



Technical Note 05-22

セットを効果的に使用する-その 2

By Kent Wilbur, Manager Information Systems, 4D, Inc.
Technical Note 05-21

(原題: Clusters: Using Saved Sets Effectively - Part II)

概要

サンプルデータベースを使用して、クラスター（セット）の効果を検証します。title および textarea フィールドを対象に、"または"、"かつ"検索を行なう方法を紹介します。例外リストに含まれない単語は、すべて[Words]テーブルに登録されており、キーワードとして利用することができます。キーワードに対応するレコードの集合を示すクラスター（ブール配列）は、[Words]テーブルの BLOB フィールドに納められています。

検証: クラスター検索の実行速度

サンプルデータベースを起動し、ファイルメニューから TextBlocks を選択すると、[TextBlocks]テーブルの全レコードが表示されますので、これを確認したら Demo メニューから Cluster Demo を選択してください。最初のレコードの title フィールドは"Upgrading to 4D Server 6.5.1"です。本文中、"upgrading and converting"という表現がありますが、これはデータベース中、このレコードにユニークな表現です。

このレコードはキャッシュされてしまったので、クエリのテストをする前に 4D を終了しましょう。以下は私の環境で"upgrading and converting"を検索した場合のテスト結果です。

手段	時間/単位ミリ秒	レコード数
クエリ"含む"/キャッシュなし	1219	1
クエリ"含む"/キャッシュあり	1097	1
クラスター"または"	3	211
クラスター"かつ"	3	7
クラスター"含む"	4	1

クラスターによるサーチはキャッシュの有無に関係なく一定です。このサンプルは 6000 レコードほどの規模ですが、データベースが大きくなると、"含む"クエリは相当の時間がかかります。それに対して、クラスターによる検索は、実質的に所要時間が不変です。上記のテストでは、99.67%の時間短縮を達成することができています。

コードを幾らか変更する手間と、大容量のディスクドライブをさえ惜しまなければ、クラスターの使用によってこれだけ著しいスピードアップを実現することができるということです。

補足：上記のデータは数回のテストを実行した結果の平均値です。マシンによって多少の違いはありますが、ごく一般的な結果であると思います。

コードの説明

クラスター検索のコードでは、まず受け取ったワード（複数）の配列を作り、これを例外ワードリストと比較して、キーワードの配列を作っています。この配列の各要素については、[Words]テーブルに対応するレコードがあり、そのレコードの BLOB フィールドからはブール配列が、ブール配列からはセットが作られます。

セットが作られると、"かつ"、"または"、"含む"のうち、どの条件を適用するかによって、UNION か INTERSECTION コマンドのいずれかが使用されます。たとえば、クラスター"含む"の場合、INTERSECTION の結果（"かつ"の結果）に対して"含む"クエリを実行します。上記の例だと、7レコードまで絞られたセレクションを対象にクエリを実行したので、通常よりもずっと（99%以上）速く結果が得られました。

以下はクラスター検索ダイアログのメソッド M_ClusterQuery です。

```

▼ If (False)
    ` Method: M_ClusterQuery
    ` 4D Summit 2003
    ` Created by: Kent Wilbur
    ` Date: 8/26/2003

    ` Purpose: Demonstrated the Cluster Query method

    <>f_Version2003x1:=True
    <>fk_Wilbur:=True

End if

    ` Declare local variables
    C_LONGINT($EndTime)
    C_LONGINT($LPid)
    C_LONGINT($LStartTime)
    C_POINTER($pTable)

    $LWindowID:=Open form window([zDialogs];"QueryDemo";Movable dialog box ;Horizontally Centered ;Vertically Centered ;*)
    DIALOG([zDialogs];"QueryDemo")
    CLOSE WINDOW($LWindowID)

▼ If (OK=1)
    $LStartTime:=Milliseconds
    ▼ Case of
        ▼ ¥ (Length(tQueryText)=0)
            ALERT("You must enter a value to query.")
        ▼ ¥ (rb1=1) ` Old way of doing a contains search
            QUERY([TextBlocks];[TextBlocks]Title="@"+tQueryText+"@";*)
            QUERY([TextBlocks];1 ;[TextBlocks]TextArea="@"+tQueryText+"@")
        ▼ Else
            $pTable:=>[TextBlocks] ` This is the only table for this database but others could be handled here
            ▼ Case of
                ▼ ¥ (rb3=1) ` New method of doing a contains search
                    CLUSTER_DoQuery($pTable;tQueryText;"Contains")
                ▼ ¥ (sb1=1) ` Do AND query
                    CLUSTER_DoQuery($pTable;tQueryText;"And")
                ▼ Else
                    CLUSTER_DoQuery($pTable;tQueryText;"OR")
            End case
            ▼ Case of
                ▼ ¥ (Size of array(atQueryValues)=0)
                    ALERT("Sorry the value entered contains only non-indexed words¥rPlease try another phrase" + ".")
                ▼ ¥ (Records in selection($pTable->)=0)
                    ALERT("No records found.")
            End case
        End case
        $LEndTime:=Milliseconds
        ALERT("Query took "+String($LEndTime-$LStartTime)+" Milliseconds and found "+String(Records in selection([TextBlocks]))+" records.")
        WIN_OutputWindowTitle
    End if

```

次の Cluster_DoQuery メソッドは、もともと複数のテーブルを想定して作成したものです。Case 文の条件を増やせば、複数のテーブルに対応することができます。その場合は、[Words]テーブルにクラスターを使用するテーブルの数だけ BLOB フィールドを追加する必要があります。セットはテーブルに対してひとつ、と決まっているからです。4D Partner Central の場合、メインテーブルが3つあるので、[Words]テーブルにも BLOB フィールドが3つあります。次にコールする Cluster_ProcessWordFinds には、True=AND というように、クエリ条件をブール変数で渡します。

```

▼ If (False)
    ` Method: CLUSTER_DoQuery(ptr;text;text)
    ` 4D Summit 2003
    ` Created by: Kent Wilbur
    ` Date: 8/26/2003

    ` Purpose: Does actual Cluster Query

    <>f_Version2003x1:=True
    <>fk_Wilbur:=True

End if

    ` Declare parameters
    C_POINTER($1;$pTable)
    C_TEXT($2;$tQueryText)
    C_TEXT($3;$tQueryType)

    ` Reassign for readability
    $pTable:=$1
    $tQueryText:=$2
    $tQueryType:=$3

    CLUSTER_Text2Array($tQueryText;->atQueryValues)

▼ Case of
    ▼ ¥ (Size of array(atQueryValues)=0) ` Nothing to do, nothing to search
    ▼ ¥ ($pTable=(->[TextBlocks])) ` Do a keyword AND search
        ▼ Case of
            ▼ ¥ ($tQueryType#"Contains") ` Do a keyword AND/OR search
                CLUSTER_ProcessWordFinds($pTable;->[Words]TextBlockSet;->atQueryValues;($tQueryType="And"))
            ▼ Else ` Method of doing a contains search by using clusters
                CLUSTER_ProcessWordFinds($pTable;->[Words]TextBlockSet;->atQueryValues;True) ` Do an AND search
                QUERY SELECTION($pTable->[TextBlocks]Title=@"+"$tQueryText+"@";*) ` The query what remains
                QUERY SELECTION($pTable->[TextBlocks]TextArea=@"+"$tQueryText+"@")
            End case
        End case
    End case

```

Cluster_ProcessWordFinds メソッドは、クラスターの示すレコード群をセクションに展開します。該当するレコードがない場合、セクションをレコードなしにします。処理するキーワードがひとつだけの場合、UNION、INTERSECTION の処理は飛ばして直接、USE SET を実行します。

▼ **If (False)**

` Method: CLUSTER_ProcessWordFinds(ptr;ptr;ptr;bool)
` 4D Summit 2003
` Created by: Kent Wilbur
` Date: 8/26/2003

` Purpose: Processes the array of words to query

` \$1 = file pointer
` \$2 = pointer to blob field containing boolean
` \$3 = pointer to array of words to match
` \$4 = AND vs OR indicator True = AND

<>f_Version2003x1:=True

<>fk_Wilbur:=True

End if

` Declare parameters

C_POINTER(\$1;\$pTable)

C_POINTER(\$2;\$pBLOBField)

C_POINTER(\$3;\$pArray)

C_BOOLEAN(\$4;\$fAndQuery)

` Declare local variables

C_LONGINT(\$LSizeOfArray)

` Reassign for readability

\$pTable:=\$1

\$pBLOBField:=\$2

\$pArray:=\$3

\$fAndQuery:=\$4

\$LSizeOfArray:=Size of array(\$pArray->)

▼ **Case of**

▼ ¥ (\$LSizeOfArray=0)

REDUCE SELECTION(\$pTable->;0)

▼ ¥ (\$LSizeOfArray=1)

QUERY([Words];[Words]Word=\$pArray->{1})

CLUSTER_LoadFromBLOB(\$pTable;\$pBLOBField)

▼ Else

QUERY([Words];[Words]Word=\$pArray->{1})

CLUSTER_LoadFromBLOB(\$pTable;\$pBLOBField;"TempSet")

NEXT RECORD([Words])

▼ For (\$i;2;\$LSizeOfArray)

QUERY([Words];[Words]Word=\$pArray->{\$i})

CLUSTER_LoadFromBLOB(\$pTable;\$pBLOBField;"TempSet2")

▼ If (\$fAndQuery)

INTERSECTION("TempSet";"TempSet2";"TempSet")

▼ Else

UNION("TempSet";"TempSet2";"TempSet")

End if

NEXT RECORD([Words])

End for

USE SET("TempSet")

CLEAR SET("TempSet")

CLEAR SET("TempSet2")

End case

Cluster_LoadFromBLOB メソッドは、BLOB フィールドから取り出したブール配列からセットを作ります (CREATE SET FROM ARRAY)。キーワードがひとつだけの場合、セットは必要ないので、直接、セレクションを作ります (CREATE SELECTION FROM ARRAY)。

```
▼ If (False)
    ` Method: CLUSTER_LoadFromBLOB(ptr;ptr{;str})
    ` 4D Summit 2003
    ` Created by: Kent Wilbur
    ` Date: 8/26/2003

    ` Purpose: Loads the blob into a set

    ` $1 = table pointer
    ` $2 = pointer to blob field containing boolean
    ` $3 = set name, if length = 0 then just make current selection

    <>f_Version2003x1:=True
    <>fk_Wilbur:=True

End if

    ` Declare parameters
    C_POINTER( $1;$pTable)
    C_POINTER( $2;$pBLOBField)
    C_STRING(31;$3;$tSetName)

    ` Declare local variables
    ARRAY BOOLEAN($afBoolean;0)
    $pTable:=$1
    $pBLOBField:=$2

    ` Reassign for readability
    $tSetName:=""
▼ If (Count parameters>2)
    $tSetName:=$3
End if

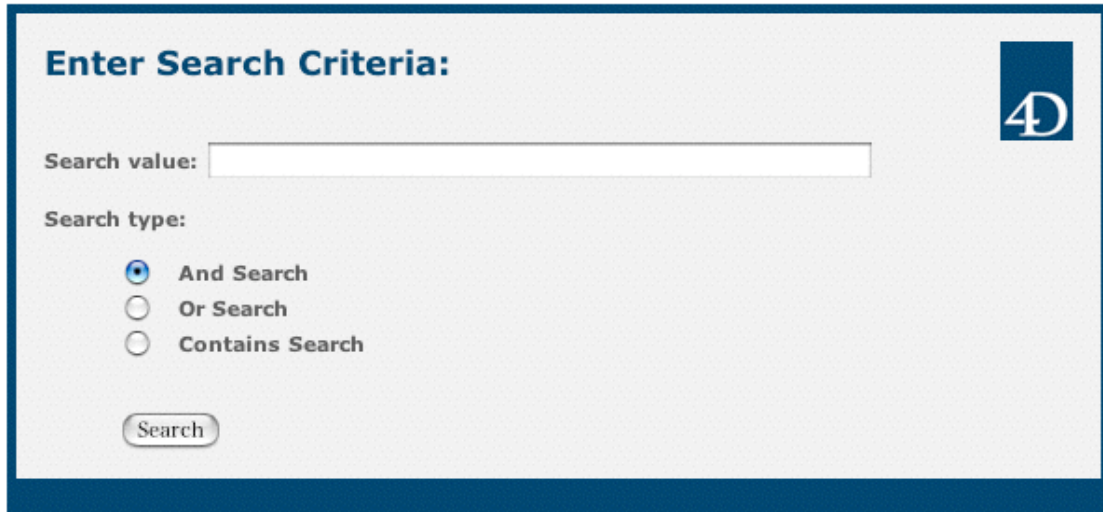
    BLOB TO VARIABLE($pBLOBField->,$afBoolean) ` Get the boolean array out of the record

▼ If (Length($tSetName)=0) ` We are not making a set but a current selection
    ARRAY BOOLEAN($afBoolean;Size of array($afBoolean)+1)
    CREATE SELECTION FROM ARRAY($pTable->,$afBoolean; "")
▼ Else
    CREATE SET FROM ARRAY($pTable->,$afBoolean;$tSetName)
End if

    ` End of method
```

Web インタフェース

Web 経由でクラスター検索を可能にするために必要なインタフェースは、それほど複雑ではありません。メソッドと HTML ページがそれぞれ数件、必要なだけです。サンプルデータベースのデフォルトページ index.html も、簡単なフォームです。ブラウザに `http://127.0.0.1` と入力すると次の画面が表示されます。



Cluster_DoQuery メソッドでは、クエリ方法を予約された語句 "And"、"Or"、"Contains" で判別しています。それで、フォームのラジオボタンは、3つの語句のうちどれかを送信するようにしています。

```
<input type="radio" name="tQueryType" value="And" checked> <strong> And Search</strong><br>
<input type="radio" name="tQueryType" value="Or"> <strong> Or Search</strong><br>
<input type="radio" name="tQueryType" value="Contains"> <strong> Contains Search</strong><br>
```

送信ボタンは、4D ACTION を使用してメソッド WEB_ClusterQuery を呼び出しています。4D に組み込まれた Web Server は、変数をマッピングしてくれますが、これが正しく機能するためには、COMPILER_WEB メソッドで Web から 4D に渡される変数の型を宣言しておく必要があります。

```

C_TEXT(tQueryText)
C_TEXT(tQueryType)
C_LONGINT(LRecordNumber)
C_TEXT(tCGI)

```

```

tQueryText:=""
tQueryType:=""
LRecordNumber:=0

```

COMPILER_WEB で変数が宣言されていれば、4D は Web フォームから受信した変数を対応する 4D の変数に割り当てます。このように Web 経由で変数をやりとりする場合、COMPILER_WEB メソッドは必須です。

WEB_ClusterQuery メソッドは、基本的に M_ClusterQuery メソッドと同じ処理を実行しています。

```

▼ If (False)
    ` Method: WEB_ClusterQuery
    ` 4D Summit 2003
    ` Created by: Kent Wilbur
    ` Date: 9/2/2003

    ` Purpose: Demonstrated the Cluster Query method via the web

    <>f_Version2003x1:=True
    <>fk_Wilbur:=True

End if

CLUSTER_DoQuery(->[TextBlocks];tQueryText;tQueryType)
ARRAY LONGINT(aLRecordNumber;0)
ARRAY TEXT(atTitle;0)
SELECTION TO ARRAY([TextBlocks];aLRecordNumber;[TextBlocks]Title;atTitle)
SEND HTML FILE("DisplayRecords.shtml")
`End of method

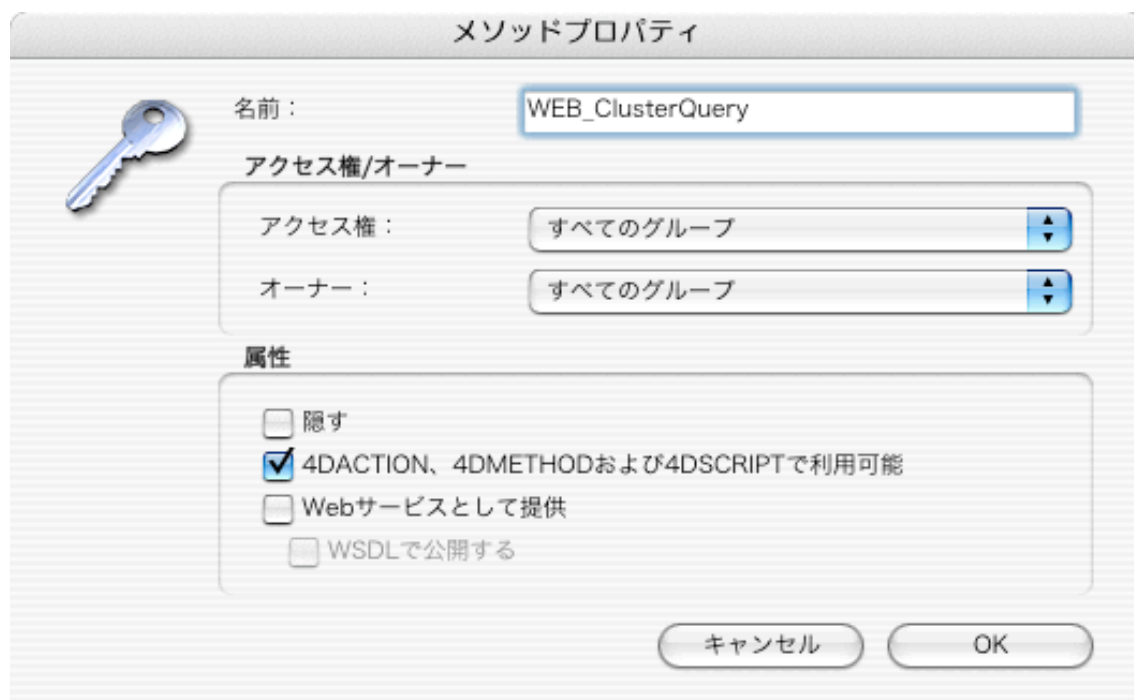
```

ループを使用してレコードを Web に出力する方法はふたつあります。つまり、レコードそのものを使用するか、配列を使用するかのどちらかです。項目の一覧の中からひとつを選択すると、詳細な内容が閲覧できるようなインタフェースを作成したいとします。そのためには、レコードを特定するために値のユニ

ークなフィールドが必要ですが、今回のサンプルでは、物事を簡単にとどめるためにそのようなキーフィールドをあらかじめ設けませんでした。この場合、配列を使用してレコード番号でループするほうが楽です。レコード番号と件名の配列を作成するわけですが、このとき留意しなければならないのは、Web に配列を出力する場合にローカル配列は使用できないという点です。

送信される HTML ページには.shtml 拡張子が付いています。この拡張子がついていると、4D は HTML を解析して処理すべき 4dvar タグを探そうとします。この拡張子がついていないと、ページがスタティックページとして扱われてしまい、そのまま送られてしまう恐れがあります。

4DACTION で直接メソッドを呼び出すことにしたので、メソッドのプロパティを確認して 4DACTION で利用可能のチェックボックスを有効にしておきます。セキュリティのため、デフォルトでこのプロパティはオフになっていますが、2003 よりも前から変換されたデータベースの場合は、オンになっている可能性が高いので、注意が必要です。



メソッドプロパティ

名前: WEB_ClusterQuery

アクセス権/オーナー

アクセス権: すべてのグループ

オーナー: すべてのグループ

属性

☐ 隠す

☒ 4DACTION、4DMETHODおよび4DSCRIPTで利用可能

☐ Webサービスとして提供

☐ WSDLで公開する

キャンセル OK

DisplayRecords.shtml ページ

このサンプルデータベースで、面倒な処理を行なっている箇所をあえて挙げるとしたら、このページということになります。検索結果を出力し、詳細を表示するためのリンクを出力することに加え、想定される入力エラーを適切に処理する必要があります。

```

<body>
<div id="wrapper">
<!--#4dif (Size of array(atTitle)=0)-->
<p><!--#4dif (Length(tQueryText)=0)-->You must enter something to search if you expect to find anything!
<!--#4delse--><!--#4dif (Size of array(atQueryValues)=0)-->Sorry the value entered contains only non-indexed
words. Please try another phrase.
<!--#4delse-->Sorry, no records were found matching your request. Please try different words.<!--#4dendif--><!--
#4dendif--><br><br><br><br></p>
<!--#4delse-->
<!--#4DLoop atTitle-->
<h1>Title:<a href="<!--#4dvar tCGI-->/4daction/Web_ShowRecord?LRecordNumber=<!--#4dvar
aLRecordNumber(atTitle)-->"><!--#4DVAR atTitle(atTitle)--></a></h1>
<!--#4DEndLoop-->
<!--#4dendif-->
</div>
<a href="http://www.4d.com/" id="logo"></a>
</body>

```

想定されるエラーは、条件が何も入力されなかった場合、語句がすべて例外語句だった場合、レコードがひとつもみつからなかった場合の 3 種類です。相互に排他的な条件であるとはいえ、HTML には case 文に相当するようなシンタックスはないので、一連の if 文を使用してこの処理を行なっています。4D タグは <!--something--> のような html コメントの形で挿入します。# は省略しても機能するかもしれませんが、HTML エディタで正しく解釈されるように、必ず # を含めることが勧められています。

出力するべきレコードがある場合は、#4dloop が発動します。このタグ <!--#4DLoop atTitle--> には、少なくともひとつの要素をもつ配列か、少なくともセレクションにひとつのレコードをもつテーブルを渡します。

ループでは、件名と詳細ページへのリンク <a href> がその都度、作られます。

```

<!--#4dvar tCGI-->/4daction/Web_ShowRecord?LRecordNumber=<!--#4dvar aLRecordNumber(atTitle)-->

```

この場合、tCGI には値がないので、リンクはブラウザに表示されている URL に対して相対的なものとなります。

結果を HTML で出力する

DisplayRecords.shtml ページでユーザがリンクをクリックすると、WEB_ShowRecord メソッドが呼び出されます。先のメソッドと同じように、プロパティでコールできるようにしておきます。

メソッドはレコードを見つけ出すと、ShowRecord.shtml ページを送信します。このページはロードされたカレントレコードの内容を表示するだけの単純なものです。

```

<body>
<div id="wrapper">
<h1><strong>Summary:</strong> <!--#4DVAR [TextBlocks]Title --></h1>
<p><strong>Description:</strong> <!--#4DHTMLVAR [TextBlocks]TextArea --></p>
</div>
<a href="http://www.4d.com/" id="logo"></a>
</body>

```

Summary: Condition Statements in 4D Business Kit



Description: When you want to use conditional HTML code, place that code within a 4DBKIF - 4DBKElse - 4DBKEndIF statement. Only when the condition of the 4DBKIF command evaluates to true, will the code between it and the next 4DBKElse or 4DBKEndIF command be processed and displayed. If the condition is false, the code between the 4DBKElse and 4DBKEndIF will be processed and displayed. This is applicable for both 4DBK tags/commands as well as HTML code. Note: The same can be applied to the 4DBKCaseOF - 4DBKCase - 4DBKEndCaseOF condition statement tag/commands.

まとめ

クラスターを効果的に活用すれば、頻繁に発生するクエリの実行速度を大きく改善することができます。データベースを最適化する方法には、コンパイルなど様々な手段があるとはいえ、もしクラスターの使用が現実的ならば、これに代わるスピードアップはないといってもよいでしょう。