

補足資料 -2000-7-13 HiBase 独立キー (UNBIND KEY)

2000年7月13日 西林瑞夫/ホロン株式会社

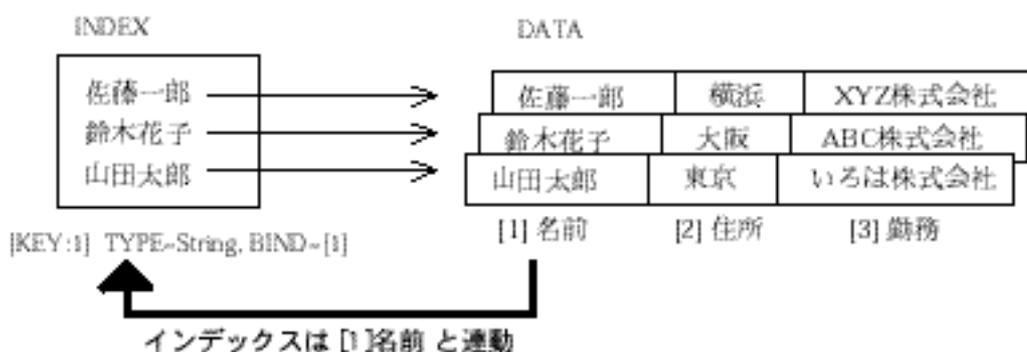
本文は、HiBase 独立キー (UNBIND KEY) の目的と具体的な利用の方法を解説するものです。

HiBase 独立キーは、HiBase-415 Rev-000720 以降のHiBase データベース管理システムで利用可能です。

目的：

HiBase 独立キー (UNBIND KEY) は、レコードと連動しないキーを提供します。

普通、HiBaseのキー (従属キー : BIND KEY) はレコード内の項目 (アイテム) と関連づけがなされています。例えば、スキーマ定義ファイルの「BIND = [1]」は、[1]項目をキーにすることを指定しています。そして、データベースに追加されるレコードの[1]項目からキーの値が取りだされ、自動的にインデックスに反映されます。同様に変更の場合も自動的にインデックスの修正がなされます。

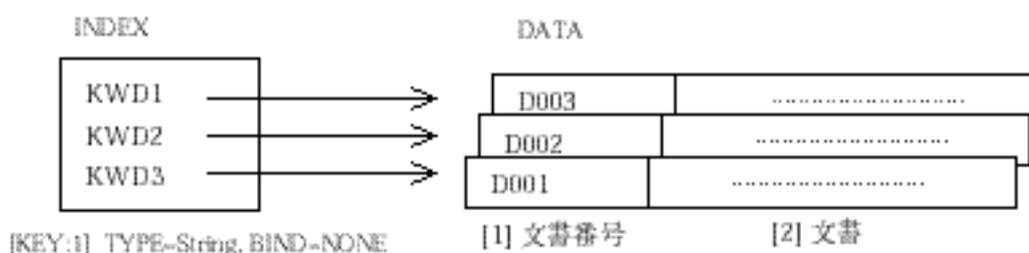


HiBase 従属キー : BIND KEY

HiBaseの従属キーは、レコード内の項目の値との関係を維持しています。

例えば、上記は[1]名前項目の値に関連づけがなされているインデックスの関係を示しています。

一方、HiBase独立キー（unbind key）は、スキーマ定義ファイルで「BIND = NONE」と定義します。つまり、レコードとの連動が無いいためインデックスのメンテナンス（キーの値の追加、変更、削除）は自動的になされず、アプリケーションプログラムが行わねばなりません



HiBase 独立キー：UNBIND KEY

HiBaseの独立キーは、レコード内の項目の値との関係が存在しません。

例えば、文書データベースを作る場合、本文（[2] 文書）への検索キーのインデックスに適しています。

このように、HiBase独立キーは、普通のキーに比べると不便です。しかし、以下のような高度な利用に効果があります。

a. マルチメディアデータベースのキー

文章、絵、音などのキーはデータと関連しません。このようなマルチメディアデータを含むスキーマには独立キーが適しています。

b. 255 以上のマルチバリューキー

HiBaseではマルチバリュー項目をキーにすることができます。しかし、レコードと連動しているためマルチバリューの制限（255以内）、レコード長の制限（物理ブロック長以内）を注意する必要があります。一方、独立キーはレコードの値との関連がないため、「値の数」の制限やレコード長の制限は一切ありません。ただし、キーの値の最大長が255バイト以内という制限は存在します。

方法：

スキーマの定義

独立キー（unbind key）は、スキーマ定義ファイルで以下のように指定します。

ex.1 独立キーを「番号：1」、「データ型：文字列」として定義します。

```
[KEY: 1], TYPE = String, BIND = NONE
```

ex.2 「BIND = NONE」を省略したなら独立キー（unbind key）の定義となります。

```
[KEY: 1], TYPE = String
```

ex.3 独立キーをメタインデックス（IgnoreJCase：ひらがな/カタカナの字種を無視）と併用することもできます。

```
[KEY: 1], TYPE = "String, IgnoreJCase", BIND = NONE
```

API

独立キー（unbind key）の操作は、以下のAPIで行います。

バインド（値とレコードの対応関係）の追加

```
ECode BindRecord (FNbr fNbr, RNbr rNbr, JWord ki d, const JByte* pKey);
```

説明

BindRecord は、インデックス（インバーテッドインデックス）にバインドを追加します。この時、パラメータには、ファイル番号：fNbr、レコード番号：rNbr、キー番号：kid、キーの値：pKeyを設定します。

バインド（値とレコードの対応関係）の削除

```
ECode UnbindRecord(FNbr fNbr, RNbr rNbr, JWord ki d, const JByte* pKey);
```

説明

UnbindRecord は、インデックス（インバーテッドインデックス）からバインドを削除します。この時、パラメータには、ファイル番号：fNbr、レコード番号：rNbr、キー番号：kid、キーの値：pKeyを設定します。

利用の際の注意点

1. HiBase API のレコード追加 (HDBHandle::InsRecord)、レコード更新 (HDBHandle::UpdRecord) において、HiBase独立キーに影響は全くありません。このことはHiBase独立キーへの値の追加、変更はアプリケーションプログラムが責任を持って行わねばならないことを意味しています。
2. HiBase API のレコード削除 (HDBHandle::DelRecord) は、HiBase独立キーインデックスから削除要求のあったレコード番号を取り去ります。言い換えると、独立キーであってもレコード削除の際のメンテナンスは自動的になされます。この場合、HiBase**独立キーインデックス全体から当該のレコード番号を探してこれを削除する**必要があるため、大量の値を持つような独立キーでは処理時間が遅くなります。つまり、HiBase独立キーを定義しているデータベースファイルでのレコード削除は従属キーのレコード削除よりも大きな負荷となるため更新の激しいデータベースファイルへの適用は避ける必要があります。