

記事

[Mihoko Iijima](#) · 2020年12月20日 3m read

揮発性テーブルのビットマップ・インデックスの圧縮（維持管理）方法

これは [InterSystems FAQ サイト](#) の記事です

揮発性テーブル（多数のINSERT、DELETEが行われるテーブル）では、ビットマップ・インデックス用ストレージは徐々に効率が低下する可能性があります。

例えば、以下の定義からなるデータが数千件あり、一定期間保持した後 TRUNCATE TABLE で一括削除を行うオペレーションが繰り返し行われているとします。

```
Class MyWork.MonthData Extends (%Persistent, %Populate)
{
  /// ???
  Property Satisfaction As %String(VALUELIST = ", ??,????,????,??,");
  /// ??
  Property Age As %Integer(MAXVAL = 70, MINVAL = 20);
  Index AgeIdx On Age [ Type = bitmap ];
}
```

INSERT によってできたビットマップ・インデックスのストレージのイメージ（一部）は以下の通りです。

```
?INSERT??
^MyWork.MonthDataI("AgeIdx",20,1) = $zwc(401,120,4,75,102,10,???? 958)/*$bit(5,76,103,107???
^MyWork.MonthDataI("AgeIdx",21,1) = $zwc(407,121,29,178,251,2????,732,772,898,960)/*$bit(3???
^MyWork.MonthDataI("AgeIdx",22,1) = $zwc(402,96,5,57,74,164,????,0,4)/*$bit(20,63,77,92,10???
^MyWork.MonthDataI("AgeIdx",23,1) = $zwc(133,116)_$(0,0,8,0????,64,0,4)/*$bit(20,63,77,92???
^MyWork.MonthDataI("AgeIdx",25,1) = $zwc(404,119,105,155,235????,947)/*$bit(106,156,236,30???
^MyWork.MonthDataI("AgeIdx",26,1) = $zwc(128,119)_$(0,0,0,2,????,0,128)/*$bit(26,80,115,1???
??????
```

TRUNCATE TABLE でデータを一括削除した場合、レコードデータは消えますがビットマップ・インデックス用ストレージの一部が残ります（イメージ）。

```
?TRUNCATE??
^MyWork.MonthDataI("AgeIdx",20,1) = $zwc(145,120)/*$bit()*/
^MyWork.MonthDataI("AgeIdx",21,1) = $zwc(151,121)/*$bit()*/
```

Published on InterSystems Developer Community (<https://community.intersystems.com>)

Page 2 of 2