05
資料儲存方式
(Understanding Data Storage)

5-1	正規化
5-2	主鍵、外來鍵、複合鍵
5-3	索引

5-1. 正規化

◆ 正規化說明

資料庫正規化（Normalization），是指將資料庫邏輯設計簡化，以達到最佳效能的過程。

將一張原始資料表，依據正規化形式條件，分割成數張資料表的過程，稱之為「正規化」。

透過刪除重複性和不一致的相依性，保護資料並讓資料庫更有彈性，可避免新增、修改、或刪除資料時，所可能產生的更新異常現象，確保資料的正確與完整。

良好的資料庫邏輯設計，可替最佳的資料庫和應用程式效能奠定根基。不良的資料庫邏輯設計，則會妨礙整個系統的效能。在資料庫中，重複的資料不僅浪費磁碟空間，也會產生維護方面的問題。

◆ 正規化的型式

1) 第一正規化型式（First Normal Form，簡稱 1NF）：移除重複的群組，定義出唯一的索引值。
2) 第二正規化型式（Second Normal Form，簡稱 2NF）：所有非主鍵欄位，只能「功能相依」於主鍵。
3) 第三正規化型式（Third Normal Form，簡稱 3NF）：刪除任一非主鍵欄位相依於另一非主鍵欄位。
4) 第四正規化型式（Fourth Normal Form，簡稱 4NF）：也稱為「Boyce Codd 正規形式」（BCNF）。
5) 第五正規化型式（Fifth Normal Form，簡稱 5NF）。

大部分應用程式，都是以第三正規化形式為最高階的正規化形式。使用者若是沒有遵守正規化的規則，並不會影響資料庫功能，只是資料庫的設計不夠完美而已。

經過正規化的資料表，必須對其做額外的處理，會降低系統的效能，因此有時為了改善資料庫的執行效率，需要對資料庫進行反正規化的動作。

◆ 第一正規化

- 將每個資料表中重複的群組刪除。
- 為每一組關聯的資料建立不同的資料表。
- 使用主索引鍵識別每一組關聯的資料。
- 資料庫必須排除複合索引鍵、重複的資料列、外部索引鍵、重複的群組。

  使用第一正規化型式時，不要在單一資料表中，使用多重欄位儲存類似的資料。

◆ 第二正規化

- 為可套用於多筆記錄的多組值，建立不同的資料表。
- 使用外部索引鍵，讓這些資料表產生關聯。
- 使用第二正規化型式時，記錄不應依賴資料表主索引鍵之外的索引鍵，所有非主鍵欄位，只能「功能相依」於主鍵，但是必要時可使用複合索引鍵。

◆ 第三正規化

- 刪除不依賴索引鍵的欄位，亦即刪除任一非主鍵欄位相依於另一非主鍵欄位。

資料庫經過正規化之後，會分成多個關聯的資料表，具有少量資料行與數個窄小資料表，這是正規化資料庫的特性，合理的正規化通常可改善效能。反之，未經過正規化的資料庫，具有較多資料行且較寬的資料表。

◆ **使用資料庫正規化具有下列優點：**

- 可快速建立排序和建立索引。
- 具有較多的叢集索引。
- 具有較窄、較精簡的索引。
- 每個資料表具有較少資料行，且有較少的索引。可改善 INSERT、UPDATE 和 DELETE 陳述式的效能。
- 較少 Null 值和較少機會發生不一致，如此可增加資料庫的壓縮度

5-2 主鍵、外來鍵、複合鍵

在資料庫中，可用來唯一（unique）識別出資料表中，某一筆記錄的欄位或欄位組合稱為「鍵」（Key）。「鍵」（Key）依其在不同資料表之間的關係與用途又區分為：

- **主鍵Primary Key**

  每個主鍵中的每一筆資料都是表格中的唯一值。換言之，它是用來獨一無二地確認一個表格中的每一行資料。

  使用者可以藉由這些屬性欄位，識別資料表內每一筆記錄，並提供資料索引，因此設計資料表時，要慎選主鍵。

  一個關聯表只能有一個主鍵，但主鍵可以包含一個或多個欄位，若主鍵包含多個欄位，此時就稱為複合鍵（Composite Key）。

◆ 複合式主鍵

資料表內的主鍵，可以使用一個或一個以上欄位記錄做為主鍵，這種由兩個以上欄位所組成的主鍵稱之為「複合式主鍵」。

◆ 候選鍵Candidate Key

在一個關聯表中，符合主鍵條件的屬性集合可能有好幾個，這些屬性集合稱之為「候選鍵」。而主鍵便是由候選鍵中所挑選出來的。要成為合格的候選鍵，必需滿足唯一性 (Uniqueness Property)，亦即在一個關聯表中每一橫列至少有一個欄位值是與其他橫列不相同。

◆ 次要鍵Secondary Key

資料庫系統中，除了使用主鍵這個唯一的識別值做為索引之外，也可以設定其他欄位做為資料索引。

◆ 外來鍵 Foreign Key

外來鍵（Foreign Key），簡稱FK，這個欄位會存放其他資料表的主鍵。外來鍵主要用來確定資料的參考完整性 。

◆ 超級索引鍵、複合鍵、SuperKey

在每個資料表中，除了有主鍵、次要鍵做為資料庫的資料索引外，也可以利用兩個以上的欄位做組合，而產生一組可以用作資料識別的新欄位，這種組合出來的欄位，就是超級索引鍵，其實也就是非主鍵的複合索引鍵。

5-3 索引

- **索引**

  資料庫中的索引，其功能類似書籍的目錄或圖書館的圖書分類，可以加快使用者，從資料表或檢視中，擷取特定資料列的速度。

  索引鍵儲存在B 型樹狀結構（B-tree，Balanced Tree，平衡樹）中，使用者可以快速有效地找到與索引鍵值相關的一或多個資料列。

  一個設計良好的索引可縮短存取資料表的時間，並耗用較少的系統資源，因此可改善查詢效能，尤其當資料量非常龐大時，索引設計適當與否就格外重要。

◆ 索引(續)

索引對於包含 INSERT、UPDATE、或 DELETE 陳述式的各種查詢非常有用，但不當的索引反而會增加INSERT、UPDATE、或 DELETE 資料所需的時間，此時不僅無法達到最佳效能，還會浪費磁碟空間。

索引雖可增加查詢的效能，並非資料表內的每個欄位，都必須建立索引。

一般只會將索引，建立在經常用來做搜尋的欄位，例如WHERE 子句、ORDER BY 等子句中。

◆ 索引類型

根據資料庫系統的功能，使用者可以在資料庫設計工具中，建立三種索引類型：唯一索引、主索引和叢集索引。

◆ 建立索引

一個索引可以涵蓋一或多個欄位。建立索引的語法如下：

```
CREATE [UNIQUE] INDEX index_name ON
table_name(column_1[ASC | DESC], column_2[ASC | DESC], .....)
```

- UNIQUE ：建立索引時，可宣告UNIQUE 來確保索引鍵值的唯一性、不重複。
- index_name：索引名稱，索引的命名沒有固定的方式。通常會在名稱前加一個字首，例如 “index_” ，來避免與資料庫中的其他物件混淆。
- table_name：欲建立索引的資料表名稱。
- column_1， column_2：建立索引的欄位名稱（一個索引可以涵蓋一或多個欄位）。
- ASC：Ascending 的縮寫，代表依鍵值遞增建立索引。
- DESC：Deccending 的縮寫，代表依鍵值遞減來建立索引。

◆ 刪除索引

當使用者不需要資料表中的某些索引時，可以將該索引刪除。刪除索引的語法如下：

```
DROP INDEX index_name ON table_name(column_1, column_2, .....)
```

- index_name：欲刪除的索引名稱。
- table_name：欲刪除索引的資料表名稱。
- column_1， column_2：欲刪除索引的欄位名稱（一個索引可以涵蓋一或多個欄位）。

◆ 唯一的索引

唯一的索引，即為兩個資料列中，不允許有相同的索引值。

如果現有資料中，含有重複的關鍵值，則您無法在大部分的資料庫中，儲存含有新建立唯一的索引之資料表。您可能也無法在資料庫中，增加可在資料表中建立重複索引值的新資料。

◆ 主索引鍵索引

資料庫資料表，通常都具有一資料行或資料行組合，其中該資料行的索引值是唯一可辨識資料表中的每個資料列。這個資料行即為資料表的主索引鍵。

在資料庫圖表中定義資料表的主索引鍵的同時，便會自動建立主索引鍵索引，該索引則為唯一索引的一種特定類型。這個索引的主索引鍵值必須是唯一的。當您在查詢中使用主索引鍵時便可快速存取資料。

◆ 叢集索引與非叢集索引

索引可分為：叢集索引與非叢集索引。

➢ 叢集索引

叢集索引是將資料表或檢視中的資料列，依其索引鍵值排序與儲存，這些就是索引定義中包含的資料行。因為資料列本身只能以一種順序排序，所以每個資料表只能有一個叢集索引。

若資料表有叢集索引時，資料表又稱為叢集資料表。若資料表沒有任何叢集索引，資料列就儲存在未經排序的結構中，此結構稱為堆積結構。

➢ 非叢集索引

非叢集索引是與資料列完全分開的結構。非叢集索引包含非叢集索引鍵值，每個索引鍵值項目都有一個指標，該指標指向包含索引鍵值的資料列。

叢集索引是將資料與索引儲存在一起，每個表格只能有一個叢集索引，且資料是經過排序，速度較快。而非叢集索引是將資料與索引分開儲存，類似指標的概念，且資料可能未經排序。在資料表中可以設定多個非叢集索引，但只能設定一個叢集索引，因為非叢集索引不會影響實際資料的排列順序。

➢ 叢集索引使用時機

1. 欄位內包含大量不同資料。
2. 經常做為排序依據的欄位。

補 充 說 明

- **實體完整性**

  強制資料完整性可確保資料庫中的資料品質。

  設計資料表過程中兩個重要的步驟是識別資料欄的有效值，以及決定如何強制資料欄中資料的完整性。資料完整性分成下列類別1.　實體完整性

  2. 值域完整性
  3. 參考完整性
  4. 使用者自訂完整性

◆ 實體完整性

1. 實體完整性，定義一資料列作為特定資料表的唯一實體。
2. 實體完整性透過 UNIQUE 索引、UNIQUE 條件約束或 PRIMARY KEY 條件約束，強制識別碼資料行或資料表主索引鍵的完整性。

◆ 值域完整性

1. 值域完整性，是指特定資料行的項目有效性。
2. 可以強制值域完整性，藉由使用資料類型來限制類型、使用 CHECK 條件約束與規則來限制格式、或使用 FOREIGN KEY 條件約束、CHECK 條件約束、DEFAULT 定義、NOT NULL 定義與規則來限制可能值的範圍。

◆ 參考完整性

1. 參考完整性，可在輸入或刪除資料列時，保留資料表之間已定義的關聯性。
2. SQL Server 中，參考完整性是以外部索引鍵與主索引鍵間的關聯性，或外部索引鍵與唯一索引鍵間的關聯性為基礎（透過 FOREIGN KEY 和 CHECK 條件約束）。
3. 參考完整性，可確保索引鍵值在各資料表間的一致性。這種一致性要求不可參考不存在的值，且若索引鍵值發生變更，資料庫中所有指向它的參考也必須一致變更。
4. 當您強制參考完整性時，SQL Server 會防止使用者執行下列動作：
5. 若主要資料表中沒有關聯的資料列，會在相關的資料表中加入或變更資料列。
6. 變更主要資料表中的值，而在相關資料表內產生遺棄的資料列。
7. 若有符合的相關資料列，會從主要資料表中刪除資料列。

◆ 使用者自訂完整性

1. 使用者自訂完整性，可讓您定義不屬於其他完整性類別目錄的特定商務規則。
2. 所有完整性類別目錄，都支援使用者自訂完整性。其中包括 CREATE TABLE、預存程序與觸發程序中，所有資料行層級與資料表層級的條件約束。

◆ 笛卡兒乘積

➢ CROSS JOIN - 交叉連接

笛卡兒乘積（Cartesian product），為兩個資料表間的交叉連結，兩個資料表在結合時，將兩個資料表中所有的可能排列組合出來。

➢ CROSS JOIN 語法（1）

```
SELECT table_column1, table_column2 · · ·
FROM table_name1
CROSS JOIN table_name2;
```