

小朱® 的技術隨手寫

本部落格不吹捧、不自戀、不造神，也絕不用技術文偷渡政治文，請各位朋友安心服用。

2019-11-24

請用「空白」區分關鍵字

Q

[Database] 到底該不該用 GUID 當 PK?

👁 4537 💬 0 📄 SQL Server and SQL Azure

在資料庫設計上，PK (主鍵) 一直都是設計的重點之一，但主鍵使用的資料型態有可能直接或間接影響到資料庫的存取效能，然而技術是會演進的，30年前可能不行的作法在30年後的今天是否仍然不適用? 似乎有待商榷。

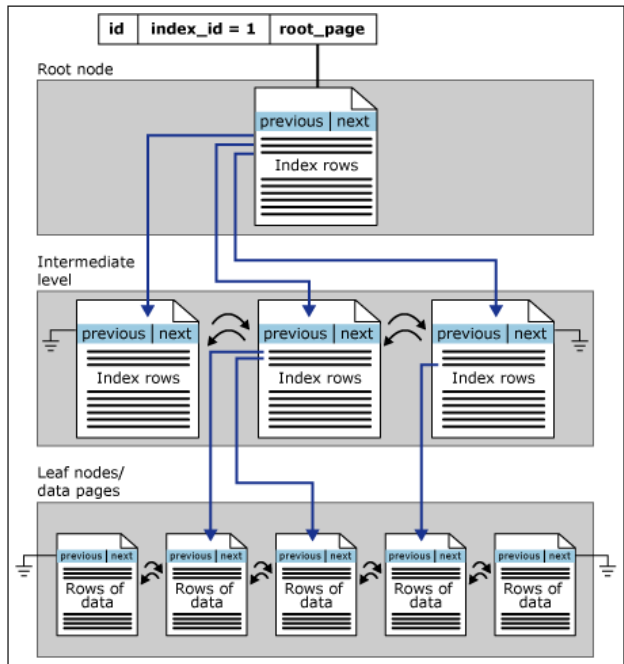
技術的選擇與評估一定要考慮到環境 (environment) 與情境 (situation)，否則很容易產生偏見或誤解。

主鍵 (Primary Key) 在資料庫設計中一直都是重要的因子，它具有唯一性 (unique)，與其他資料表鍵值的連結 (Foreign Key, FK)，達成邏輯上的參考完整性 (Referential Integrity)，有助於消除重覆性資料以及提升資料的可維護性，然而在實務上，參考完整性愈強的資料庫，在異動鍵值的運算 (最常見的是新增資料與刪除資料) 會觸發對參考目標的資料表的鍵值比對運算，會增加些許的比對損耗。不過，雖然鍵值在資料庫設計中很重要，但實際上資料庫在運作時，主鍵和外來鍵組成的參考完整性運算，使用的是索引 (index)，索引連結到資料儲存的位址，一個資料庫如果沒有任何索引的話，存取資料的速度會慢到超乎想像，所以**資料庫內一定要有索引，否則每次的 SQL 指令運算都要去掃一次資料表所有資料，光是掃描資料的 I/O 速度就足以讓人哭出來了。**

I/O 包含記憶體存取與磁碟存取，但資料庫一般都是儲存在磁碟中，所以資料庫的 I/O 一般都是指磁碟的 I/O 動作，除非資料庫是放在記憶體內或是放在 I/O 速度高的設備如 SSD 或磁碟陣列外，在效能調校時 I/O 動作還是要考慮在內。

資料庫索引概念

索引是一種資料庫物件，它會以設定的條件在資料庫的儲存區域內建立一個搜尋樹 (Search Tree)，資料庫所使用的搜尋樹主流是 B+-tree，是一種在新增、刪除節點時會執行自我平衡 (self-balanced) 的搜尋樹，與 B-tree 的差異是它在子葉節點中亦存有指向鄰近子葉節點的指標，可快速執行掃描作業 (Index Scan)，同時亦保有搜尋樹的搜尋能力 (Index Seek)。

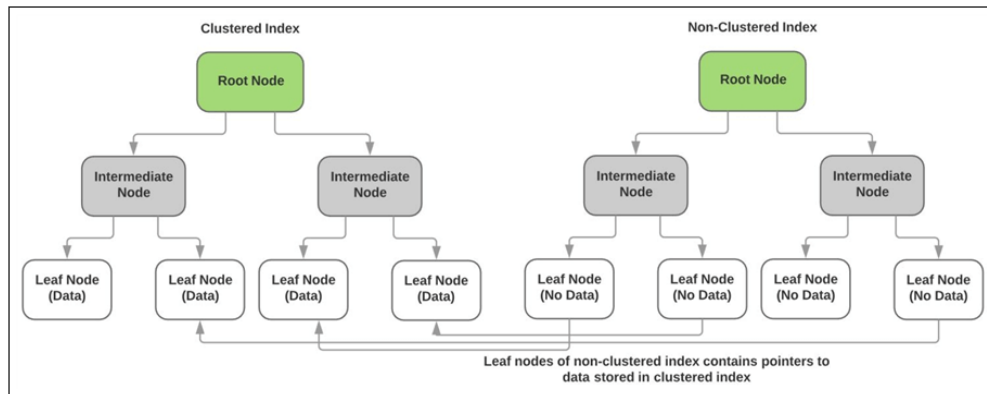


另外，在 SQL Server (其他的 DBMS 也類似) 內的索引分為 Clustered Index 以及 Nonclustered Index，這兩者的差異在於 **Clustered Index 在生成時就會依照指定鍵值進行排序** (節點內會包含該鍵值)，Nonclustered Index 則只是組織資料列位址到節點內而已，但若 Clustered Index 和

標籤雲

[.NET](#) [.NET Framework](#) [ADO.NET](#) [ASP.NET](#) [ASP.NET MVC](#) [C#](#) [Cloud Computing](#) [Data Binding](#) [IADP](#) [Intel AppUp Developer Program](#) [JavaScript MCP](#) [OAuth](#) [Office Development](#) [Open Authorization](#) [ORM](#) [Security](#) [Silverlight](#) [SQL Azure](#) [SQL Server](#) [VB.NET](#) [Visual Studio](#) [VS2010](#) [WCF](#) [Windows 8](#) [Windows Azure](#) [Windows Azure Platform](#) [Windows Azure SDK](#) [中鳥](#) [基礎概念](#) [學生](#) [學習](#) [架構設計實戰](#) [程式開發](#) [經驗談](#) [考試](#) [經驗談](#) [職場](#) [菜鳥](#)

序，若 Clustered Index 不存在時則會僅會指向資料列的位址。想當然，當任何索引都不存在時，資料列位址僅會存在於資料儲存的邏輯結構內。



SQL Server 預設會將 PK 設置為 Clustered Index，因 Clustered Index 會自動形成有序資料 (Ordered Data)，使得許多涉及排序作業的運算都會使用到 Clustered Index (其實較廣泛的運算就是範圍查詢，例如日期區間或值域等)，可想見沒有 Clustered Index 會對效能的傷害有多大 (因為都要先掃描再排序，而不是直接取一個已排序的資料集，在沒有 Clustered Index 的時候會導致 Table Scan)。

索引與資料型態的選用

前面提到 Clustered Index 會形成有序資料，在不影響 B+ tree 插入資料時期排序的速度下，以可自動形成順序的資料最為有利，因此很多人都會使用 int, bigint 這類具有快速判斷順序的資料型態作為 Clustered Index 的材料，除了有序性外，另一種影響到索引新增速度的因素就是資料長度，在新增資料時若節點可用空間不足以置放新增的資料時，會產生頁分裂 (Page Split) 的效應，因此若作為索引的資料型態太長的話，發生索引分裂的機率會較高，另外空間不足的分頁也無法存放資料，空間碎裂 (Fragment) 的情況會較嚴重，也就是分頁內的可用空間可能會有一定比例是沒使用到的。

SQL Server 的每一個分頁的大小是 8,060 bytes (約 8KB)

索引的空間碎裂問題可以用重建索引的方式解決，但頁分裂會使得 I/O 次數增加，影響索引新增節點的速度，不過若資料庫所在的磁碟是具有高速 I/O 能力的磁碟，這點效能損耗或許還可以忽略，只是若是很要求時間的應用程式就一定要正視這個問題。

GUID於主鍵與索引的使用

GUID / UUID 是一種唯一性很強的資料形態，依照 GUID 的演算法，想要得到重覆的 GUID，得要在 85 年的時間內每秒都產 10 億個 GUID 才可能會發生重覆，或是剛好電腦環境有問題才有可能出現，所以 GUID 通常拿來當需要很強唯一性的識別資料之用。那麼 GUID 是否能做為主鍵？答案幾乎是肯定的，因為要作為 PK 的鍵值本身就要有很強的唯一性，所以 GUID 的特性相當符合 PK 的要求，若使用 int, bigint 等型態作為 PK 的話，可能還需要自行編寫演算法處理重覆問題，不過 GUID 的生成方式可能會有些許影響，尤其是跨不同類型資料庫 (如混合 SQL Server, MySQL 等 DBMS 的環境)，相同的 API 無法直接運用。

會決定 GUID 適用與否的其實不是 PK/FK，而是索引。

前面有提到 Clustered Index 是有序的資料，但 GUID 一般都是採用隨機亂數方式產生 (也就是 version 4 的 GUID 演算法)，當然也有資料庫實作有序的 GUID (例如 SQL Server 的 Sequential ID)，隨機亂數產生的 GUID 不利於 Clustered Index 的處理，尤其是新增資料時的排序作業，會增加 I/O 的次數。不過這個問題只發生在 Clustered Index 上，若是使用到 Nonclustered Index 則沒有這個問題。

因此要避過 GUID 用於 Clustered Index 的問題，使用代理鍵 (Surrogate Key) 是個不錯的作法，使用一個可判斷排序的資料型態作為 Clustered Index，然後將 GUID 的欄位設為 Nonclustered Index，此時 Nonclustered Index 會指向 Clustered Index，可降低因為排序作業導致的效能損耗，不過新增作業上的 I/O 則會略增，不過若不是非常頻繁的新增資料，新增作業的 I/O 倒是可以忽略不看，除非很計較新增資料的 I/O 速度。

索引內產生較多的碎裂狀態的話，會讓索引使用空間增加不少，但現代的磁碟空間都很大，儲存問題應該不大，但若是要涉到跨網路複寫或是資料庫備份的話就會有一定的影響。

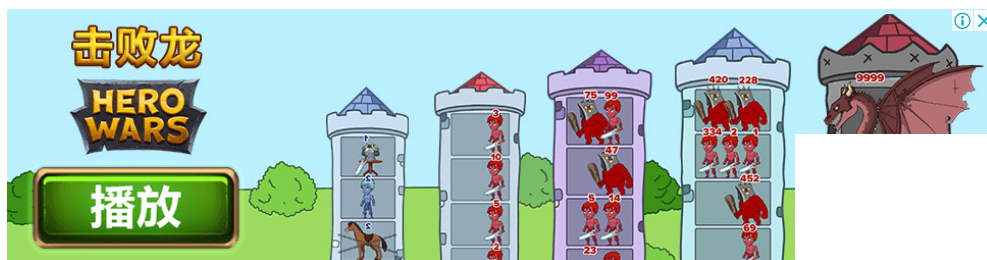
參考1：<https://www.slideshare.net/yftzeng/btreepart-1>

參考2：<https://blog.darkthread.net/blog/guid-as-pk-on-db/>

參考3：<https://docs.microsoft.com/zh-tw/sql/relational-databases/indexes/clustered-and-nonclustered-indexes-described?view=sql-server-ver15>

Database • Performance • Index

[回首頁](#)



關聯文章

SQL Server 的 ORDER BY 兩三事

[.NET][LINQ] Any() vs. Count() 何時可用? 何時不可用?

讓資料保持彈性的設計：Profile 架構

[.NET][Database] 資料庫中的 NULL 值

[Data Access] ORM 原理 (11): 效能議題

系列文章

在做 SQL Server 組態工作時，使用 Management Studio 很方便，但 T-SQL 會更有效率。

Troubleshooting: SQL Server 2008 在資料庫物件有修改時，查詢編輯器的 Intellisense 無法即時反應的問題

[VS2010] 好用的 SQL Azure 資料庫開發的輔助工具：SQL Azure Explorer

[SQL Server] 在安裝時未將管理帳戶加入 SQL Server 的補救措施

[SQL Server] 游標使用的藝術

[SQL Server] 鎖定使用的藝術 (Part 1) - 鎖定控制類型

[SQL Server] 鎖定使用的藝術 (Part 2) - 隔離層次 (Isolation Level)

[SQL Server] 簡單產生亂數字串

[SQL Server] SQL Server Data Tools (SSDT)

[SQL Server] SQL Server 2012 的 Core-Based License 授權模式

[Windows Azure] 使用 Linked Server 方式連接 Windows Azure SQL Database

[SQL Database] 讓 SQL Database Migration Wizard 支援繁體中文系統

[SQL Server] SQL Server Management Studio 啟動失敗 (無法讀到 Registered Servers) 的問題與解決

[Database] 到底該不該用 GUID 當 PK?

[Security] ...

5年前 • 1 則留言

IT 技術人的部落格平台

6年前 • 2則留言

快來開設專屬部落格！分享心得、散播你的熱情 :D

[C#] 在 Visual Studio 2015 使用 C# 7

5年前 • 3則留言

昨天在 "C# 7 不能編譯?" 一文的最後，我有提到不是一定要 Visual Studio 2017 ...

[Troubleshoot Server] 解決

6年前 • 1 則留言

這是今天在處 ...

0則留言

DotBlog

Disqus 隱私政策

1 登錄

Favorite 1

Tweet

分享

按最新排序



開始討論...

通過以下方式登錄

或注冊一個 DISQUS 帳號 ?

姓名

來做第一個留言的人吧！

訂閱 將Disqus添加到您的網站 添加 Disqus 添加 不要出售我的數據

本頁段落

- 資料庫索引概念
- 索引與資料型態的選用
- GUID於主鍵與索引的使用

