






GUID Primary Key資料庫避雷守則

 2016-01-29 07:16 AM  25  42,460

【聲明】該不該用GUID當Primary Key是可以讓開發人員大戰三百回合的好題材，由標題可知我屬於GUID陣營，這篇文章不打算花時間論證該不該用GUID PK，假設讀者已接受使用GUID當PK，只聚焦如何避免GUID PK導致資料庫效能悲劇。

故事源起MVP James最近寫的幾篇[GUID鬼故事](https://www.dotblogs.com.tw/jamesfu/series/1?qq=GUID)

[<https://www.dotblogs.com.tw/jamesfu/series/1?qq=GUID>](https://www.dotblogs.com.tw/jamesfu/series/1?qq=GUID)（包含一起寫入資料3秒變40秒的[案例](#)

[<https://www.dotblogs.com.tw/jamesfu/2016/01/18/guid_1>](https://www.dotblogs.com.tw/jamesfu/2016/01/18/guid_1)，也實證了GUID作為叢集索引造成[索引破碎現象](#)

[<https://www.dotblogs.com.tw/jamesfu/2016/01/20/guid_2>](https://www.dotblogs.com.tw/jamesfu/2016/01/20/guid_2)，值得一看），讓我有所警覺，身為一個偏好GUID Primary Key的開發者，有必要正視這個問題，避免掉進資料庫效能陷阱。

簡單歸納我愛用GUID當Primary Key的理由：

1. 不仰賴資料庫就能取得唯一鍵值

建立新資料的當下就能決定PK，放心地使用它衍生相關應用。即使程式執行當下資料庫還沒建好、三個月後才會寫入資料庫，或是一年後來自十個資料庫的資料要合併成一個資料表，都不必擔心鍵值重複的問題。（註：傳說GUID仍有重複的可能，但機率極低，在此忽略）

2. 無法被猜測，用於參數時內含安全防護效果

但使用GUID當Key也有缺點，例如：

1. 不利於人工查詢或偵錯

例如：GetTradeData.aspx?no=617ad98a-c010-4cd2-bc07-9a64d907154f，SELECT * FROM MyTable WHERE Id = '617ad98a-c010-4cd2-bc07-9a64d907154f'，大多得靠複製貼上處理參數，難以手工輸入。

2. 增加儲存空間

每個GUID有16 Byte，相較於整數4 Byte，多耗用4倍的儲存空間。

3. 衝擊資料庫效能

GUID的非連續特性，易導致索引破碎（Index Fragement），降低系統效能。

依我的看法，人工查詢不便頂多費工不會致命，12 Byte vs 4 Byte在資料筆數很龐大時才有顯著影響，而第3點可能嚴重影響效能，輕忽不得。

由James提供的案例可知，在資料庫使用GUID PK，稍有不慎便會發生悲劇。所以該好好思索，如果你打算使用GUID PK，要怎麼樣才能避免掉下效能陷阱？

首先，鬼故事裡有一個共同關鍵 - 「GUID PK被設成叢集索引（Clustered Index）」，我們都知道，資料庫的索引有兩種，Clustered Index與Nonclustered Index，依據[MSDN <https://msdn.microsoft.com/en-us/library/ms190457.aspx?WT.mc_id=DOP-MVP-37580>](https://msdn.microsoft.com/en-us/library/ms190457.aspx?WT.mc_id=DOP-MVP-37580)，二者比較如下：

» Clustered Index

叢集索引決定資料表儲存資料的實際順序，每個資料表最多只能有一個叢集索引，而叢集索引不需額外空間儲存鍵值。有叢集索引的資料表稱為叢集資料表（Clustered Table）。找到索引鍵時一併找得到資料列，查詢效能比非叢集索引好，用在最常用的鍵值（例如：Primary Key）可以產生最大效益。若資料表沒設叢集索引，資料會以沒有固定排序的方式儲存，這種儲存結構稱之為Heap。

» Nonclustered Index

非叢集索引需在實際資料列之外另行建立資料結構，每筆索引除了索引包含欄位外，還需要一個指標（Pointer）指向資料位置，若資料

限，完全涵蓋查詢條件提升效能。（參考：[Create Indexes with Included Columns <https://msdn.microsoft.com/en-us/library/ms190806.aspx?WT.mc_id=DOP-MVP-37580>](https://msdn.microsoft.com/en-us/library/ms190806.aspx?WT.mc_id=DOP-MVP-37580)。）

PK使用頻率很高，設成叢集索引對效能最有利，故慣例上會設成叢集索引以提升效能。然而，因為GUID具有不連續的隨機性，即使循序寫入資料，常常後寫的資料GUID排序較前，依叢集索引特性，實體儲存位置應擺在前段，造成每次寫入資料都需挪動調整既有資料造成索引破碎，拖累寫入與查詢效能。

由此看來，將GUID PK設為叢集索引的缺點蓋過優點。轉念一想，沒人規定PK一定要設成叢集索引呀！只要將PK改為非叢集索引就能避開GUID導致索引破碎的危機。當然，非叢集索引的效能不如叢集索引，但減損幅度不致太嚴重，相較於其降低的風險是划算的。

但這衍生另一個議題，PK不設成叢集索引，資料表就只有非叢集索引可行嗎？前面提到，沒有叢集索引的資料表稱為Heap Table。

依據MSDN https://msdn.microsoft.com/en-us/library/hh213609.aspx?WT.mc_id=DOP-MVP-37580，Heap Table只適合資料極少（例如數十筆）的場合，即使不設任何Index，Table Scan也很有效率，或者某些資料架構師會巧妙利用非叢集索引配合Row Identifier (RID，由File Number, Data Page Number, Slot on The Page組成，長度不長)，達到比叢集索引還好的效率。但絕大部分的情況下，設定叢集索引有好處：

1. 循序讀取一段資料時，叢集索引可以節省排序動作
2. GROUP BY時，分群前必須先排序，叢集索引可以省去排序作業
3. 記得避免資料多又無非叢集索引可用的狀況，如此永遠只能Table Scan，包準慢死你

而我覺得缺乏叢集索引的最致命點是 - Heap Table也會產生破碎現象，一旦出現，依MSDN的建議是建個Clustered Index再砍掉，網路上提到的其他做法還有把資料先搬出來再重新塞回去、匯出到新資料表再更名，不管哪一種做法，聽起來都好沒效率，好可怕。[2016-01-30更新]SQL 2008起，增加了[ALTER TABLE ... REBUILD <http://www.sqlmaestros.com/sql-server-alter-table-rebuild/>](http://www.sqlmaestros.com/sql-server-alter-table-rebuild/) 指令，背後使用建立Clustered Index再刪除做法，但省去了建索引過程並重新非叢集索引的代價，明顯提升效率。

至此，我得到兩點結論：1.將GUID PK設成非叢集索引利大於弊 2.資料表欠缺叢集索引就會形成Heap Table，弊大於利。所以，最好的折衷方案就是「GUID PK設成非叢集索引，並另外增設叢集索引」，而這個額外的叢集索引，自動跳號整數會是首選。如此我們降低GUID PK導致索引破碎的風險，自動跳號叢集索引避免新增資料造成索引破碎，而叢集索引讓資料表可以透過索引重建重組改善破碎狀況，同時避開索引破碎及Heap Table陷阱。

綜合以上，來段CREATE SCRIPT示意：

[排版顯示](#) [純文字](#)

```
CREATE TABLE [dbo].[MiniFlow] (
    [SeqNo] [int] IDENTITY(1,1) NOT NULL,
    [FlowId] [uniqueidentifier] NOT NULL,
    [FormCode] [varchar](4) NOT NULL,
    [FormNo] [varchar](16) NOT NULL,
    [Subject] [nvarchar](256) NOT NULL,
    CONSTRAINT [PK_MiniFlow] PRIMARY KEY NONCLUSTERED
    (
        [FlowId] ASC
    )
)
GO

CREATE CLUSTERED INDEX [IX_MiniFlow] ON [dbo].[MiniFlow]
(
    [SeqNo] ASC
)
GO
```

有幾個重點：

1. PK外外增設SeqNo INT，以IDENTITY(1,1)設定自動跳號
2. FlowId為GUID是MiniFlow資料表的Primary Key，但設定時加上NONCLUSTERED指定為非叢集索引
3. 利用CREATE CLUSTERED INDEX將SeqNo建為叢集索引

都還OK，不打算積極調整，但日後開發使用GUID PK新系統，我應該會採用這種設計方式。

最後提一下NEWSEQUENTIALID，有不少人建議用它取代GUID避免索引破碎，但為NEWSEQUENTIALID只能用於資料庫INSERT時自動產生，又可以被預測，並不符合我期待GUID PK應提供的隨時可取及防猜保密要求，我認為只適合用在處理跨資料庫合併用的鍵值。

