

Actian Zen v15

SQL のテーブルを Btrieve API でアクセスする



株式会社エージーテック

2022 年 9 月 30 日

## 免責事項

株式会社エージーテックは本書の使用を、利用者またはその会社に対して「現状のまま」でのみ許諾するものです。株式会社エージーテックは、いかなる場合にも本書に記載された内容に関するその他の一切の保証を、明示的にも黙示的にも行いません。本書の内容は予告なく変更される場合があります。

## 商標

© Copyright 2022 AG-TECH Corp. All rights reserved. 本書の全文、一部に関わりなく複製、複写、配布をすることは、前もって発行者の書面による同意がない限り禁止します。すべての **Pervasive** ブランド名および製品名は、**Pervasive Software Inc.** の米国およびその他の国における登録商標または商標です。また、すべての **Actian** のブランド名は、**Actian Corporation** の米国およびその他の国における登録商標または商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

**Actian Zen v15 SQL のテーブルを Btrieve API でアクセスする**

最終更新：2022 年 9 月 30 日

SQL で使用しているテーブルを Btrieve API でアクセスしたいことがあります。

この時、注意点が幾つかあります。

一つは、NULL 値を含むデータのアクセスです。

もう一つは、LONGVARCHAR 型、NLONGVARCHAR 型、LONGVARBINARY 型のデータアクセスです。

## ◆NULL 値を含むデータのアクセス

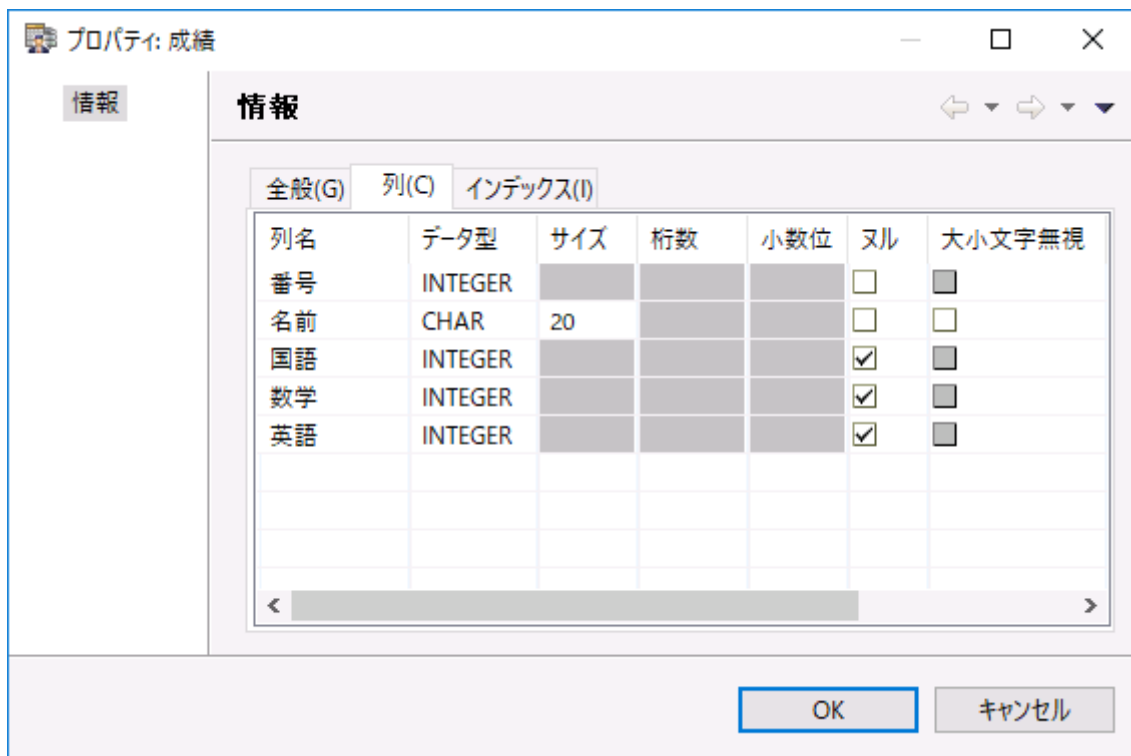
NULL 値を含むデータは、その項目の直前に 1 バイトの NULL 値を識別する領域が追加されます。

SQL でアクセスする際には意識する必要がありませんが、Btrieve API でアクセスする場合には、NULL 値を識別する領域（以降 NULL インジケータと表記します）が挿入されていることを意識する必要があります。

また、該当データが NULL 値であるか判断することも必要で、NULL インジケータが NULL 値を示す値であった場合、データは NULL として処理を行う必要があります。

例えば、次のような 3 科目のテストの結果が格納されるとします。ある学生がテストを欠席したようなケースでは、名前フィールドのみ設定し、点数フィールド（国語、数学、英語）には値を設定しないようにします。ヌル値をサポートしない場合、何も値を設定しなければ数値フィールドでは 0 を返すため、0 点だったのか欠席だったのかを識別できませんが、ヌル値をサポートすることで識別可能になるのです。

番号	名前	国語	数学	英語
1	山田太郎	欠席	100	80
2	名古屋一郎	90	欠席	70
3	大阪花子	80	60	80
平均点		85	80	76.66667

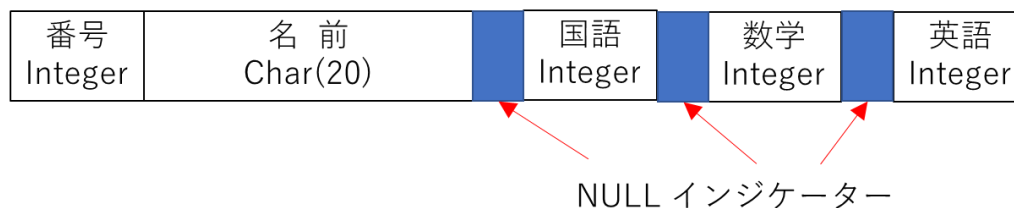


実際のレコードイメージは次のようになります。

ヌル値は無効

番号 Integer	名前 Char(20)	国語 Integer	数学 Integer	英語 Integer
---------------	----------------	---------------	---------------	---------------

ヌル値は有効



ヌル値を許可しない場合、レコード長は 36 バイトになりますが、国語、数学、英語にヌル値を許可した場合、それぞれの項目の直前に NULL インジケータが追加されるため、レコード長は 39 バイトになります。

NULL インジケータの値は、0x00 かそれ以外で判断されます。

通常 0x00 が非ヌル値を表し、0x01 がヌル値を表します。

◆LONGVARCHAR 型、NLONGVARCHAR 型、LONGVARBINARY 型のデータアクセス

LONGVARCHAR 型、NLONGVARCHAR 型、LONGVARBINARY 型のフィールドが含まれるテーブルを Btrieve API でアクセスする場合、他のフィールドと異なりレコードの構造に注意して使用する必要があります。これは、通常のレコードレイアウトとは異なる構造でレコードが構成されるためです。

LONGVARCHAR 型を例に説明しますが、NLONGVARCHAR 型、LONGVARBINARY 型も同様です。

列名	データ型	サイズ	ヌル	大小文字無視
RECNO	IDENTITY		<input type="checkbox"/>	<input type="checkbox"/>
USERID	CHAR	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>
COP	CHAR	30	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NAME	CHAR	16	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DATEQ	CHAR	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Q	LONGVARCHAR		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DATEA	CHAR	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
A	LONGVARCHAR		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

(図1)

例えば、図 1 のような構造のテーブルは実際どのように格納されているかを示したものが図 2 になります。これは実際のレコードの内容を 16 進ダンプ形式で表示したものです。COP、NAME、DATEQ、Q、DATEA、A はヌル値を許可するフィールドなので、各フィールドの先頭に NULL インジケータが置かれます。(黒枠で囲んだ部分) Q、A は LONGVARCHAR 型のフィールドで、他のフィールドとは異なりポインタと実体で構成され、実体は異なる場所に配置されます。

Btrieve では、通常固定長のレコードを使用しますが、longvarchar 型のデータは可変長のデータとなるため、実体は別の場所に保存し、実体を指し示すポインタのみを、固定長部分に持っています。

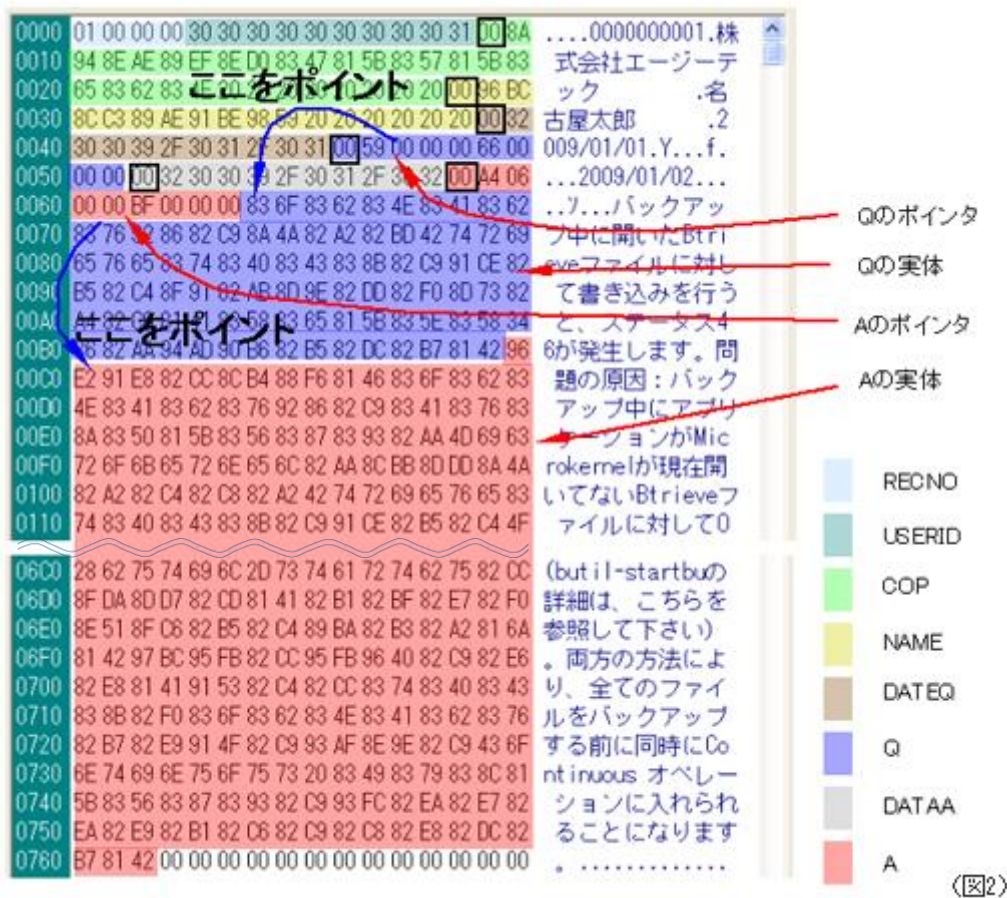


図 2 では、0065h までが固定長部分（固定部）で、それに続くのが可変長部分（可変部）になり、LONGVARCHAR 型フィールドの実体が格納されます。ポインタは 8 バイトで構成され、前半 4 バイトが実体の長さで、後半の 4 バイトがオフセットになります。オフセットは、固定部の先頭から何バイト目かをあらわします。NULL 値を許可する場合、8 バイトのポインタの前に NULL インジケータが置かれます。

LONGVARCHAR 型フィールドを書き込む場合、一度に書き込める長さは固定部、可変部を合わせて 64512 バイトまでに制限されます。この長さを超えるレコードは、Update Chunk オペレーションを使用して追加する必要があります。もちろん Update Chunk オペレーションで追加できるデータも制限されますから、長いデータは何回かに分けて Update Chunk オペレーションを繰り返す必要があります。

お勧めは、固定部を書き込んだ後、LONGVARCHAR 型フィールドを個別に Update Chunk オペレーションで書き込む方法です。64512 バイトに収まる場合、レコード長はデータバッファレングスに指定した長さとなることも注意が必要です。これは、固定部に

LONGVARCHAR 型のフィールドの長さを合計した長さとは必ずしも一致しないことを意味します。実際には、書き込みを行う際、固定部に LONGVARCHAR 型のフィールドの長さを合計した長さとデータバッファレングスは同じにしてください。

読み込みを行う場合には、各レコードのレコード長を事前に知ることが出来ないことに留意ください。データバッファレングスが小さければエラー22 が返ります。このため、バッファを大きく取るか、Get Direct/Chunk オペレーションで続きを読み込む必要があります。お勧めは、固定部をまずは読み込み、LONGVARCHAR 型フィールドを個別に Get Direct/Chunk オペレーションで読み込む方法です。これならば、余分なバッファを用意する必要がありません。書き込みを行う場合と同様、LONGVARCHAR 型データの長さが 64512 バイトを超える場合、64512 バイト以下に 分割して読み込むため、Get Direct/Chunk オペレーションを複数回実行する必要があります。(実際には、Get Direct/Chunk オペレーションのためのオーバーヘッドが必要なので 64512 バイトより少なくなります。)

LONGVARCHAR 型では、NULL 値の場合、NULL 値であるかを識別するフィールドに 0x01 を設定し、ポインタの 8 バイトは全て 0x00 を設定してください。