」(Schema)：由資料庫物件所構成的集合，這個集合是一個「命名空間」(Namespace)，集合中每個元素的名稱都必須唯一。



Why Schema (結構描述) ?

- 「結構描述」(Schema) 與本書第一章 1.6 節所提到的各層資料的定義—「綱要」(Schema) 有所不同，但是它們使用同一個單字 Schema 。
- 「結構描述」是由資料庫物件構成的集合 (命名空間—Namespace)，此集合中每個元素的名稱都必須是唯一的。
- 一個物件的擁有者是包含該物件的「結構描述」：
 - 「結構描述」與「資料庫使用者」兩者間互相獨立，以簡化 DBA 工作
 - 因為管理者刪除某個資料庫使用者時，不需更改其物件的擁有者
 - 多個使用者可以擁有同一個結構描述，一個使用者也可以擁有多個結構描述。



資料庫物件的完整命名方式

- 在 SQL Server 2008 的完整命名格式 (Four-Part Naming Convention) : *Server.Database.Schema.Object*
- 「預設結構描述」 (Default Schema) 的概念，讓物件可以使用不完整的名稱表示：e.g., [Server_1].[DB_1].[Table_1]。
- 系統一般會以 dbo 當成預設結構描述 (其意思是代表 **d**atabase **o**wner 這個結構描述)。
- SQL Server 2008 的預設作法：把同一部伺服器名稱當做第一個部份，預設資料庫當做第二部份，預設的結構描述當做第三部份 (若無則使用 dbo)，因此直接簡寫成 [Table] 亦可。



SQL Server 2008 安全機制的目標

- 「最少權限原則」(The Principle of Least Privilege)：使用者與應用程式應該在存取的物件上，設定符合需求的最小權限，
- 導正過去程式設計師從頭到尾都使用 'sa' (系統管理員) 權限，暴露安全漏洞的弊病。



主體 (Principal) 的種類

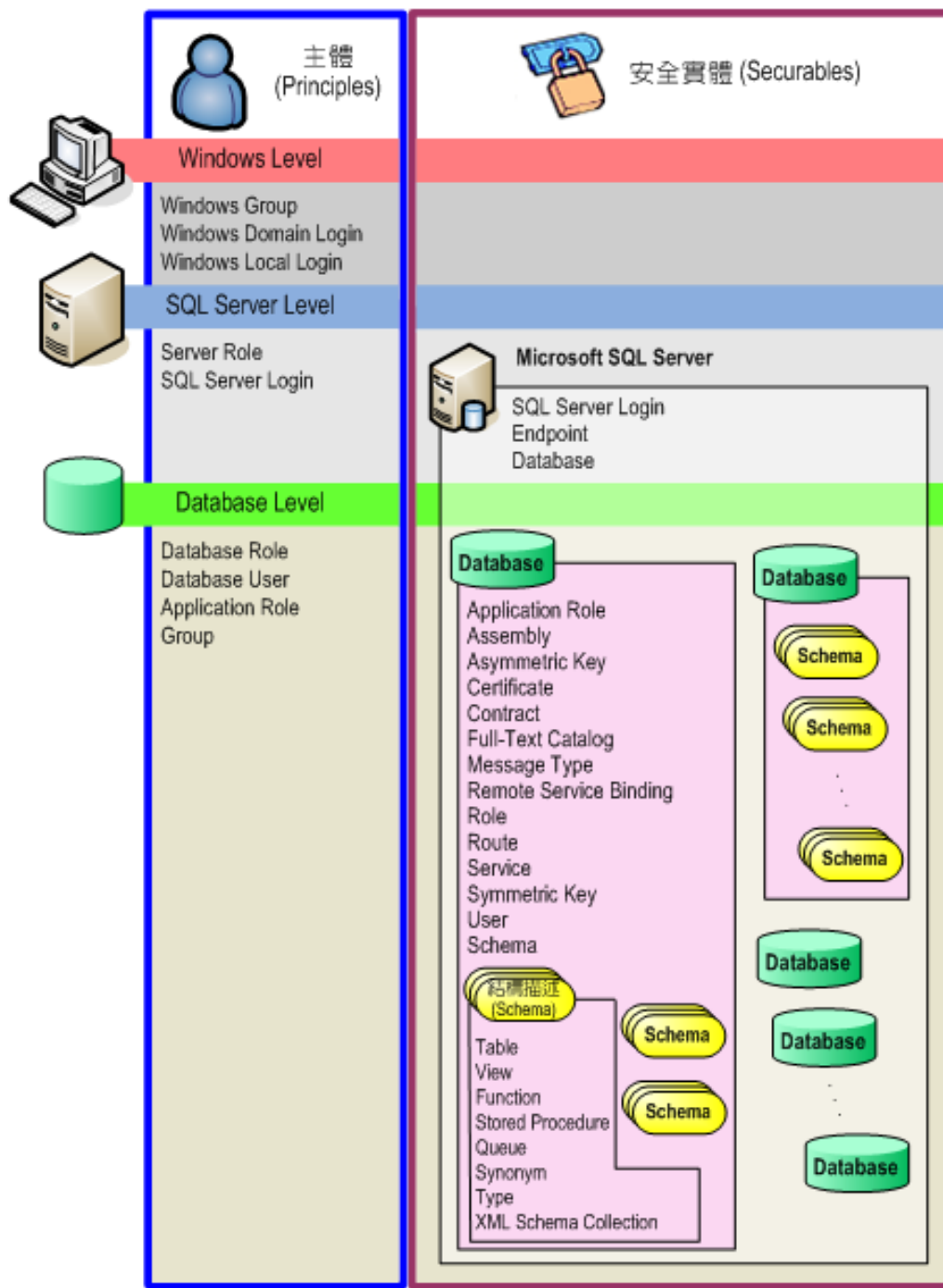
- 「主體」(Principal) 分成以下三種層級：
 - Windows 層級 (Windows Level) : Windows Group、Domain User Account，以及 Local User Account
 - 資料庫伺服器層級 (SQL Server Level) : Login、Server Role 等
 - 資料庫層級 (Database Level) : Database User、Database User/Role、Application Role，以及 Group 等。
- 可以查詢 `sys.server_principals` 及 `sys.database_principals` [系統檢視] 取得上述主體的資訊



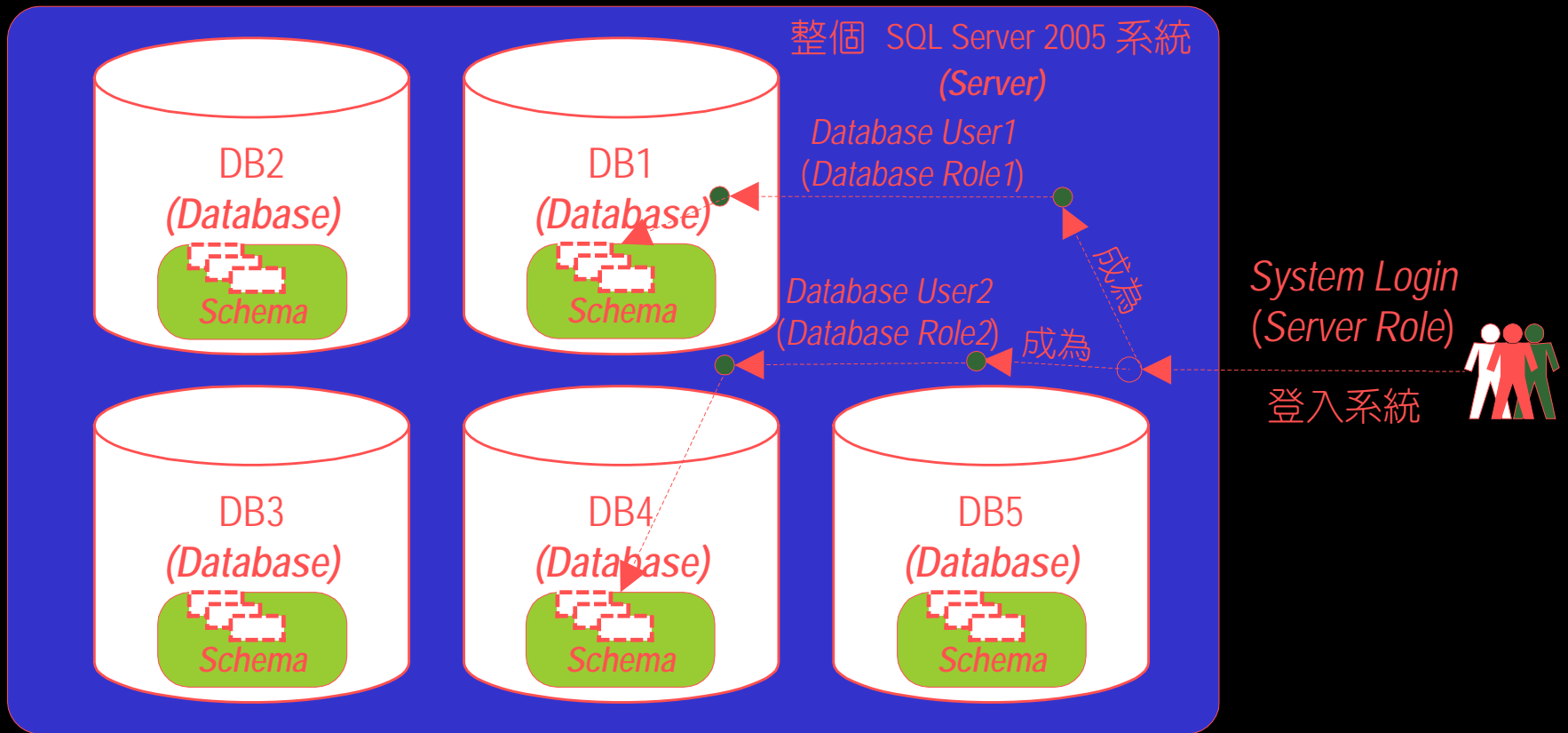
安全實體 (Securable) 的種類

- 「安全實體」 (Securable) 分成以下三種層級：
 - **Server** Securable Scope：資料庫、System Login、Server Role 等
 - **Database** Securable Scope：結構描述 (Schema)、Database User、以及 Database Role，或是 Application Role 等
 - **Schema** Securable Scope：類型 (Type)、XML結構描述集 (XML Schema Collection)、物件 (Object)：包含關聯表、視界、預儲程序等。

主體與安全實體的關係



SQL Server 2008 的安全機制



SQL Server 2008 安全機制總覽





查詢最佳化處理

- SQL (只要指出 “What”) 的抽象層次高於 C, COBOL (必須指出 “How”) 所以，最佳化非常重要
- 搜尋時的處理方式：
 - 循序搜尋 (Sequential Search)：資料量很小時適合
 - 利用索引 (Index)：資料量很大的時候，通常利用索引會較快，但不絕對是。
 - 利用索引須要多存取一次磁碟 (或記憶體)
 - 新增、刪除，或修改時則一定會降低效能



查詢最佳化處理 (續)

- 定性查詢最佳化處理：
 - 利用關聯式代數做查詢轉換
 - 與關聯表的大小並無多大關係
 - 用更簡化或更快速的運算式來取得結果
- 定量查詢最佳化處理：
 - 與關聯表目前的大小有很大的關係
 - 希望將某些內含大量值組的關聯表先行過濾
 - 使用者無法處理，必須由系統透過系統目錄評估



定量查詢最佳化的考量

- 先決定需要用到那些關聯表以取得資料
- 這些關聯表有多大，需要透過它們的索引嗎？
- 目前有那些索引可用？如何利用它們？
現存索引對於加速查詢處理的幫助有多大？
- 資料是否 “叢聚” (Clustering) 於磁碟上？
 - 將資料在邏輯上與實際在磁碟中的順序保持一致
 - 叢聚索引在任何關聯表中最多只能有一個



叢聚索引 (Clustering Index)

儘量將資料按順序放在磁碟的同一頁 (Page) 中

1	三國演義	羅貫中	120	古文出版社
2	水滸傳	施耐庵	170	中庸出版社
3	紅樓夢	曹雪芹	170	春秋出版社
4	西遊記	吳承恩	140	聊齋出版社



非叢聚索引

資料按不見得按順序放在磁碟的同一頁 (Page) 中

3 紅樓夢 曹雪芹 170 春秋出版社

2 水滸傳 施耐庵 170 中庸出版社

1 三國演義 羅貫中 120 古文出版社

4 西遊記 吳承恩 140 聊齋出版社



系統目錄

- 在 SQL Server 中整個系統的總系統目錄便是系統內建的 master 資料庫
- 系統目錄的資料又稱為 metadata，意思就是「用來描述資料的資料」
- 系統目錄僅限於查詢動作，不可任意新增、刪除或更改內容，因此 SQL Server 2008 將大多數的系統目錄都藏起來，並提供 INFORMATION_SCHEMA.xxxxxx 的一系列系統檢視讓使用者查詢
- 使用者 (非建立者) 對關聯表綱要 (Schema) 不熟悉時，便可透過 SQL 查看系統目錄的資料



系統目錄的內容

- 使用者的帳號、密碼、安全規則等
 - 讀取該資料庫 (關聯表) 中的資料
 - 新增、刪除或更改資料庫 (關聯表) 中的資料
 - 建立新資料庫，或刪除舊資料庫的權利
 - 資料庫上載 (Upload)、下載 (Dump, Download) 或備份 (Backup)
- 系統內定的資料型態以及所有由使用者自訂的資料型態。



系統目錄的內容(續)

- 資料庫中每個關聯表的：
 - 名稱、建立者名字、欄位個數、記錄筆數—其中記錄筆數是最佳化處理時的重要參考。
 - 欄位名稱、欄位的資料型態、欄位值的限制條件。
 - 主鍵、外鍵、索引名稱、以及索引欄位等資訊。



系統目錄的內容(續)

- 資料庫中每個視界的：定義、名稱、建立者名字、欄位個數、欄位名稱、欄位的資料型態，以及所參考到的關聯表名稱等資訊。
- 資料庫中每個預儲程序和觸發程序的：定義、所參考到的關聯表（或視界）名稱，以及所參考到的欄位名稱等資訊。



系統目錄的內容(續)

- 記錄資料庫中所有關聯表、視界的名稱 (*name*)、建立者 (*creator*)、維度 (*degree*) 等資訊的關聯表，以圖 4.2 的 BOB 為例，sysobjects內容如下 (在 SQL Server 2008 中，可透過查詢 `INFORMATION_SCHEMA.TABLES` 取得)：

<i>Table_name</i>	<i>creator</i>	<i>degree</i>	...
Bookstores	Frank	4	...
Books	Frank	5	...
Orders	Frank	3	...



系統目錄的內容 (續)

- 記錄資料庫中每個關聯表、視界的所有屬性名稱 (*name*)、資料型態 (*coltype*) 等資訊的關聯表，
- 以 BOB 資料庫為例，可查詢 `INFORMATION_SCHEMA.Columns` 內容取得如下圖所示：



<i>Column_name</i>	<i>Table_name</i>	<i>Column_type</i>	...
no	Bookstores	int	...
name	Bookstores	char	...
rank	Bookstores	tinyint	...
city	Bookstores	char	...
id	Books	int	...
bookname	Books	char	...
author	Books	char	...
price	Books	smallint	...
publisher	Books	char	...
no	Orders	int	...
id	Orders	int	...
quantity	Orders	int	...



系統目錄的內容 (續)

- 記錄資料庫中每個關聯表的所有索引名稱 (*name*)、索引欄位數 (*colcount*) 等資訊
- 以 BOB 資料庫為例，可查詢 `sys.indexes` 或 `sys.sysindexes` 內容取得類似下圖所示:

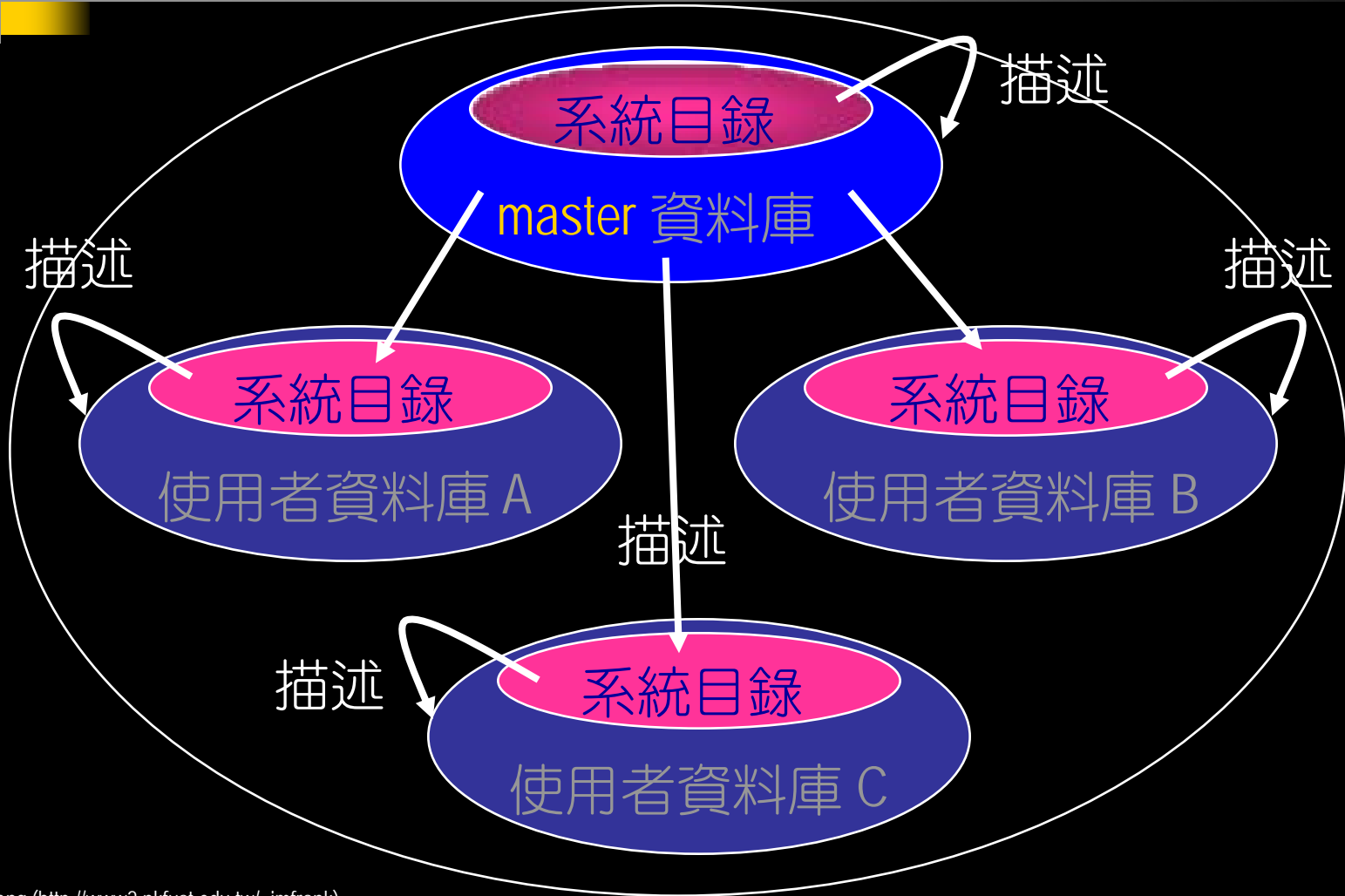
<i>name</i>	<i>tbname</i>	<i>colcount</i>	...
Xn	Bookstores	1	...
Xrn	Bookstores	2	...
Xi	Books	1	...
Xni	Orders	2	...



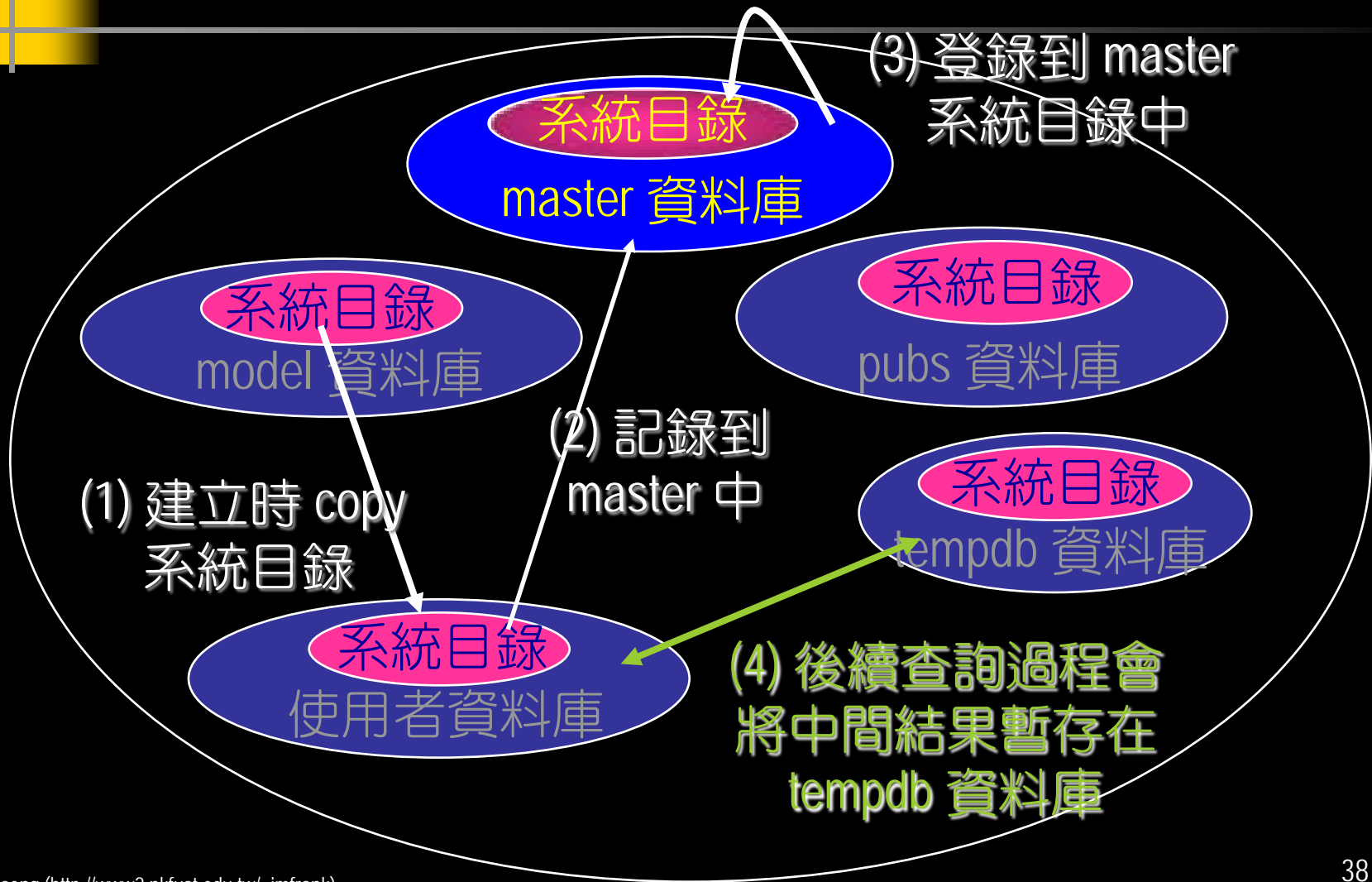
SQL Server 系統資料庫

- **master** 資料庫：存放描述所有資料庫的資料，包括 master 資料庫本身，可以說是所有資料庫所屬系統目錄的資料庫，等於是一個 meta-database，也就是整個系統的總系統目錄。
- **model** 資料庫：內含一般資料庫的基本綱要
- **msdb** 資料庫：用來存放微軟相關產品的整合資料，在此我們不討論它。
- **tempdb** 資料庫：系統用來儲存暫時關聯表

SQL Server 2008 系統目錄



系統資料庫 vs. 使用者資料庫





SQL Server 2008 系統目錄

- 每一個資料庫有一個系統目錄 (但是已經隱藏起來)

- 整個資料庫系統有一個總系統目錄 (master)

- 所有屬於某資料庫的系統目錄以 sys 開頭

sysobjects

syslogs

sysprotects

sysprocedures

sysusers

syssegments

systypes

syscolumns

...

...

Master 資料庫

整個系統
的系統資料庫

系統目錄 (隱藏起來了)

系統檢視

sysobjects
sysprotects
sysusers

syslogs
sysprocedures
syssegments

存放描述 master 資料庫的資料

系統檢視
INFORMATION_SCHEMA

sys.servers

syslogins

sysdevices

syslanguages

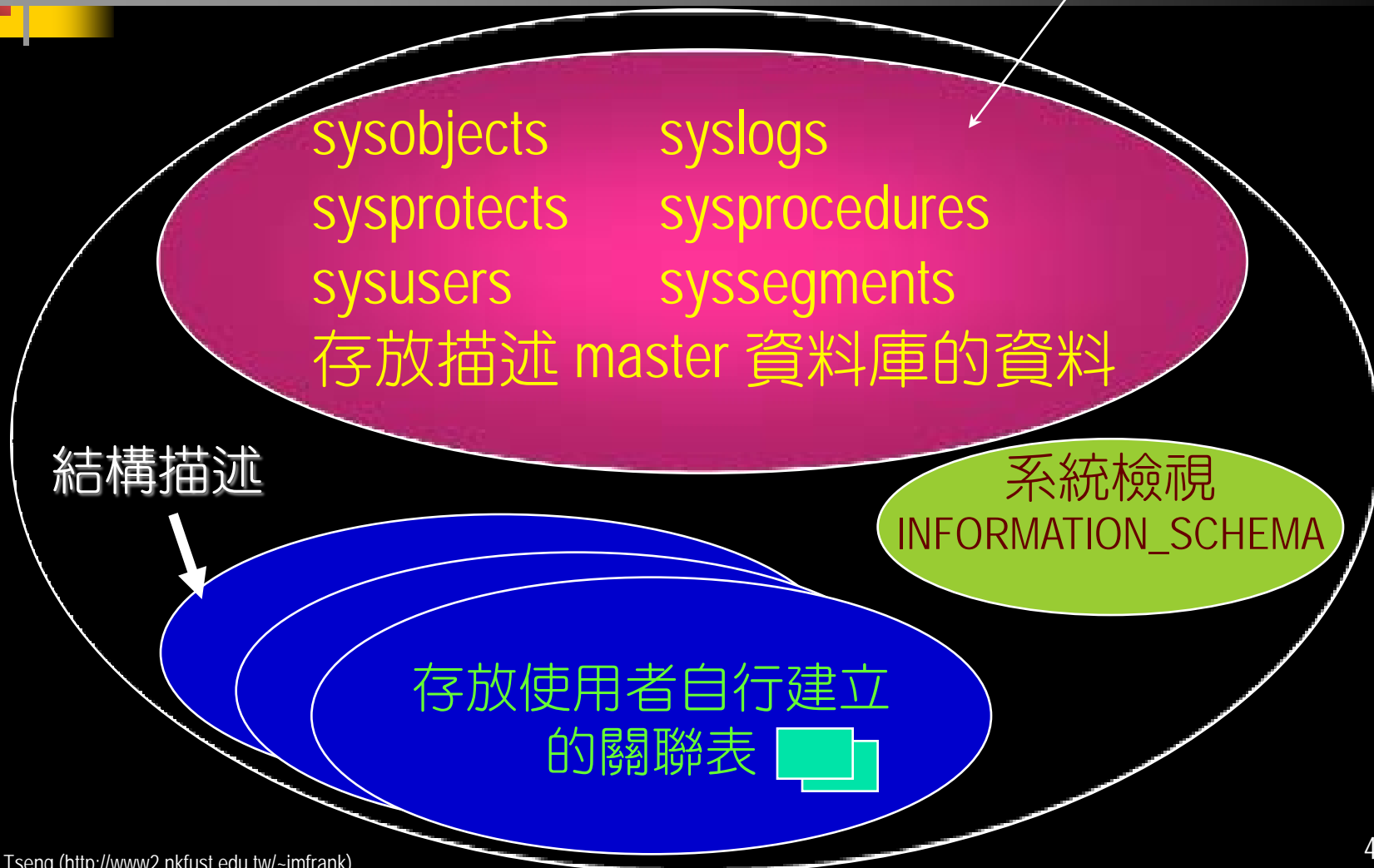
sysprocesses

sysmessages

存放描述整個資料庫系統的資料

一般使用者資料庫

系統目錄 (隱藏起來了)





系統出問題時 Rebuild Master 資料庫

- 將 master 重新建立，可以讓系統重新開始
- 注意：會將系統中的一般資料庫完全被清除 (效果相當於 format 整體資料庫)
 - 先用 SQL Server Configuration Manager 停止所有 SQL Server 2008 服務，然後由 DOS 下執行
“`sqlserver.exe -m`” 以啟用 Single User Mode
 - 在 Dos 下執行：`start /wait setup.exe /qn`
`INSTANCENAME=MSSQLSERVER REINSTALL=SQL_Engine`
`REBUILDDATABASE=1 SAPWD=<New SA Password>`
 - 重新由 SQL Server Configuration Manager 啟動需要的 SQL Server 2008 服務



使用 MS SQL Server

- MS SQL Server 2008 Enterprise Evaluation Edition 可以安裝在 Windows XP-SP2 與 Windows 2003 Server-SP2 以上。
- MS SQL Server 2008 比起先前的版本, 在功能與效能上有很大的變革, 以下分三大領域探討：
 - 「企業資料管理」領域
 - 「開發人員生產力」領域
 - 「商業智慧」(Business Intelligence, BI) 領域



「企業資料管理」領域的新功能

- Database Mirroring、Failover Cluster、Database Snapshots、Row Versioning、Snapshot Isolation、Native Partitioning 及 Service Broker 等
- 與 Visual Studio 2008 整合式的管理介面與功能
- 支援 64 位元硬體。
- 高安全性 (High Security)：所有與外部互通的機制預設都是關閉的，確保整體安全性
- 主動通報服務 (Notification Service)



「開發人員生產力」的新功能

- 以 Common Language Runtime (CLR) 整合任何 Microsoft .NET 語言來開發資料庫物件，透過「組合」(Assembly) 的建構機制，延伸資料庫的功能。
- 與 Visual Studio 2008 開發環境完全整合的環境
- 與 XML 的整合功能：支援 XML 資料型態，並透過 XQuery 語法來儲存、管理和分析資料。
- 支援 HTTP、SOAP，以及 XML 結構描述定義 XSD (XML Schema Definition) 等 Web Service 開發標準
- Transact-SQL 功能的增強：如本書第七章所述。



「商業智慧」領域的新功能

- 新的資料轉換與整合工具 (SQL Server Integration Service, **SSIS**)：分開定義 Dataflow 與 Control Flow 的介面
- 新的資料倉儲建置介面 (SQL Server Analysis Service, **SSAS**)：Unified Dimensional Model (UDM)、Key Performance Indicators (KPI)、OLAP 強化
- 九種資料採礦 (**Data Mining**) 演算法。
- 報表服務 (Reporting Services)：**Reporting Designer, Reporting Builder**
- 主動通報服務 (**Notification Service**)



SQL Server 2008 版本

- 除了有 32 位元與 64 位元之分以外，依功能、價格大致上可以區分為
 - 快速版 (Express Edition)：免費
 - 標準版 (Standard Edition)
 - 工作群組版 (Workgroup Edition)
 - 程式開發版 (Developer Edition)
 - 企業版 (Enterprise Edition)
 - 還有其他版本 (像 SQL Server 2008 Web) 無法一一詳述
- 企業評估版 (Enterprise Evaluation Edition)：可以安裝在 Windows XP 的環境中 (本書採用此版為範例說明)。



安裝 SQL Server 2008

- 建議電腦至少要有 512 MB 以上的記憶體空間 (最好有 2GB)
- 到 [http:// www.microsoft.com /downloads/](http://www.microsoft.com/downloads/) 尋找並下載，
- 下載前需先註冊。或使用 hotmail、msn，或 Microsoft Passport 帳號登入後下載。
- 有 32 位元的 x86 版本與 64 位元的 IA64、x64 版本之區分
- 若想模擬或燒錄成為 DVD 光碟後安裝，請下載 ISO 格式檔
- 如果想直接從硬碟安裝的話，請下載 exe 格式檔。
- 下載後，請同學自行參考本書所附光碟的檔案：
“\安裝SQL Server 2008介紹\如何安裝_啟動_及建立資料庫於SQL Server 2008.pdf” 檔案。
- 於下週上課前完成軟體下載與安裝試用

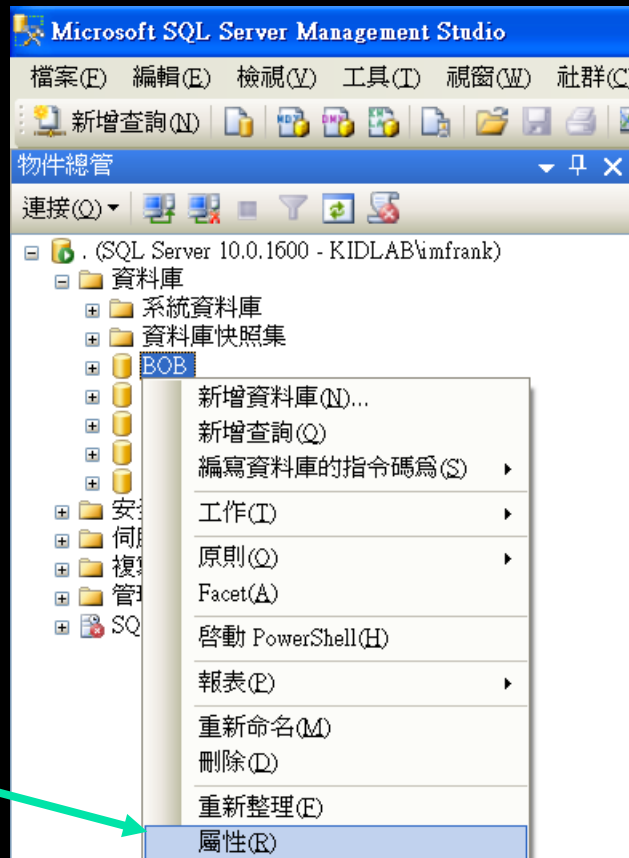


在 SQL Server 2008 建立資料庫

- 要先取得建立資料庫的權限
- 開啟 SQL Enterprise Manager 建立資料庫
- 在 [Databases] 處按滑鼠右鍵選取 [New Database...], 建立畫面出現後請輸入 BOB
- 系統會自動地將資料庫與其異動記錄所對應的檔案分別命名為 BOB_Data.mdf 與 BOB_Log.ldf

設定資料庫的性質

- 請先依照 3.5.4 節建立資料庫 BOB (內容可以是空的)



設定資料庫的性質

資料庫屬性 - BOB

選取頁面

- 一般
- 檔案
- 檔案群組
- 選項
- 變更追蹤
- 權限
- 擴充屬性
- 鏡像
- 交易記錄傳送

連接

伺服器:

連接:
KIDLAB\imfrank

檢視連接屬性

進度

就緒

指令碼 說明

定序(O): Chinese_Taiwan_Stroke_CI_AS

復原模式(M): 完整

相容性層級(L): SQL Server 2008 (100)

其他選項(O):

☒ Service Broker

Broker 已啟用	False
Service Broker 識別碼	03c83652-2851-479a-a070-e8d81fa85b95
接受 Broker 優先權	False

☒ 自動

自動更新統計資料	True
自動非同步更新統計資料	False
自動建立統計資料	True
自動壓縮	False
自動關閉	False

☒ 其他

ANSI NULL 預設值	False
ANSI NULLS 已啟用	False
ANSI 填補已啟用	False
ANSI 警告已啟用	False
VarDecimal 儲存格式已啟用	True
已啟用日期相互關聯最佳化	False
已啟用跨資料庫擁有權鏈結	False
引號識別碼已啟用	False
可信任	False
串連 Null 產生 Null	False
參數化	簡單
算術中止已啟用	False
遞迴觸發程序已啟用	False
數值捨入中止	False

☒ 狀態

ANSI NULL 預設值

確定 取消



資料庫上的選項說明

- “select into/bulkcopy” 選項
- “Columns Null by Default” 選項
- “No Checkpoint on Recovery” 選項
- “Truncate Log on Checkpoint” 選項
- “Single user” 選項
- “DBO Use Only” 選項
- “Read Only” 選項



資料庫選項說明

- “Recursive triggers” 選項
- “Torn page detection” 選項
- “Auto close” 選項
- “Auto shrink” 選項
- “Auto create statistics” 選項
- “Auto update statistics” 選項



資料庫選項的查詢

- `select databasepropertyex('BOB', 'Collation')`
- `select databasepropertyex('BOB', 'IsAutoCreateStatistics')`
- `select databasepropertyex('BOB', 'IsAnsiNullDefault')`
- `select databasepropertyex('BOB', 'IsAnsiNullsEnabled')`
- `select databasepropertyex('BOB', 'IsAutoClose')`
- `select databasepropertyex('BOB', 'IsAutoShrink')`
- `select databasepropertyex('BOB', 'IsAutoUpdateStatistics')`
- `select databasepropertyex('BOB', 'IsCloseCursorsOnCommitEnabled')`
- `select databasepropertyex('BOB', 'IsTornPageDetectionEnabled')`
- `select databasepropertyex('BOB', 'Status')`
- `select databasepropertyex('BOB', 'Recovery')`
- `select databasepropertyex('BOB', 'Updateability')`
- `select databasepropertyex('BOB', 'UserAccess')`
- `select databasepropertyex('BOB', 'Version')`



資料庫的初始內容

- 系統目錄 (建好資料庫後就會自動由 model 資料庫 copy 一份 Schema 到資料庫中)
- 隨後可以再增加
 - 關聯表 (Relations)
 - 索引 (Indexes)
 - 視界 (Views)
 - 預儲程序 (Stored Procedures) 或使用者自訂函數 (User-Defined Functions)
 - 觸發 (Triggers)
 - 資料使用規則 (Rules) 與預設值 (Defaults)
 - 使用者自訂型態 (User-Defined Types) 等



卸離與回存資料庫

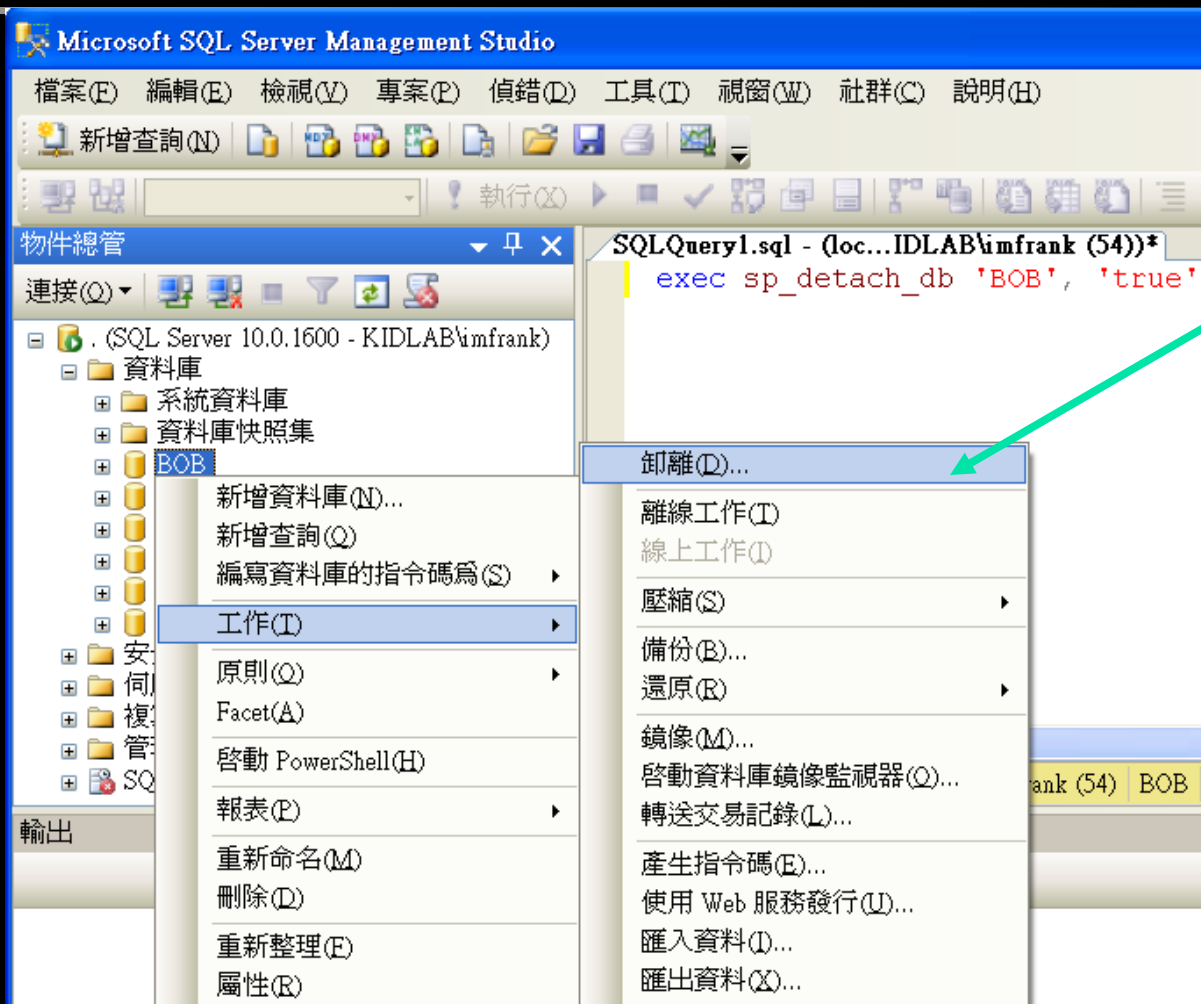
- 建好資料庫之後，如果要把整個資料庫搬到另一台伺服器上時，使用卸離與回存最方便
 - 先「卸離」(Detach) 資料庫。
 - 接著將兩個資料檔：主要資料檔 (Master Data File) *database_name_Data.mdf* 和異動記錄檔 (Log Data File) *database_name_Log.ldf*，複製到目的伺服器上。
 - 最後，將上述的兩個檔案「回存」(Attach) 到新伺服器上的 SQL Server 2000 即可



「卸離」資料庫的方法

- 「卸離」資料庫的方法有兩種，可以在Query Analyzer 中以 `sp_detach_db` 指令來完成
 - `EXEC sp_detach_db 'BOB', 'true'`
- 也可以在 Enterprise Manager 中手動執行。在欲卸離的資料庫上，按下滑鼠右鍵選取 [所有工作(K) ▶ | Detach Database...]
- 卸離後，可以將資料庫所佔用的實際檔案 (包含資料檔與異動記錄) 複製到別台 Server 上「回存」使用

「卸離」資料庫的方法



兩種方法皆可



「回存」資料庫的方法

- 「回存」資料庫的方法有兩種，可以在Query Analyzer 中以 Create database 指令來完成：
 - create database BOB on (Filename = 'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\BOB.mdf') for attach
- 也可以在 Enterprise Manager 中手動執行。如下頁圖示。

「回存」資料庫的方法





以上動作也可以用來取代備份

- 以往的備份動作也可以用 “卸離與回存資料庫” 來取代之 (如果系統允許短暫停止服務的話)
- 因為卸離後的資料庫檔案可以透過壓縮來節省空間，或傳送到另一部伺服器上
- 若所回存的資料庫是以 SQL Server 2000 所建立的，則回存後可以設定其相容層級如下：

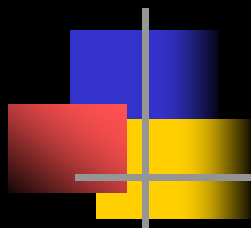
```
alter database [BOB] set compatibility_level = 100
```

或使用舊指令 `exec sp_dbcmtlevel 'BOB', '100'`

-- 100 代表 SQL Server 2008 版相容資料庫

-- 90 代表 SQL Server 2005 版相容資料庫

-- 80 代表 SQL Server 2000 版相容資料庫



本章結束
The End.