

SQL 教學講義

2019 暑期工作坊

張庭禎、涂明蕙

目錄

第 1 章 認識資料庫與 SQL	3
第 1 節 資料庫簡介	3
第 2 節 介紹 SQL 語言	5
第 3 節 MS SQL server 的好處	5
第 2 章 SSMS、資料庫結構	6
第 1 節 SSMS 介面介紹與操作	6
第 2 節 建立存放資料庫的資料夾	9
第 3 節 Sample 資料庫 AdventureWork2017	10
第 4 節 範例資料庫 AdventureWork2017 簡介	15
第 5 節 主鍵(Primary Key)、外鍵(foreign key)	17
第 3 章 SQL 基本指令	19
第 1 節 基本資料格式	19
第 2 節 基本查詢指令	20
2.1 SQL 運算子 Operators	20
2.2 SQL 基本語法	21
第 3 節 進階查詢指令	26
3.1 資料表合併	26

3.2 Grouping	31
3.3 子查詢	32
3.4 新增欄位	34
3.5 修改欄位屬性.....	35
3.6 刪除欄位	35
3.7 填入欄位	35
3.8 建立資料表	36
3.9 匯入 csv	36
第 4 章 實例操作	37

第 1 章 認識資料庫與 SQL

第 1 節 資料庫簡介¹²³⁴⁵⁶

資料庫(Database, DB)是一個存放龐大資料的地方，裡面是由各式各樣的資料匯集而成；而資料庫管理系統(Database Management System, DBMS)則提供了使用者一個環境，讓使用者在不需要了解資料庫內部實際運作，也能有效率且方便的對資料庫進行管理。

一般我們在處理"資料"時，可能只需要 Excel 就已足夠，但是當資料量愈趨龐大，Excel 等軟體已不足以支援，"資料庫"因應而生。

資料庫有以下的優點：

- (1) 降低資料的重複性(Redundancy)，讓每個資料表都設計成有自己的功能。
- (2) 達成資料的一致性(Consistency)，更動資料表內容，由關聯規則使全部更動
- (3) 達成資料的共享性(Data Sharing)，不只共享，還可以有權限地管理。
- (4) 達成資料的獨立性(Data Independence)
- (5) 達成資料的完整性(Integrity)
- (6) 避免紙張與空間浪費(Reduce Paper)
- (7) 達成資料的安全性(Security)

當然，有優點就存在著缺點：

- (1) 資料庫管理系統(DBMS)的成本較高
- (2) 資料庫管理師(Database Administrator, DBA)專業人員較少
- (3) 當 DBMS 發生故障時，比較難復原(集中控制)

¹ <https://www.inside.com.tw/2015/03/19/big-data-5-must-have-skills>

² <http://mirlab.org/jang/books/asp/databaseIntro.asp?title=17-1%20%B8%EA%AE%C6%AEw%C2%B2%A4%B6>

³ http://cc.cust.edu.tw/~ccchen/doc/db_01.pdf

⁴ <http://wiki.mbalib.com/zh-tw/%E6%95%B0%E6%8D%AE%E5%BA%93>

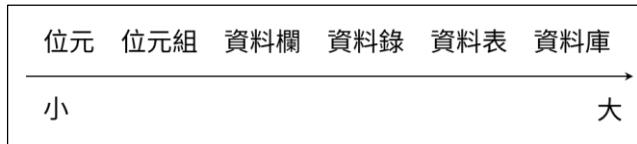
⁵ <http://www.wkb.idv.tw/study/dbase/>

⁶ <https://tw.answers.yahoo.com/question/index?qid=20061227000016KK04298>

(4) 提供安全性、同步控制、復原機制與整合性，花費較大量的資源。

資料庫的階層是有循序的關係，也就是由小到大的排列，其最小的單位是 Bit (位元)，而最大的單位則是 Data Base (資料庫)。資料依其單位的大小與相互關係的層次如下：

Bit(位元) → Byte(字元) → Field(資料欄) → Record(資料錄) → Table(資料表)
→ Data Base(資料庫)



資料庫是由許多資料表所組成，每一個資料表則由許多筆記錄所組成，每一筆記錄又以許多欄位組合而成，每一個欄位則存放著一筆資料。

資料庫中的每一個欄位，皆只能存放一筆資料，這些資料必須遵守著一定的結構標準來記錄各種訊息。例如：文字、數字或日期等格式，而在資料表中的欄位值也可能是空值(Null)。

資料階層	階層描述	資料範例
位元(Bit)	1. 數位資料最基本的組成單位 2. 二進位數值	0或1
位元組(Byte)	1. 由8個位元所組成 2. 透過不同位元組合方式可代表數字、英文字母、符號等，又稱為字元(character) 3. 一個中文字元是由兩個位元組所組成	10100100
欄位(Field)	1. 由數個字元所組成 2. 一個資料欄位可能由中文字元、英文字元、數字或符號字元組合而成	學號
資料錄(Record)	1. 描述一個實體(Entity)相關欄位的集合 2. 數個欄位組合形成一筆記錄	個人學籍資料
資料表(Table)	由相同格式定義之紀錄所組成	全班學籍資料
資料庫(Database)	由多個相關資料表所組成	校務行政資料庫，包括：成績資料表、學籍資料表、選課資料表…等
資料倉儲(Data Warehouse)	1. 整合性的資料儲存體 2. 內含各種與主題相關的大量資料來源 3. 可提供企業決策性資訊	教育部的全國校務行政資料倉儲，可進行彙整分析 提供決策資訊

以紀錄為主的資料模式，可以分成三種⁷：(1)階層式資料模式(Hierarchical Data Model)、(2)網路式資料模式(Network Data Model)、(3)關聯式資料模式(Relational Data Model)。有興趣的同學可以再自行找資料閱讀。

⁷ <http://faculty.stust.edu.tw/~jehuang/oracle/ch1/1-3.htm>

第 2 節 介紹 SQL 語言

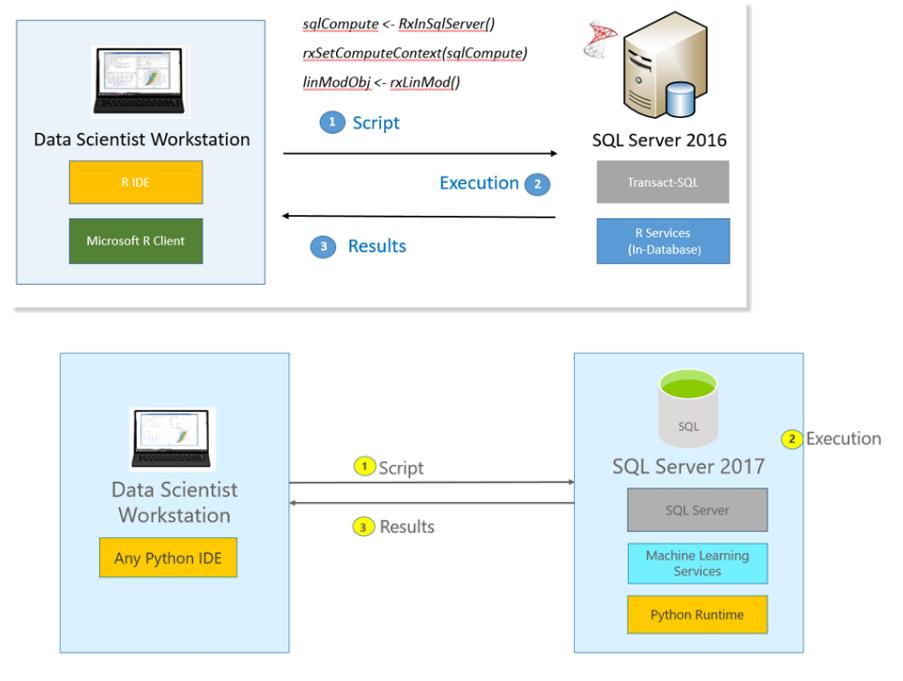
結構化查詢語言 (Structured Query Language, SQL)，誕生於 1970 年代後期，為專門用於關聯式資料庫的一種查詢語言，其可以定義資料庫結構、建立表格、指定欄位型態與長度，也能新增、異動或查詢資料。簡言之，SQL 是一種用來從資料庫讀取與儲存資料的電腦語言，讓使用者可以順利達到資料庫之間的溝通，且更有組織的管理資料。由於是共同的語法，因此不論在 MS-SQL、ORACLE、DB2、ACCESS、MySQL、T-SQL、NoSQL 等都可以通用。

本次 Workshop 所使用的軟體為美國微軟公司所推出的 SQL Server。我們將學習一般常見的幾種語法，就可以進行平常的資料查詢管理囉！

第 3 節 MS SQL server 的好處

除了資料庫軟體外，微軟的 SQL server 目前也支援許多的統計分析軟體，例如：R、SAS 等，因此未來在工作上處理龐大資料時，擁有統計分析軟體結合 SQL 的程式語言能力，將是必備的技能。

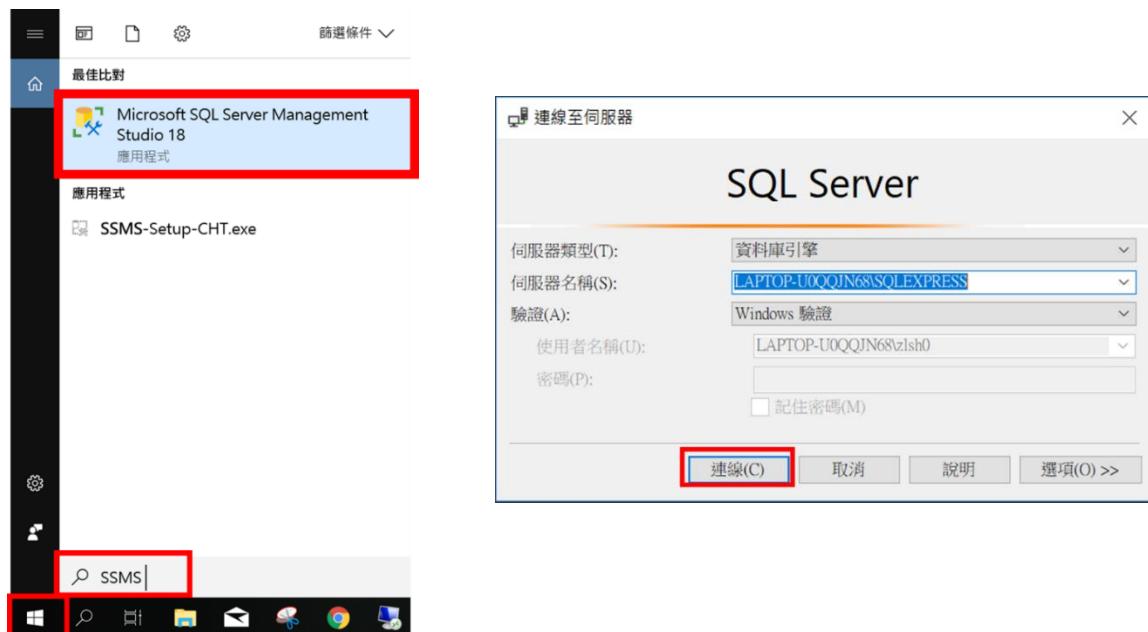
註：SQL Server 2016 開始支援 R。SQL Server 2017 則支援 R 和 Python。



第 2 章 SSMS、資料庫結構

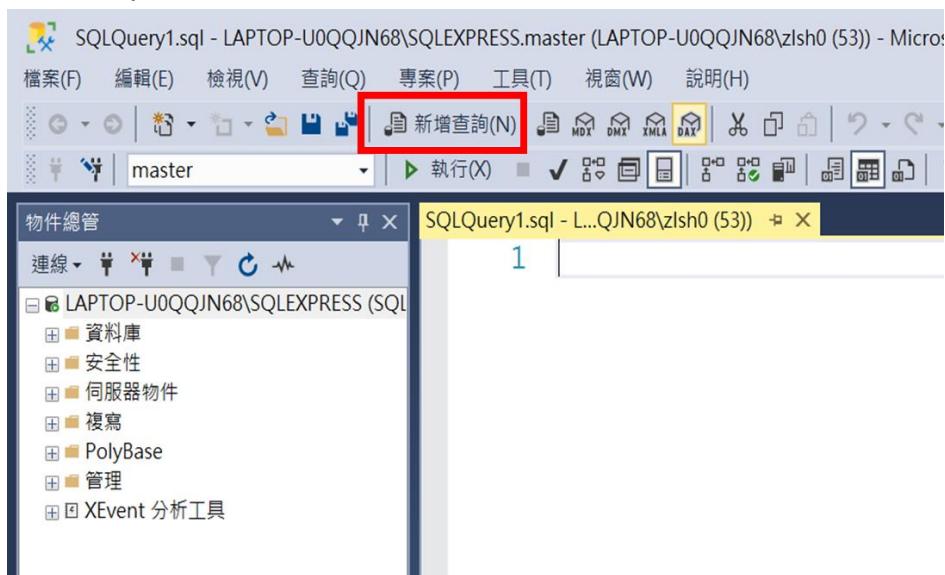
第 1 節 SSMS 介面介紹與操作

按左下角「開始」，點選 Microsoft SQL Server Management (SSMS)，軟體執行後然後按下「連線」。



接著，左邊會出現物件總管。

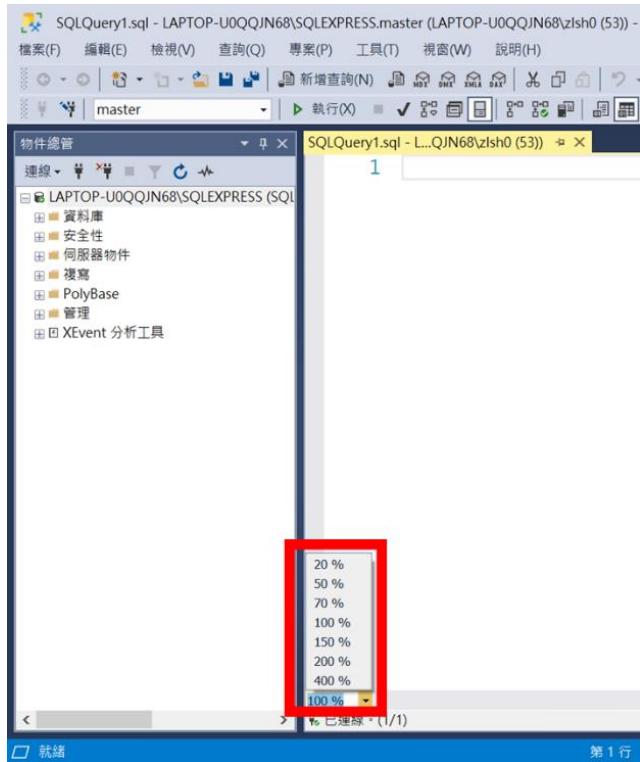
按下上方工具列的「新增查詢」，或是快速鍵 Ctrl+N，就可以開啟一個新的 SQL script。



以下，介紹一些編寫 SQL 程式時的一些小輔助。

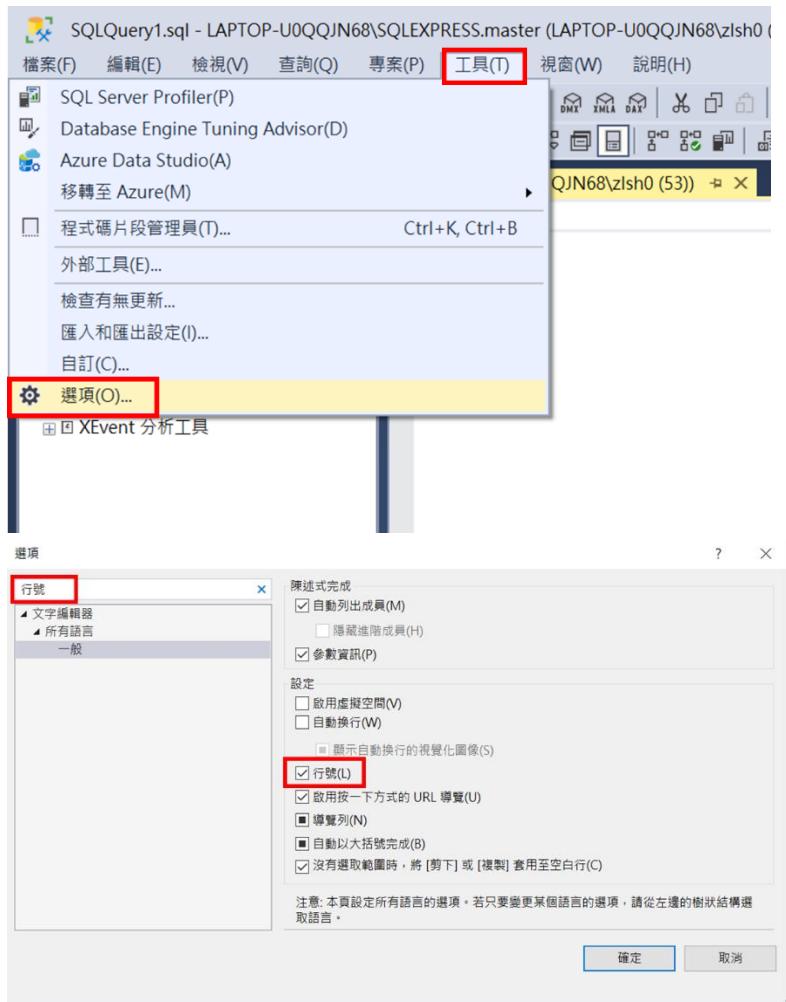
[字體大小]

按著「Ctrl」，加上滑鼠的滾輪，即可放大與縮小。也可以在 script 畫面的左下角，選擇畫面放大比例。



[行號設定]

如果是第一次使用，寫程式的 script 區域會沒有行號，這對我們編寫程式和 debug 過程十分不方便。要顯示行號，點選上方功能區的「工具」→「選項」→在搜尋選項中輸入"行號"，並將行號的勾選框框選取→按下「確定」。如此一來，撰寫 SQL script 就會有行號輔助了。



[註解]

除了行號之外，在剛開始學習程式時，註解是幫助我們快速入門的筆記功能。

在 SSMS 中註釋方式有兩種：

- (1)多行註釋： /*放入要註釋的內容*/。
- (2)一行註釋： --放入要註釋的內容

註釋的結果下圖所示。

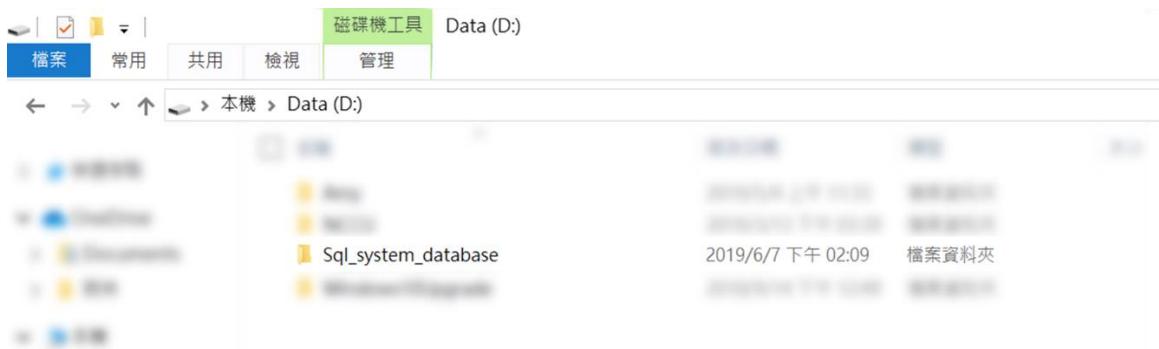
```
SQLQuery1.sql - L...QJN68\zlish0 (53)* ↗ X
1
2 --單行註解
3
4 /*
5 多行註解
6 多行註解
7 多行註解
8 */
9
```

第 2 節 建立存放資料庫的資料夾

首先，我們先在電腦內建立一個資料夾來做為存放 SQL 之 Database 的檔案路徑。

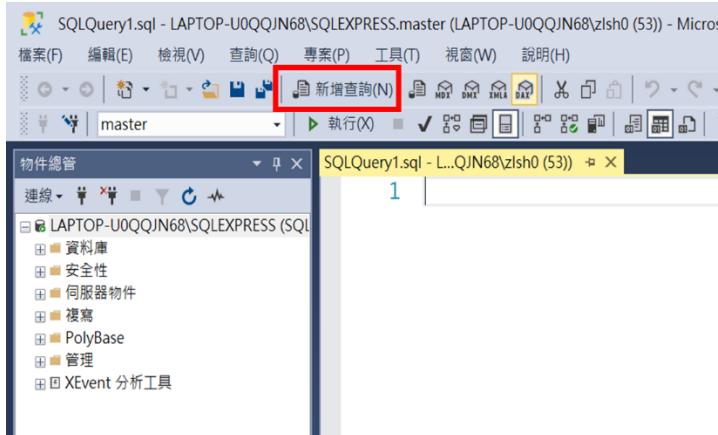
1. 先在 D 槽，新增一個資料夾"Sql_system_database"。

(NOTE：資料庫/表名稱的第一個字不可為數字。EX： 2AB、3CDE 為錯誤。)⁸



2. 開啟 SQL Server Management Studio，並點選「連線」。

3. 接著點選「新增查詢」，就會出現可輸入程式的工作區，如下圖：

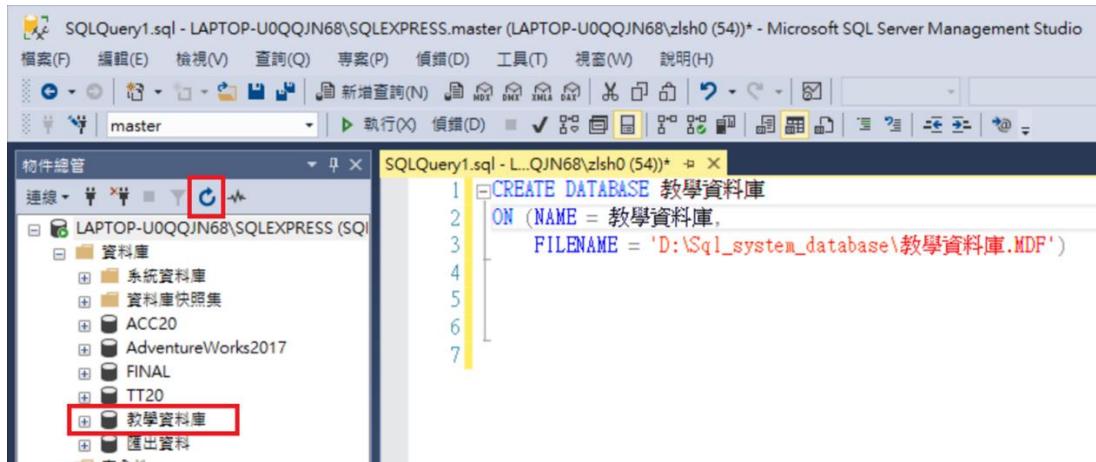


4. 在工作區，輸入下列程式碼，並按「執行」，以建立資料庫。執行完成後，在左邊物件總管上方，先按「重新整理」，再點開資料庫，就會看到剛才所建立的「教學資料庫」。

⁸ https://www.flycan.com.tw/board/modules/newbb/viewtopic.php?topic_id=78

程式碼：

```
CREATE DATABASE 教學資料庫  
ON (NAME = 教學資料庫,  
FILENAME = 'D:\Sql_system_database\教學資料庫.MDF')
```



本次工作坊結束，所建立的資料夾"Sql_system_database"，會有以下項目

- (1)存放這次的 SQL 程式語法
- (2)匯入自己的資料表(如果有時間我們再做 XD)

第 3 節 Sample 資料庫

AdventureWork2017

我們將利用 SQL Server 的範例資料庫 AdventureWork 來做學習，所以先來下載這個資料庫吧！(以下的方法是下載.bak，並用還原的方式還原資料庫)

1. 微軟已經把這個範例資料庫放到了 GitHub 上面，所以請先 Google 搜尋：
"SQL Server Samples"，點選第一個搜尋結果「GitHub - Microsoft/sql-server-samples」

Google 搜尋結果顯示了約 82,700,000 項結果。第一個結果是 [microsoft/sql-server-samples](https://github.com/microsoft/sql-server-samples)，標題為 "Official Microsoft GitHub ... - GitHub"。點選後進入 GitHub 瀏覽頁面，顯示了 "Releases"、"Samples"、"Adventure-works" 和 "Databases" 等內容。

2. 然後依照以下路徑點選：`sql-server-samples/samples/databases/adventure-works/`

GitHub 瀏覽頁面顯示了 `sql-server-samples` 這個 repository 的內容。在導航欄中，`sql-server-samples / samples / databases / adventure-works /` 被紅色方框圈出。下方列出了多個文件和資料夾，包括 `data-warehouse-install-script`、`oltp-install-script`、`README.md` 等。

3. 往下拉，到 Install from a backup 的區域，選擇 Download backup files from AdventureWorks samples databases on GitHub.

下載 48 MB AdventureWorks2017.bak

Install from a backup

Download backup files from AdventureWorks samples databases on GitHub.

You can install AdventureWorks or AdventureWorksDW by restoring a backup file. The backup files are version-specific. You can restore each backup to its respective version of SQL Server, or a later version.

For example, you can restore AdventureWorks2016 to SQL Server (starting with 2016). Regardless of whether AdventureWorks2016 is restored to SQL Server 2016, 2017, or a later version, the restored database has the database compatibility level of SQL Server 2016.

Additional OLTP backups and samples

[adventure-works-2014-oltp-in-memory-sample.zip](#)

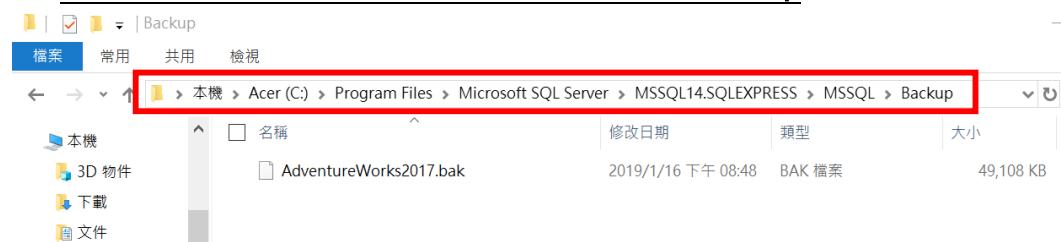
▼ Assets 20

adventure-works-2014-oltp-in-memory-sample.zip	11.9 KB
AdventureWorks-oltp-install-script.zip	20.2 MB
AdventureWorks2012.bak	44.9 MB
AdventureWorks2014.bak	44.6 MB
AdventureWorks2016.bak	46.5 MB
AdventureWorks2016_EXT.bak	125 MB
AdventureWorks2017.bak	48 MB
AdventureWorksDW-data-warehouse-install-script.zip	23.2 MB
AdventureWorksDW2012.bak	21.8 MB

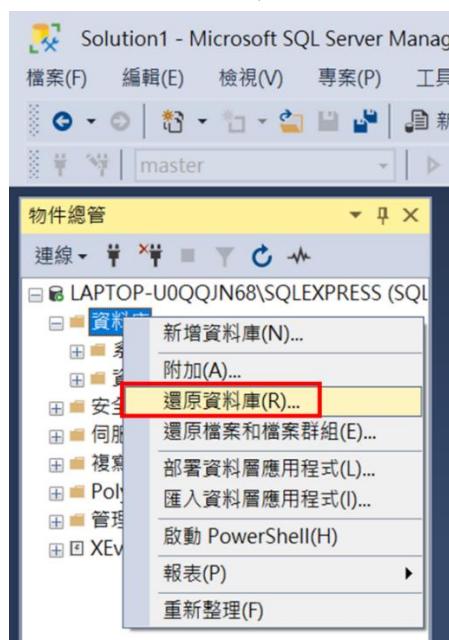
4. 解壓縮並將.bak 檔放到下面的資料夾中(每個人的 Backup 目錄不盡相同，看你把你的 SQL 安裝在哪裡)

C:\Program Files\Microsoft SQL

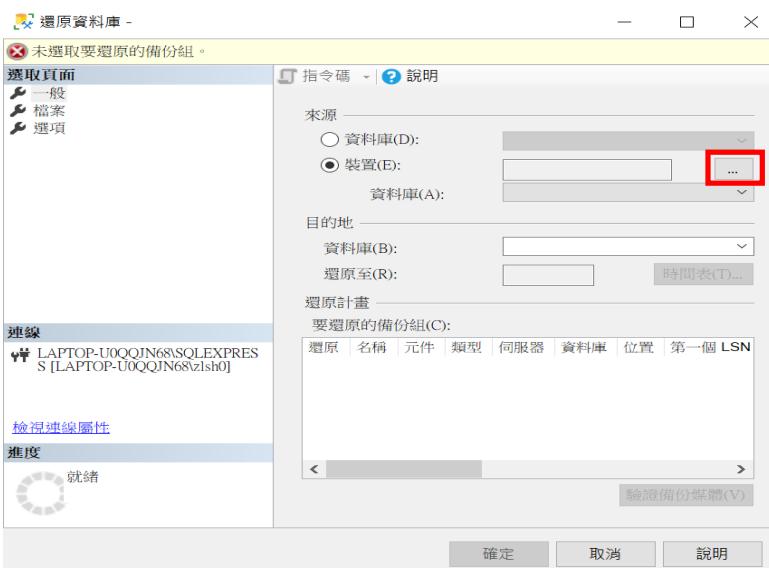
Server\MSSQL14.MSSQLSERVER\MSSQL\Backup



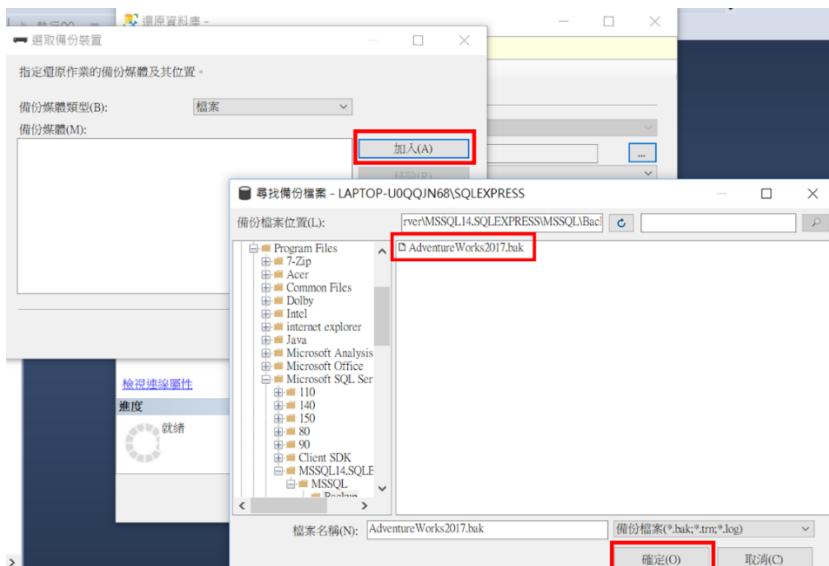
5. 在 SSMS 的左邊，物件總管的資料庫按右鍵，選擇「還原資料庫」。



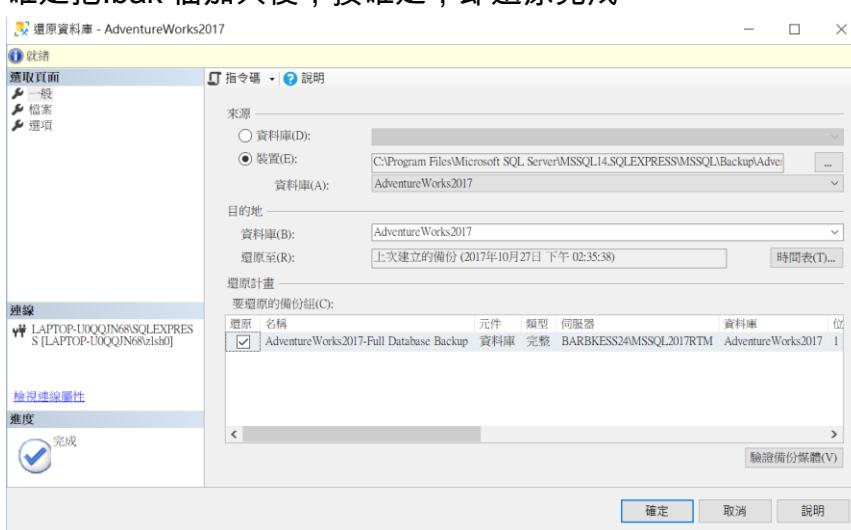
6. 點進還原資料庫後會看到下面這個介面，點選裝置，選擇右邊...的框框。



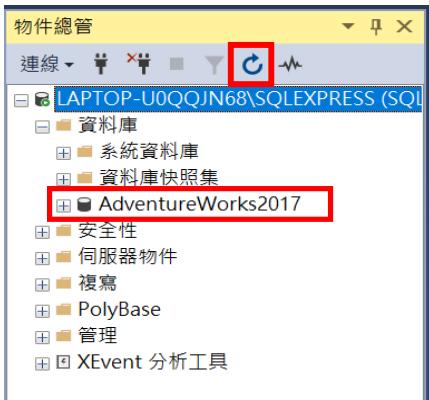
7. 點選剛剛下載的.bak 檔，然後按確定。點選加入，按確定。



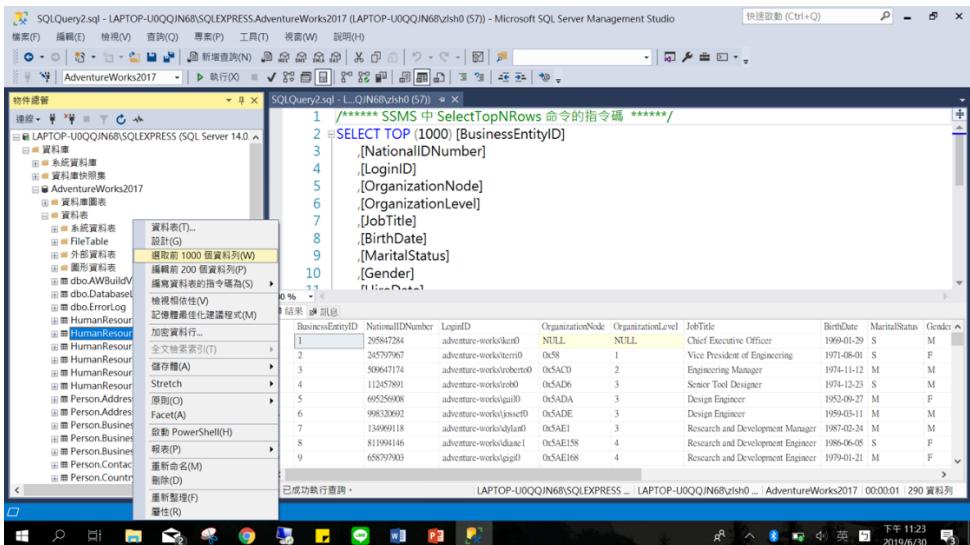
8. 確定把.bak 檔加入後，按確定，即還原完成



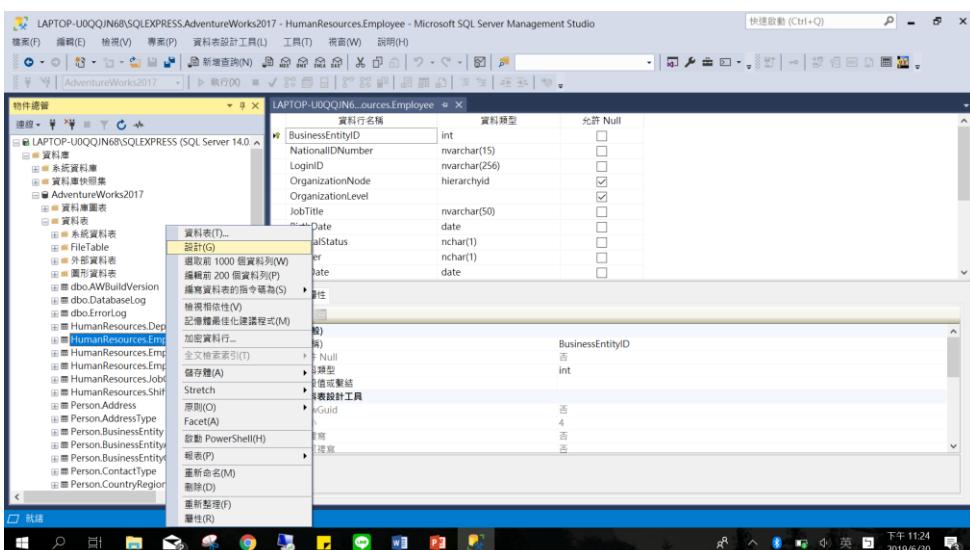
9. 按一下「重新整理」，點開「資料庫」，確認 AdventureWorks2017 是不是有還原成功！



10. 點開 AdventureWorks2017，在任意資料表按右鍵，點選「選取前 1000 個資料列(W)」，就可以看到資料輸入後的型態。



11. 在資料表按右鍵，點選「設計」，可以看到這個資料表的所擁有的欄位與鍵值。



第 4 節 範例資料庫

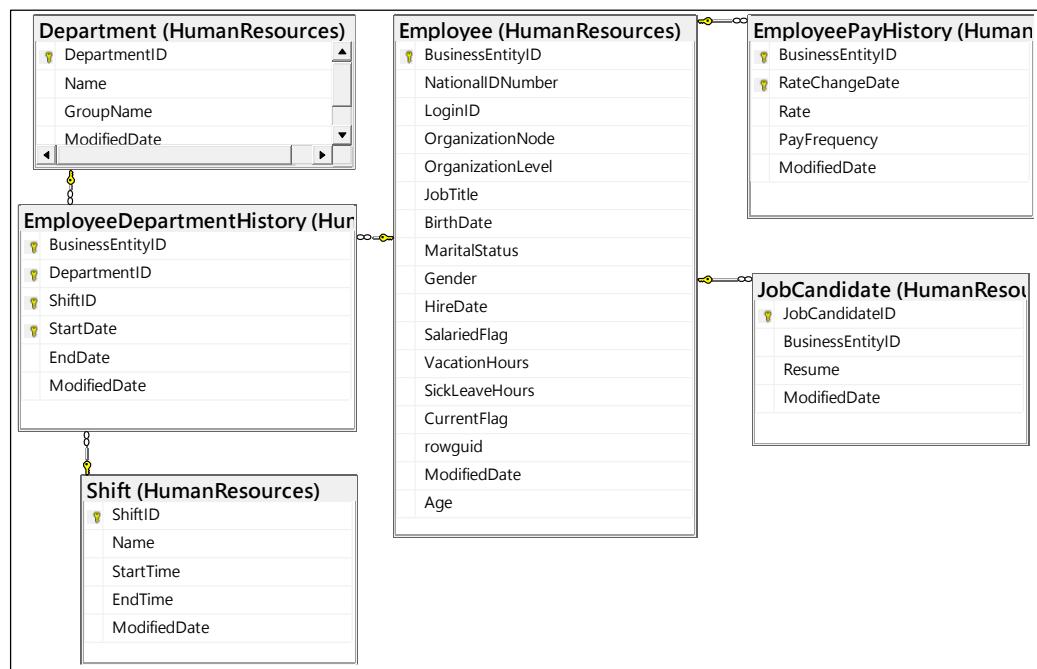
AdventureWork2017 簡介

這是一個自行車製造和銷售公司的資料庫。資料庫的結構依照 Schema(綱要)可以劃分許多模組，從而產生許多的資料表，例如模組有 HumanResources、Person、Production、Purchasing 和 Sales；而 HumanResources 模組下則有 Department、Employee 等等資料表。如果是通用的表，比如 DatabaseLog 日誌表，那麼就不屬於任何模組，使用系統預設的 Schema：dbo。而本次工作坊主要會使用下列五個資料表：

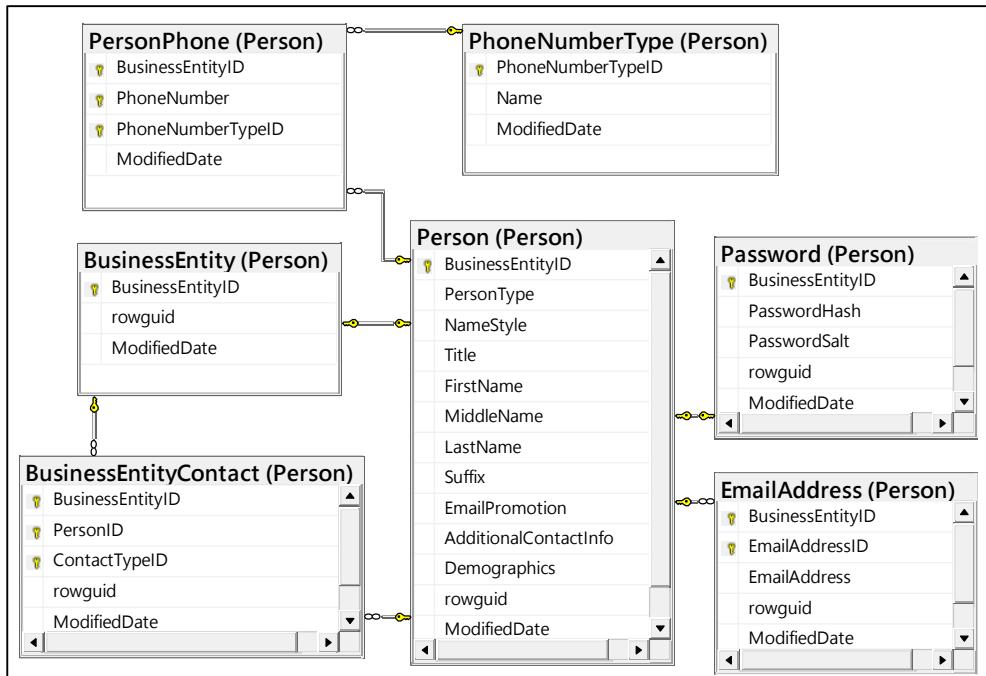
- ✓ HumanResources.Employee
- ✓ HumanResources.EmployeeDepartmentHistory
- ✓ HumanResources.Department
- ✓ Person.Person
- ✓ Sales.SalesPerson

可以看出這些資料表主要來自三個 Schema，分別是 HumanResources、Person、Sales。那我們先簡單地來看一下這三個 Schema 的資料表有哪些，了解一下 Schema 的特性。

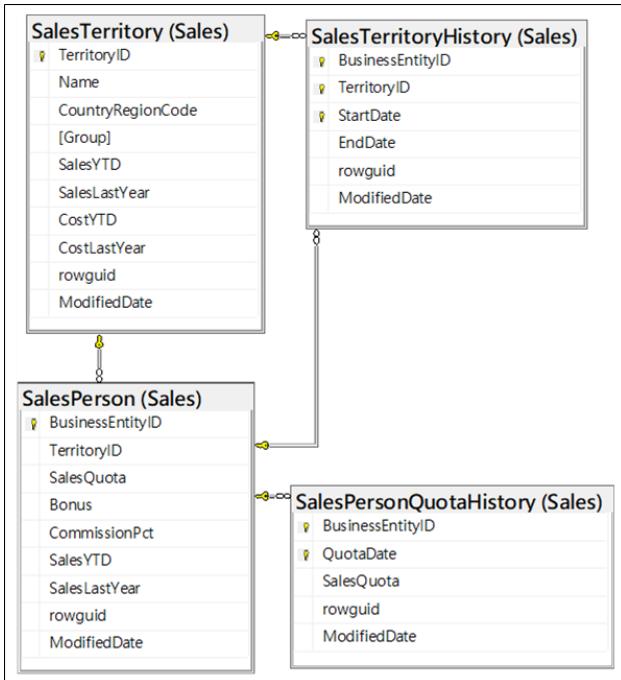
[HumanResources]



[Person]



[Sales]



總結以上，HumanResources 紀錄員工和公司有關的基本資訊，包括履歷、薪水變動、部門變動...等；Person 紀錄員工的個人資料，包括姓名、公司分配的

Email、密碼...等；Sales 紀錄員工的銷售區域、銷售配額...等。

第 5 節 主鍵(Primary Key)、外鍵

(foreign key)⁹

Well, 如果我們將所有的資料都集中在一個資料表中，不是很方便嗎？而且資料庫還是可正常運作啊！但是這樣就失去資料庫的優點了，像是節省儲存空間、減少輸入錯誤、方便資料修改...等等。那甚麼是主鍵和外鍵呢？

◎主鍵

- (1) 用來辨識記錄的欄位，具有唯一性不允許重複，因此不是每個欄位都適合作主鍵。
- (2) 資料表不一定要有主鍵，但一般來說 99.9% 都會設計成有主鍵。
- (3) 通常每個資料表只有一個欄位設定為主鍵，但有時可能沒有一個欄位具有唯一性，此時可考慮使用兩個或多個欄位組合起來做為主鍵。

我們用 HumanResources 下的 Employee 和 EmployeeDepartmentHistory 資料表對照解釋

DESKTOP-LM...s.Employee		
資料行名稱	資料類型	允許 Null
BusinessEntityID	int	<input type="checkbox"/>
NationalIDNumber	nvarchar(15)	<input type="checkbox"/>
LoginID	nvarchar(256)	<input type="checkbox"/>
OrganizationNode	hierarchyid	<input checked="" type="checkbox"/>
OrganizationLevel		<input checked="" type="checkbox"/>
JobTitle	nvarchar(50)	<input type="checkbox"/>
BirthDate	date	<input type="checkbox"/>
MaritalStatus	nchar(1)	<input type="checkbox"/>
Gender	nchar(1)	<input type="checkbox"/>
HireDate	date	<input type="checkbox"/>
SalariedFlag	Flag:bit	<input type="checkbox"/>
VacationHours	smallint	<input type="checkbox"/>
SickLeaveHours	smallint	<input type="checkbox"/>
CurrentFlag	Flag:bit	<input type="checkbox"/>
rowguid	uniqueidentifier	<input type="checkbox"/>
ModifiedDate	datetime	<input type="checkbox"/>
Age	int	<input checked="" type="checkbox"/>

DESKTOP-LM...entHistory		
資料行名稱	資料類型	允許 Null
BusinessEntityID	int	<input type="checkbox"/>
DepartmentID	smallint	<input type="checkbox"/>
ShiftID	tinyint	<input type="checkbox"/>
StartDate	date	<input type="checkbox"/>
EndDate	date	<input checked="" type="checkbox"/>
ModifiedDate	datetime	<input type="checkbox"/>

↑ EmployeeDepartmentHistory

← Employee

Employee 資料表主鍵為

BusinessEntityID；而

EmployeeDepartmentHistory 資料表主鍵則是四個欄位構成的，分別是 BusinessEntityID、DepartmentID、ShiftID、StartDate。主鍵的存在就是要讓每一列的紀錄唯一。

⁹ 外鍵(foreign key)也有人稱為外來鍵、連外鍵、外部鍵。

◎外鍵：

(1)外鍵是用來連結其他資料表用的，資料表之間的關係是藉由外鍵所建立的。我們一樣用 HumanResources 下的 Employee 和 EmployeeDepartmentHistory 資料表對照解釋

DESKTOP-LM...s.Employee		
資料行名稱	資料類型	允許 Null
BusinessEntityID	int	<input type="checkbox"/>
NationalIDNumber	nvarchar(15)	<input type="checkbox"/>
LoginID	nvarchar(256)	<input type="checkbox"/>
OrganizationNode	hierarchyid	<input checked="" type="checkbox"/>
OrganizationLevel	hierarchyid	<input checked="" type="checkbox"/>
JobTitle	nvarchar(50)	<input type="checkbox"/>
BirthDate	date	<input type="checkbox"/>
MaritalStatus	nchar(1)	<input type="checkbox"/>
Gender	nchar(1)	<input type="checkbox"/>
HireDate	date	<input type="checkbox"/>
SalariedFlag	Flag:bit	<input type="checkbox"/>
VacationHours	smallint	<input type="checkbox"/>
SickLeaveHours	smallint	<input type="checkbox"/>
CurrentFlag	Flag:bit	<input type="checkbox"/>
rowguid	uniqueidentifier	<input type="checkbox"/>
ModifiedDate	datetime	<input type="checkbox"/>

DESKTOP-LM...entHistory		
資料行名稱	資料類型	允許 Null
BusinessEntityID	int	<input type="checkbox"/>
DepartmentID	smallint	<input type="checkbox"/>
ShiftID	tinyint	<input type="checkbox"/>
StartDate	date	<input type="checkbox"/>
EndDate	date	<input checked="" type="checkbox"/>
ModifiedDate	datetime	<input type="checkbox"/>

↑EmployeeDepartmentHistory

←Employee

如果上面兩個資料表要做關聯，必須依靠 BusinessEntityID 這個欄位，所以如果我想知道該 BusinessEntityID 的這個員工在哪個部門以及員工的性別，就必須用 BusinessEntityID 做關聯去查詢 Gender。那麼此時 BusinessEntityID 這個欄位就稱為外鍵。

[動動腦時間 Q1]

如果我想知道這個 BusinessEntityID，這個員工曾經待過的部門 (DepartmentID)，以及他的名字(FirstName)，那麼需要用到哪兩個表，這兩個表之間又是用什麼欄位產生關連的呢？

[答案 Ans1]

在 HumanResources.EmployeeDepartmentHistory 可以知道該員工曾經待過的部門 DepartmentID，而 Person.Person 可以知道員工的名字。兩個資料表可以透過就 BusinessEntityID 欄位做合併。

(註解：如果需要知道部門 DepartmentID 的名字，可以用 HumanResources.Department 資料表做關聯。)

The diagram illustrates the relationship between three tables:

- HumanResources.EmployeeDepartmentHistory** (Left):

資料行名稱	資料類型
BusinessEntityID	int
DepartmentID	smallint
ShiftID	tinyint
StartDate	date
EndDate	date
ModifiedDate	datetime
- Person.Person** (Right):

資料行名稱	資料類型
BusinessEntityID	int
PersonType	nchar(2)
NameStyle	NameStyle:bit
Title	nvarchar(8)
FirstName	Name:nvarchar(40)
MiddleName	Name:nvarchar(40)
LastName	Name:nvarchar(40)
Suffix	nvarchar(10)
EmailPromotion	int
AdditionalContactInfo	xml(CONTENT P...
Demographics	xml(CONTENT P...
rowguid	uniqueidentifier
ModifiedDate	datetime
- HumanResources.Department** (Bottom):

資料行名稱	資料類型
DepartmentID	smallint
Name	Name:nvarchar(40)
GroupName	Name:nvarchar(40)
ModifiedDate	datetime

Relationships are indicated by arrows:

- An arrow points from the **BusinessEntityID** column in the **EmployeeDepartmentHistory** table to the **BusinessEntityID** column in the **Person** table.
- An arrow points from the **DepartmentID** column in the **EmployeeDepartmentHistory** table to the **DepartmentID** column in the **Department** table.
- An arrow points from the **DepartmentID** column in the **Department** table to the **BusinessEntityID** column in the **Person** table.

第 3 章 SQL 基本指令

第 1 節 基本資料格式

SQL 的資料欄位，主要分為數值欄位、字串欄位兩大類。

■ 數值欄位：

int：整數，範圍從 $-2^{31} \sim 2^{31}$

bigint：整數，範圍從 $-2^{63} \sim 2^{63} - 1$

float：近似小數資料的資料類型，範圍從 $-1.79 \times 10^{308} \sim 1.79 \times 10^{308}$

real：近似小數資料的資料類型，範圍從 $-3.04 \times 10^{38} \sim 3.04 \times 10^{38}$

■ 字串欄位：char、varchar、nchar、nvarchar

char：固定大小浪費空間，所需計算時間少，只能儲存英文字元

varchar：不固定長度，必須要花費較多的CPU計算時間，只能儲存英文字元

nchar、nvarchar：與前兩者的差異，可儲存其他字元(中文)

NOTE：數值欄位常用float。字串欄位則是nvarchar，若欄位名稱是由中文與數字組成，則在指令中需加上[](ex.[資料行0])。

第2節 基本查詢指令

2.1 SQL 運算子 Operators

SQL有三種類型的運算子，分別是算術運算子、邏輯運算子以及比較運算子。

算數運算子			
運算子	用途	用法	結果
+	加法	1+2	3
-	減法	3-5	-2
*	乘法	6*6	36
/	商數	8/5	1.6
%	餘數	8%	3

邏輯運算子	
運算子	語意
AND	且
OR	或
NOT	非

比較運算子			
運算子	語意	用法	結果
<	小於	3 < 5	TRUE
>	大於	8 > 1	TRUE
<=	小於等於	20 <= 10	TRUE
>=	大於等於	4 >= 6	TRUE
=	等於	2 = 2	TRUE
≠	不等於	7 ≠ 6	TRUE
BETWEEN	在某個範圍內		
LIKE	符合模式條件		
IN	值存在於IN集合中		

◎其他符號：

* : ALL

': 中間輸入特定值，主要是用於加入字串

-- : 後面可輸入註解，不會被程式語言讀入

/* */ : 中間可輸入註解，不會被程式語言讀入

2.2 SQL 基本語法

SELECT 欄位名稱

(INTO 表名)

FROM 表名

WHERE 條件敘述

GROUP BY 欄位名稱

ORDER BY 欄位名稱

◎EX1：

使用 HumanResources.Department 資料表，挑出 DepartmentID, Name 這兩個欄位的資料。(註：這兩個欄位分別代表部門編號與部門名稱)

<ANS1>：

```
SELECT DepartmentID, Name          /* 挑出兩個欄位的資料 */
      FROM HumanResources. Department    /* 從資料表抓資料 */
```

	DepartmentID	Name
1	12	Document Control
2	1	Engineering
3	16	Executive
4	14	Facilities and Maintenance
5	10	Finance
6	9	Human Resources

◎EX2：

使用 HumanResources.Department 表，挑出 DepartmentID, Name 這兩個欄位的資料，並依照 DepartmentID 排序。

<ANS2>：

```
SELECT DepartmentID, Name          /* 挑出兩個欄位的資料 */
      FROM HumanResources. Department    /* 從資料表抓資料 */
      ORDER BY DepartmentID            /* 預設是由小到大，可以加上 DESC 改為
                                         由大到小*/
```

	DepartmentID	Name
1	1	Engineering
2	2	Tool Design
3	3	Sales
4	4	Marketing
5	5	Purchasing
6	6	Research and Development

◎EX3:

將 HumanResources.Employee 表依照性別和生日排序，且性別女生 F 排前面，生日由大到小排序。

<ANS3> :

<pre>SELECT * FROM HumanResources.Employee ORDER BY Gender, 7 DESC</pre>	<pre>SELECT Gender, BirthDate FROM HumanResources.Employee ORDER BY 1, 2 DESC</pre>
--	---

➤ 上面這兩種寫法有什麼差別呢？

- (1) ORDER BY後面的數字，是SELECT出的欄位順序。
- (2) DESC的效果只會跟前面的參數有關，如果要ORDER BY兩個條件都DESC，則兩個參數後面都要加DESC。
- (3) ORDER BY 是預設由小到大排序；加上ASC/DESC，分別是遞增/遞減。

	Gender	BirthDate
1	F	1991-05-31
2	F	1990-11-01
3	F	1990-01-25
4	F	1989-09-28
5	F	1989-08-09
6	F	1989-05-28

◎EX4:

想知道 HumanResources.Employee 裡的員工有哪些職稱。

<ANS4> : 比較下列兩種寫法

<pre>SELECT JobTitle FROM HumanResources.Employee</pre>	<pre>SELECT DISTINCT (JobTitle) FROM HumanResources.Employee</pre>
---	--

	JobTitle
1	Chief Executive Officer
2	Vice President of Engineering
3	Engineering Manager
4	Senior Tool Designer
5	Design Engineer
6	Design Engineer
7	Research and Development Manager
8	Research and Development Engineer
9	Research and Development Engineer



	JobTitle
1	Accountant
2	Accounts Manager
3	Accounts Payable Specialist
4	Accounts Receivable Specialist
5	Application Specialist
6	Assistant to the Chief Financial Officer
7	Benefits Specialist
8	Buyer
9	Chief Executive Officer

◎EX5:

查詢 HumanResources.Employee 裡的員工，性別為女性，且職稱為 Design Engineer 的員工，查詢結果要有 BusinessEntityID 和雇用日期 HireDate。

<ANS5> :

```
SELECT BusinessEntityID, JobTitle, Gender, HireDate  
FROM HumanResources.Employee  
WHERE JobTitle='Design Engineer' AND Gender='F'
```

	BusinessEntityID	JobTitle	Gender	HireDate
1	5	Design Engineer	F	2008-01-06
2	15	Design Engineer	F	2011-01-18

➤ AND 改成 OR 會變成怎麼樣？查詢結果的條件就會變成是 Design Engineer 或女生。

	BusinessEntityID	JobTitle	Gender	HireDate
1	2	Vice President of Engineering	F	2008-01-31
2	5	Design Engineer	F	2008-01-06
3	6	Design Engineer	M	2008-01-24
4	8	Research and Development Engineer	F	2008-12-29
5	9	Research and Development Engineer	F	2009-01-16
6	13	Tool Designer	F	2010-12-23

[動動腦時間 Q1]

目前我們已經學到了如何從一個資料表，利用 SELECT 抓出特定的欄位，並且可以用 WHERE 設定查詢條件，也可以用 ORDER BY 將結果排序。

請利用 HumanResources.Employee 資料表，找出女生的員工，結果要顯示 BusinessEntityID, Gender, HireDate, Jobtitle，並且依照 HireDate 由小到大排序。

[答案 Ans1]

```
SELECT BusinessEntityID, Gender, HireDate, JobTitle  
FROM HumanResources.Employee  
WHERE Gender='F'  
ORDER BY HireDate
```

	BusinessEntityID	Gender	HireDate	JobTitle
1	40	F	2007-12-26	Production Supervisor - WC60
2	5	F	2008-01-06	Design Engineer
3	48	F	2008-01-06	Production Technician - WC10
4	2	F	2008-01-31	Vice President of Engineering
5	27	F	2008-02-27	Production Supervisor - WC60
6	52	F	2008-03-10	Production Technician - WC10

◎EX6:

已知行銷部門的職稱開頭都為 Marketing 開頭，利用 HumanResources.Employee 資料表中 JobTitle 欄位的查詢行銷部門的員工有誰。

<ANS6> :

```
SELECT *
FROM HumanResources.Employee
WHERE JobTitle LIKE 'Marketing%'
```

(註解：關於%的應用，可以 Google"SQL 模糊查詢"。%表示任意 0 個或多個字元。可匹配任意類型和長度的字元)

	BusinessEntityID	NationalIDNumber	LoginID	OrganizationNode	OrganizationLevel	JobTitle	BirthDate
1	16	24756624	adventure-works\david0	0x68	1	Marketing Manager	1975-03-19
2	17	253022876	adventure-works\kevin0	0x6AC0	2	Marketing Assistant	1987-05-03
3	18	222969461	adventure-works\john5	0x6B40	2	Marketing Specialist	1978-03-06
4	19	52541318	adventure-works\mary2	0x6BC0	2	Marketing Assistant	1978-01-29
5	20	323403273	adventure-works\wanida0	0x6C20	2	Marketing Assistant	1975-03-17
6	21	243322160	adventure-works\terry0	0x6C60	2	Marketing Specialist	1986-02-04

◎EX7:

利用 HumanResources.Employee 查詢非行銷部門的員工有誰。

<ANS7> :

```
SELECT *
FROM HumanResources.Employee
WHERE JobTitle NOT LIKE 'Marketing%'
```

	BusinessEntityID	NationalIDNumber	LoginID	OrganizationNode	OrganizationLevel	JobTitle	BirthDate
1	1	295847284	adventure-works\ken0	NULL	NULL	Chief Executive Officer	1969-01-29
2	2	245797967	adventure-works\terri0	0x58	1	Vice President of Engineering	1971-08-01
3	3	509647174	adventure-works\roberto0	0x5AC0	2	Engineering Manager	1974-11-12
4	4	112457891	adventure-works\rob0	0x5AD6	3	Senior Tool Designer	1974-12-23
5	5	695256908	adventure-works\gail0	0x5ADA	3	Design Engineer	1952-09-27
6	6	998320692	adventure-works\jossef0	0x5ADE	3	Design Engineer	1959-03-11

◎EX8:

利用 HumanResources.Employee 查詢員工 VacationHour 在 40-45 間的員工。

<ANS8> :

```
SELECT BusinessEntityID, VacationHours  
FROM HumanResources.Employee  
WHERE VacationHours BETWEEN 40 AND 45
```

試試看用 WHERE ... AND ... 寫出跟上面一樣的結果。

```
SELECT BusinessEntityID, VacationHours  
FROM HumanResources.Employee  
WHERE VacationHours >= 40 AND VacationHours <= 45
```

	BusinessEntityID	VacationHours
1	16	40
2	17	42
3	19	43
4	20	41
5	21	44
6	22	45

◎EX9:

利用 HumanResources.Employee 查詢員工 SickLeaveHour 為 28、36、64、80 的員工。

<ANS9> :

```
SELECT BusinessEntityID, SickLeaveHours  
FROM HumanResources.Employee  
WHERE SickLeaveHours IN (28, 36, 64, 81)
```

	BusinessEntityID	SickLeaveHours
1	10	64
2	33	28
3	37	28
4	49	64
5	54	64
6	67	36

[動動腦時間 Q2]

在上一頁，我們學會了如何利用比較運算子，讓 WHERE 的查詢條件更靈活。

請利用 Person.Person 資料表，找出 FirstName 為 Jason, Nicole, Steve，且 BusinessEntityID 小於 1000 的員工，最後查詢結果要顯示 BusinessEntityID, FirstName, MiddleName, LastName。

[答案 Ans2]

```
SELECT BusinessEntityID, FirstName, MiddleName, LastName  
FROM Person.Person  
WHERE FirstName in ('Jason', 'Nicole', 'Steve') AND  
BusinessEntityID <=1000
```

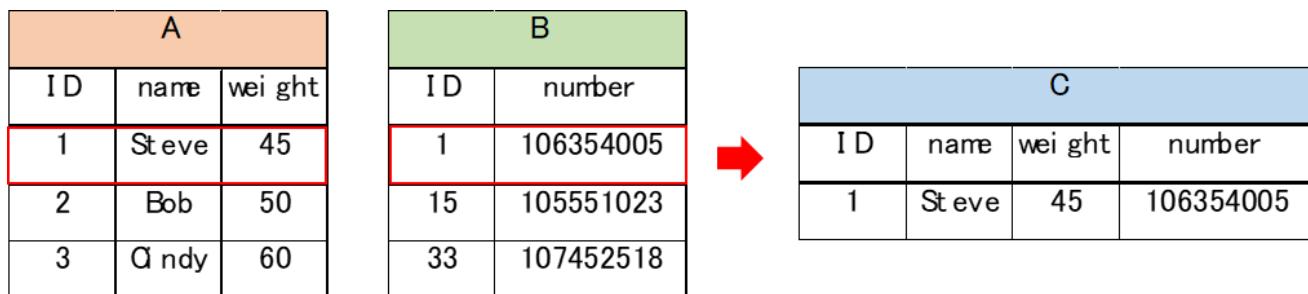
	BusinessEntityID	FirstName	MiddleName	LastName
1	69	Steve	F	Masters
2	132	Nicole	B	Holliday
3	148	Jason	M	Watters
4	633	Steve	J.	Carnes

第 3 節 進階查詢指令

3.1 資料表合併

➤ 平行合併

```
SELECT *          /* 挑出所有欄位的資料 */  
FROM A, B        /* 從A, B表抓資料 */  
WHERE A.ID=B.ID  /* 將A, B表中相同ID的合併 */
```



◎EX1:

利用 HumanResources.EmployeeDepartmentHistory 和
HumanResources.Department 這兩張表中的 DepartmentID , 查詢員工隸屬在
哪個部門 Name , 和部門所屬的團隊名稱 GroupName。

<ANS1> :

```
SELECT HumanResources.EmployeeDepartmentHistory.BusinessEntityID,  
       HumanResources.Department.DepartmentID, HumanResources.Department.  
       Name, HumanResources.Department.GroupName  
  
FROM HumanResources.Department,  
      HumanResources.EmployeeDepartmentHistory  
  
WHERE  
  
HumanResources.Department.DepartmentID=HumanResources.EmployeeDep  
artmentHistory.DepartmentID
```

➤ 天啊這語法也太長！不能簡短一點嗎？

```
SELECT edp.BusinessEntityID, dp.DepartmentID, dp.GroupName,  
       dp.Name  
  
FROM HumanResources.Department dp,  
      HumanResources.EmployeeDepartmentHistory edp  
  
WHERE dp.DepartmentID=edp.DepartmentID
```

➤ 試試看用 JOIN... ON... , 也會有一樣結果

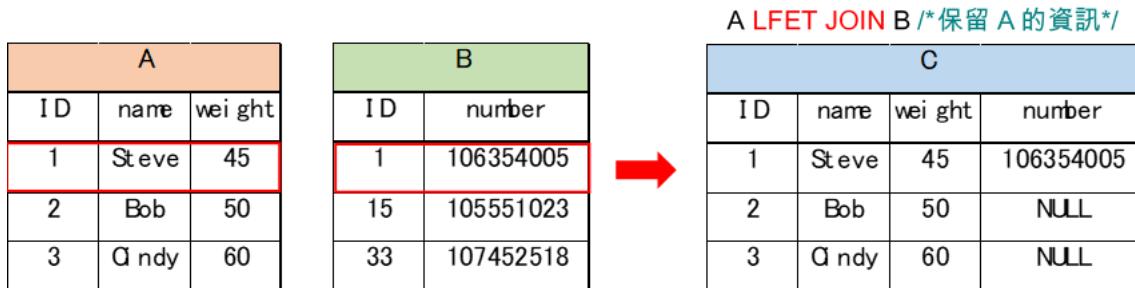
```
SELECT *          /* 挑出所有欄位的資料 */  
FROM A JOIN B    /* 從A,B資料表抓資料 */  
ON A.ID=B.ID     /* 將A,B表中相同ID的合併 */
```

```
SELECT edp.BusinessEntityID, dp.DepartmentID, dp.GroupName,  
       dp.Name  
  
FROM HumanResources.Department dp JOIN  
      HumanResources.EmployeeDepartmentHistory edp  
ON dp.DepartmentID=edp.DepartmentID
```

	BusinessEntityID	DepartmentID	GroupName	Name
1	2	1	Research and Development	Engineering
2	3	1	Research and Development	Engineering
3	4	1	Research and Development	Engineering
4	5	1	Research and Development	Engineering
5	6	1	Research and Development	Engineering
6	14	1	Research and Development	Engineering
7	15	1	Research and Development	Engineering
8	4	2	Research and Development	Tool Design
9	11	2	Research and Development	Tool Design

◎EX2:

Person.Person 裡面有公司所有員工資料，HumanResources.Employee 只有部分員工，請合併兩張表並保留 Person.Person 的資訊。



<ANS2> :

```

SELECT p.BusinessEntityID, p.FirstName, p.LastName, e.Gender,
e.BirthDate, e.HireDate, e.JobTitle
FROM Person.Person p LEFT JOIN HumanResources.Employee e
ON p.BusinessEntityID=e.BusinessEntityID
    
```

	BusinessEntityID	FirstName	LastName	Gender	BirthDate	HireDate	JobTitle
1	285	Syed	Abbas	M	1975-01-11	2013-03-14	Pacific Sales Manager
2	293	Catherine	Abel	NULL	NULL	NULL	NULL
3	295	Kim	Abercrombie	NULL	NULL	NULL	NULL
4	2170	Kim	Abercrombie	NULL	NULL	NULL	NULL
5	38	Kim	Abercrombie	F	1966-12-14	2010-01-16	Production Technician - WC60
6	211	Hazem	Abolrous	M	1977-10-26	2009-02-28	Quality Assurance Manager

- 注意 LEFT JOIN 的語法，合併後會以哪邊的資訊為主要的資料表作保留
- (1) 如果是大 LEFT JOIN 小，會有 NULL(空值)的產生。
- (2) 如果是小 LEFT JOIN 大，則不會有上述情況。
- (3) RIGHT JOIN 就是情況反過來。

➤ 垂直合併

```
SELECT *      /* 挑出所有欄位的資料 */
INTO B        /* 存到新的資料表B */
FROM A1       /* 先從A1表抓資料 */
UNION ALL
SELECT *
FROM A2       /* 再從A2表抓資料 */
```

The diagram illustrates the vertical merge (UNION ALL) of two tables, A1 and A2, into a new table B. On the left, there are two tables: A1 (orange header) and A2 (green header). Both tables have columns: ID, name, weight, and number. Table A1 contains 3 rows: (1, Steve, 45, 106354005), (2, Bob, 50, 105551023), and (3, Candy, 60, 107452518). Table A2 contains 3 rows: (25, Jay, 45, 106354005), (65, Madison, 50, 105551023), and (44, Hamilton, 60, 107452518). An arrow points from both tables to the right, leading to a third table, B (blue header), which contains all 6 rows from both A1 and A2. The columns in table B are identical to those in A1 and A2.

A1			
ID	name	weight	number
1	Steve	45	106354005
2	Bob	50	105551023
3	Candy	60	107452518

A2			
ID	name	weight	number
25	Jay	45	106354005
65	Madison	50	105551023
44	Hamilton	60	107452518

B			
ID	name	weight	number
1	Steve	45	106354005
2	Bob	50	105551023
3	Candy	60	107452518
25	Jay	45	106354005
65	Madison	50	105551023
44	Hamilton	60	107452518

◎EX3：

查詢"Marketing"開頭的職稱與"Design"開頭的職稱，並把列成一張表。

<ANS3>：

```
SELECT *
FROM HumanResources.Employee
WHERE JobTitle LIKE 'Marketing%'
UNION ALL
SELECT *
FROM HumanResources.Employee
WHERE JobTitle LIKE 'Design%'
```

	BusinessEntityID	NationalIDNumber	LoginID	OrganizationNode	OrganizationLevel	JobTitle	BirthDate
1	16	24756624	adventure-works\david0	0x68	1	Marketing Manager	1975-03-19
2	17	253022876	adventure-works\kevin0	0x6AC0	2	Marketing Assistant	1987-05-03
3	18	222969461	adventure-works\john5	0x6B40	2	Marketing Specialist	1978-03-06
4	19	52541318	adventure-works\mary2	0x6BC0	2	Marketing Assistant	1978-01-29
5	20	323403273	adventure-works\wanida0	0x6C20	2	Marketing Assistant	1975-03-17
6	21	243322160	adventure-works\terry0	0x6C60	2	Marketing Specialist	1986-02-04
7	22	95958330	adventure-works\sariya0	0x6CA0	2	Marketing Specialist	1987-05-21
8	23	767955365	adventure-works\mary0	0x6CE0	2	Marketing Specialist	1962-09-13
9	24	72636981	adventure-works\jill0	0x6D10	2	Marketing Specialist	1979-06-18
10	5	695256908	adventure-works\gail0	0x5ADA	3	Design Engineer	1952-09-27
11	6	998320692	adventure-works\jossef0	0x5ADE	3	Design Engineer	1959-03-11
12	15	56920285	adventure-works\sharon0	0x5AE7	3	Design Engineer	1961-05-02

[動動腦時間 Q3]

這一小節，我們學會了合併兩個資料表查詢的方法，也學會了將程式碼縮短的寫法。

已知 Person.Person 資料表有全部員工的資料，Sale.SalesPerson 資料表只有收錄行銷人員，請幫我找出 Sales.SalesPerson 中，所有員工的 FirstName, MiddleName, LastName，最後查詢結果要顯示 BusinessEntityID, FirstName, MiddleName, LastName, SalesYTD(今年目前的銷售總計), SalesLastYear(前一年的銷售總計)。

[答案 Ans3]

```
SELECT sp.BusinessEntityID, p.FirstName, p.MiddleName,
p.LastName, sp.SalesYTD, sp.SalesLastYear
FROM Sales.SalesPerson sp LEFT JOIN Person.Person p
ON sp.BusinessEntityID=p.BusinessEntityID
```

(註解：也可以不用 JOIN，單純使用 WHERE 條件設定)

	BusinessEntityID	FirstName	MiddleName	LastName	SalesYTD	SalesLastYear
1	274	Stephen	Y	Jiang	559697.5639	0.00
2	275	Michael	G	Blythe	3763178.1787	1750406.4785
3	276	Linda	C	Mitchell	4251368.5497	1439156.0291
4	277	Jillian	NULL	Carson	3189418.3662	1997186.2037
5	278	Garrett	R	Vargas	1453719.4653	1620276.8966
6	279	Tsvi	Michael	Reiter	2315185.611	1849640.9418

3.2 Grouping

將指定的欄位進行分組計算，可以使用的函數有 SUM(...), AVG(...), MIN(...), MAX(...), COUNT(...)等。

```
SELECT b, COUNT(ID)      /* 挑出b與計算的資料 */
      FROM A              /* b與ID都是從A資料表中的欄位 */
     GROUP BY b            /* 根據欄位b進行分組運算 */
```

EX1 :

利用 HumanResources.EmployeeDepartmentHistory 和 HumanResources.Department 這兩張表中的 DepartmentID，計算員工隸屬同部門與相同團隊名稱(意即相同的 Name 和 GroupName)的人有幾個。

<ANS1> :

```
SELECT dp.GroupName, dp.Name, COUNT(edp.BusinessEntityID)
      FROM HumanResources.Department dp JOIN
           HumanResources.EmployeeDepartmentHistory edp
     ON dp.DepartmentID=edp.DepartmentID
    GROUP BY dp.GroupName, dp.Name
```

	GroupName	Name	(沒有資料行名稱)
1	Research and Development	Engineering	7
2	Research and Development	Tool Design	4
3	Sales and Marketing	Sales	18
4	Sales and Marketing	Marketing	10
5	Inventory Management	Purchasing	13
6	Research and Development	Research and Development	4

➤ 紿這個欄位一個名字吧！

```
SELECT dp.GroupName, dp.Name, COUNT(edp.BusinessEntityID) AS
      TotalNumber
      FROM HumanResources.Department dp JOIN
           HumanResources.EmployeeDepartmentHistory edp
     ON dp.DepartmentID=edp.DepartmentID
    GROUP BY dp.GroupName, dp.Name
```

	GroupName	Name	TotalNumber
1	Research and Development	Engineering	7
2	Research and Development	Tool Design	4
3	Sales and Marketing	Sales	18
4	Sales and Marketing	Marketing	10
5	Inventory Management	Purchasing	13
6	Research and Development	Research and Development	4

EX2 :

請使用 Sales.SalesPerson 和 HumanResources.Employee 計算這17位員工的男女分別的平均年度銷售量。

<ANS2> :

```
SELECT e.Gender, AVG(sp.SalesQuota) AS AvgSale
FROM Sales.SalesPerson sp,
HumanResources.Employee e
WHERE e.BusinessEntityID=sp.BusinessEntityID
GROUP BY e.Gender
```

	Gender	AvgSale
1	F	250000.00
2	M	268750.00

3.3 子查詢

EX1 : 請查詢高於平均時薪的員工。

<ANS1> :

- (1) 首先必須先知道平均時薪多少
 - (2) 再找出高於平均時薪的員工
- 原本要分兩步驟做的事情，利用子查詢可以一次完成！

```
SELECT BusinessEntityID, Rate
FROM HumanResources.EmployeePayHistory
WHERE Rate >
      (SELECT AVG(Rate)
       FROM HumanResources.EmployeePayHistory)
```

(註解：HumanResources.EmployeePayHistory 資料表的 Rate 欄位為時薪)

	BusinessEntityID	Rate
1	1	125.50
2	2	63.4615
3	3	43.2692
4	4	23.72
5	4	29.8462
6	5	32.6923

EX2 :

請使用 HumanResources.Employee、HumanResources.Department 、 HumanResources.EmployeeDepartmentHistory 這三個表，列出這些員工的工作職稱(Jobtitle)、性別(Gender)，以及在哪些部門(DepartmentID 和 Name)。

- 首先必須知道每個DepartmentID對應的部門為何，稱表A
- 再利用表A找出HumanResources.Employee中的員工為哪個部門
- 利用子查詢一次完成！

<ANS2> :

```

SELECT e.BusinessEntityID, e.JobTitle, e.Gender, A.DepartmentID, A.Name
FROM HumanResources.Employee e JOIN (
    SELECT edp.BusinessEntityID, dp.DepartmentID, dp.Name
    FROM HumanResources.Department dp,
        HumanResources.EmployeeDepartmentHistory edp
    WHERE dp.DepartmentID=edp.DepartmentID
)
A
ON e.BusinessEntityID=A.BusinessEntityID

```

	BusinessEntityID	JobTitle	Gender	DepartmentID	Name
1	217	Document Control Manager	M	12	Document Control
2	218	Control Specialist	M	12	Document Control
3	219	Document Control Assistant	M	12	Document Control
4	220	Document Control Assistant	F	12	Document Control
5	221	Control Specialist	M	12	Document Control
6	2	Vice President of Engineering	F	1	Engineering

- 天啊這到底在寫什麼東西???有沒有更好懂的寫法???

```

WITH A AS (
    SELECT edp.BusinessEntityID, dp.DepartmentID, dp.Name
    FROM HumanResources.Department dp,
        HumanResources.EmployeeDepartmentHistory edp
    WHERE dp.DepartmentID=edp.DepartmentID
)
SELECT e.BusinessEntityID, e.JobTitle, e.Gender, A.DepartmentID, A.Name
FROM HumanResources.Employee e JOIN A
ON e.BusinessEntityID=A.BusinessEntityID

```

- 若要寫很多個子查詢的話，`WITH ... AS ...`的可讀性較高！
- 例如：`WITH table1 AS A, table2 AS B ...`

3.4 新增欄位

EX1 :

請在HumanResources.Employee新增一欄Age

<ANS1> :

```

ALTER TABLE HumanResources.Employee
ADD Age NVARCHAR(15) /*將此表加入Age'，欄位格式為NVARCHAR(15) */

```

SickLeaveHours	CurrentFlag	rowguid	ModifiedDate	Age
69	1	F01251E5-96A3-448D-981E-0F99D789110D	2014-06-30 00:00:00.000	NULL
20	1	45E8F437-670D-4409-93CB-F9424A40D6EE	2014-06-30 00:00:00.000	NULL
21	1	9BBBFB2C-EFBB-4217-9AB7-F97689328841	2014-06-30 00:00:00.000	NULL
80	1	59747955-87B8-443F-8ED4-F8AD3AFDF3A9	2014-06-30 00:00:00.000	NULL
22	1	EC84AE09-F9B8-4A15-B4A9-6CCBAB919B08	2014-06-30 00:00:00.000	NULL
23	1	E39056F1-9CD5-478D-8945-14ACA7FBDCDD	2014-06-30 00:00:00.000	NULL

3.5 修改欄位屬性

EX1：請將Age從nvarchar(15)轉成int。

<ANS1>：

Step1：先在要修改欄位所屬的資料表按右鍵，點選”設計(G)”，可以看到目前個欄位的資料類型。

資料行名稱	資料類型	允許 Null
BusinessEntityID	int	<input type="checkbox"/>
NationalIDNumber	nvarchar(15)	<input type="checkbox"/>
LoginID	nvarchar(256)	<input type="checkbox"/>
OrganizationNode	hierarchyid	<input checked="" type="checkbox"/>
OrganizationLevel	int	<input checked="" type="checkbox"/>
JobTitle	nvarchar(50)	<input type="checkbox"/>
BirthDate	date	<input type="checkbox"/>
Rate	nchar(1)	<input type="checkbox"/>
StandardRate	nchar(1)	<input type="checkbox"/>
HireDate	date	<input type="checkbox"/>
Flag	bit	<input type="checkbox"/>

Step2：現在想把Age從nvarchar(15)轉成int。

`ALTER TABLE HumanResources.Employee`

`ALTER COLUMN Age INT`

3.6 刪除欄位

EX1：請將Age從HumanResources.Employee中刪除。

<ANS1>：

`ALTER TABLE HumanResources.Employee`

`DROP COLUMN Age`

3.7 填入欄位

EX1：請新增Age，並利用BirthDate填入這個員工的年齡。

<ANS1>：

Step1：在 HumanResources.Employee 表中新增欄位 Age。

`ALTER TABLE HumanResources.Employee`

`ADD Age INT`

Step2 : 將今年減去 BirthDate , 並填入 Age。

```
UPDATE HumanResources.Employee
```

```
SET Age=2018-YEAR(BirthDate)
```

(註解：因為 BirthDate 的資料型態為 date , 因此 YEAR (BirthDate)就會是
這個人的出生年。)

SickLeaveHours	CurrentFlag	rowguid	ModifiedDate	Age
69	1	F01251E5-96A3-448D-981E-0F99D789110D	2014-06-30 00:00:00.000	49
20	1	45E8F437-670D-4409-93CB-F9424A40D6EE	2014-06-30 00:00:00.000	47
21	1	9BBBFB2C-EFBB-4217-9AB7-F97689328841	2014-06-30 00:00:00.000	44
80	1	59747955-87B8-443F-8ED4-F8AD3AFDF3A9	2014-06-30 00:00:00.000	44
22	1	EC84AE09-F9B8-4A15-B4A9-6CCBAB919B08	2014-06-30 00:00:00.000	66
23	1	E39056F1-9CD5-478D-8945-14ACA7FBDCDD	2014-06-30 00:00:00.000	59

3.8 建立資料表

EX1 : 請新增一個成績表。

<ANS1> :

```
CREATE TABLE score --填入表名稱
(studenyID char(10), --欄位名稱 屬性
gender char(5),
chinese int,
math int,
english int)
```

3.9 讀入 csv

EX1 : 請inputsample.csv匯入成績表內。

<ANS1> :

```
BULK INSERT score
FROM 'D:\Sql_system_database\inputsample.csv'
WITH (
    FIRSTROW = 2, --CSV 第一列為欄位名稱, 因此從第2列開始讀取資料
    FIELDTERMINATOR = ',', --CSV 使用逗號分割欄位
    ROWTERMINATOR = '\n' --CSV 使用換行分割每筆資料
)
```

第 4 章 實例操作

顆顆~我們來場 Hackathon 吧！

比賽規則：

- ✓ 時間：50min~60min
- ✓ 5~6 個人一組，下面有 ABCDE 五份題目，一組隨機指派一份題目，每份題目都有四題，難度由淺至深，最後一題最難。每一題的 sample output 截圖請派一名組員到前方找講師領取。
- ✓ 將測試好、寫完整的 SQL 語言上傳至 GOOGLE 表單作繳交。如果沒問題會在計分板上呈現分數
- ✓ 最快通過四題測驗的小組獲勝。

小提示：善用助教 XD

[A 組題目]

- (1) 查詢名字 FirstName A 開頭的員工有哪些人，查詢結果要有員工的編號 (BusinessEntityID)、名字 FirstName、性別 Gender 和年齡 Age
- (2) 請求出 32 歲的女性的員工，查詢結果要有員工編號 BusinessEntityID、年齡 Age、性別 Gender、年資(即為 2018-HireDate 的年份)、部門名稱 Name 以及 Jobtitle。並將結果根據年資大到小排列。
- (3) 市場部門 (Name='Marketing') 的員工，男女共有多少人，查詢結果要有性別 Gender、人數
- (4) 請查詢出每個部門領過最低時薪的人以及時薪金額，查詢結果要有部門編號 DepartmentID、部門名稱 Name、員工編號 BusinessEntityID、員工名字 FirstName 與 LastName)、時薪金額 Rate)

[B 組題目]

- (1) 查詢名字 FirstName B 開頭的員工有哪些人，查詢結果要有員工的編號 BusinessEntityID、名字 FirstName、性別 Gender 和年齡 Age
- (2) 請求出 40 歲的男性的員工，查詢結果要有員工編號 BusinessEntityID、年齡 Age、性別 Gender、年資(即為 2018-HireDate 的年份)、部門名稱 Name 以及 Jobtitle。並將結果根據年資大到小排列。
- (3) 金融部門 Name='Finance' 的員工，男女共有多少人，查詢結果要有性別 Gender、人數
- (4) 請查詢出每個部門領過最高時薪的人以及時薪金額，查詢結果要有部門編號 DepartmentID、部門名稱 Name、員工編號 BusinessEntityID、員工名字 FirstName 與 LastName)、時薪金額 Rate)

[C 組題目]

- (1) 查詢名字 FirstName C 開頭的員工有哪些人，查詢結果要有員工的編號 BusinessEntityID、名字 FirstName、性別 Gender 和年齡 Age
- (2) 請求出 32 歲的男性的員工，查詢結果要有員工編號 BusinessEntityID、年齡 Age、性別 Gender、年資(即為 2018-HireDate 的年份)、部門名稱 Name 以及 Jobtitle。並將結果根據年資大到小排列。
- (3) 產品部門 Name='Production' 的員工，男女共有多少人，查詢結果要有性別 Gender、人數
- (4) 請查詢出每個部門領過最高時薪的人以及時薪金額，查詢結果要有部門編號 DepartmentID、部門名稱 Name、員工編號 BusinessEntityID、員工名字 FirstName 與 LastName)、時薪金額 Rate)

[D 組題目]

- (1) 查詢名字 FirstName D 開頭的員工有哪些人，查詢結果要有員工的編號 BusinessEntityID、名字 FirstName、性別 Gender 和年齡 Age
- (2) 請求出 40 歲的女性的員工，查詢結果要有員工編號 BusinessEntityID、年齡 Age、性別 Gender、年資(即為 2018-HireDate 的年份)、部門名稱 Name 以及 Jobtitle。並將結果根據年資大到小排列。
- (3) 行銷部門 Name='Sales' 的員工，男女共有多少人，查詢結果要有性別 Gender、人數
- (4) 請查詢出每個部門領過最低時薪的人以及時薪金額，查詢結果要有部門編號 DepartmentID、部門名稱 Name、員工編號 BusinessEntityID、員工名字 FirstName 與 LastName)、時薪金額 Rate)

[E 組題目]

- (1) 查詢名字 FirstName S 開頭的員工有哪些人，查詢結果要有員工的編號 BusinessEntityID、名字 FirstName、性別 Gender 和年齡 Age
- (2) 請求出 34 歲的男性的員工，查詢結果要有員工編號 BusinessEntityID、年齡 Age、性別 Gender、年資(即為 2018-HireDate 的年份)、部門名稱 Name 以及 Jobtitle。並將結果根據年資大到小排列。
- (3) 工程部門 Name='Engineering' 的員工，男女共有多少人，查詢結果要有性別 Gender、人數
- (4) 請查詢出每個部門領過最高時薪的人以及時薪金額，查詢結果要有部門編號 DepartmentID、部門名稱 Name、員工編號 BusinessEntityID、員工名字 FirstName 與 LastName)、時薪金額 Rate)