

TNT 入門



QUERIE

問い合わせの作成と使用



with

TNTmips®

TNTedit™

はじめに

本書では、TNT 製品の中でデータベース問い合わせを使用する方法を紹介し、さらにユーザのベクタ、CAD、TIN オブジェクトに対応付けられた属性情報を利用するための問い合わせを作成する方法を示します。問い合わせ処理は最初は複雑に見えるかもしれませんが、本書では、1行からなる簡単な例から複数の条件命令や処理ループを含む問い合わせまで一連の練習問題を実行しながら、問い合わせで要求される構造やシンタックスを段階を追って説明して行きます。

必須基礎知識 本書では、読者が『TNT入門：地理空間データ表示』、『TNT入門：システムの基本操作』の練習問題を完了しているものと仮定しています。基本的な概念や操作についてはこれらの練習問題で説明されており、本書では繰り返して説明しませんので、必要に応じてこれらのマニュアルやTNTmips リファレンス・マニュアルで調べてください。

サンプルデータ 本書の練習問題では、TNT製品に添付されているサンプルデータを使用します。TNT製品のCDにアクセスできない場合は、マイクロイメージ社のウェブサイトからデータをダウンロードできます。特に、本書ではQUERYデータ・コレクションのサンプル・ファイルを使用します。これらのファイルのオブジェクトを使用する際に保存できるよう、ハードディスク・ドライブ上にサンプルデータのコピーがあることを確認してください。

その他の資料 本書では、データベース問い合わせの使用方法に関する概要しか示されておりません。問い合わせエディタに関する詳細は、TNTmips リファレンス・マニュアルの表示の巻の「問い合わせによるベクタ、CAD、TINの表示」の章を参照してください。また、問い合わせコマンドと関数に関する詳細は、付録2:「データベース問い合わせ」を参照してください。

TNTmips® と TNTlite® TNTmipsには2つのバージョンがあります。プロフェッショナル・バージョンと、無料バージョンであるTNTliteです。本書では、どちらのバージョンも「TNTmips」と呼ぶことにします。プロフェッショナル・バージョンにはソフトウェアライセンス・キーが必要です。このキーがない場合、TNTmipsはTNTliteモードで動作し、オブジェクトのサイズが制約されるほか、TNTliteの別のコピーとの間でしかデータを共有できません。

データベース問い合わせを使用して、TNTviewにおける地理空間オブジェクトの表示を制御したり、TNTeditで編集する要素を選択することもできます。TNTliteでは、添付されたサンプルの地理データを使用して本書のすべての練習問題を完全に実行することができます。

Randall B. Smith 博士, 2001年9月17日

本書の一部のイラストでは、カラー・コピーでないと重要な点がわかりにくい場合があります。マイクロイメージ社のウェブ・サイトから本書を入手されれば、カラーで印刷したり表示できます。また、このウェブ・サイトからは、『TNT入門』のその他のテーマに関する最新のパンフレットも入手できます。インストール・ガイド、サンプル・データ、および最新バージョンのTNTliteをダウンロードできます。アクセス先は次の通りです。

<http://www.microimages.com>

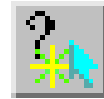
問い合わせの作成と使用の世界ようこそ

TNTmipsでは、ベクタ、CAD、TINオブジェクトのデータベース属性を使用することで、きわめて柔軟にオブジェクトの表示や印刷を制御したり、さまざまな処理で使用する要素を選択することができます。この属性情報を利用する上で最も完全で汎用的な手段が、データベース問い合わせです。

データベース問い合わせは、データベースからのレコードの選択に使用する属性の基準を定義した一連の命令です。データベース問い合わせを行うと、これらのレコードが添付された特定の(線やポリゴンなどのような)空間要素が自動的に選択され現在の処理に使用されます。問い合わせは特定の要素タイプに適用され、ユーザは1つのオブジェクト内の異なる要素タイプ群に対して別々の問い合わせを同時に使用することができます。Spatial Data Display(空間データ表示)の中でStyle By Script(スクリプトによるスタイル設定)オプションを使用すると、属性に基づいて選択された要素用の表示パラメータを指定することができます。問い合わせの中でユーザが参照する属性情報は、質的なもの(クラス名など)であっても量的なもの(作物の産出高など)であっても構いません。

問い合わせでは、TNTmipsが解釈できる標準的な「文法とシンタックス」を使用しなければなりません。使用される問い合わせ言語は、TNTmips Spatial Manipulation Language (TNTmips 空間操作言語: SML) のサブセットとなっています。問い合わせの作成は、Query Editor(問い合わせエディタ)ウィンドウで行います。このウィンドウにはメニューがあり、使用可能なデータベース・テーブルからフィールドを選択したり、正しいシンタックスのアウトラインを示すリスト・ウィンドウから記号や関数を挿入できるため、有効な問い合わせを簡単に作成することができます。Query Editor(問い合わせエディタ)には、問い合わせを適用する前に間違いを発見するのに役立つシンタックス・チェッカも用意されています。

本書の練習問題では、表示するベクタ・オブジェクト内で要素を選択したりスタイルを設定するために問い合わせを使用します。しかし、問い合わせは、ベクタ、CAD、TINオブジェクト内の構成要素を選択する任意の処理で使用できるということを覚えておいてください。このほか、一部のラスタ処理では問い合わせを使用して、処理するセル値を選択することもできます。



ステップ

- ハードディスク・ドライブ上に **QUERY** データ集のサンプルデータのコピーがあることを確認してください。
- TNTmips を起動します。
- Spatial Data Display(空間データ表示) 処理を起動し、ツールバーから New 2D Group(新規二次元グループ) を選択します。

4～9ページの練習問題では、単純な問い合わせ命令の構造、比較演算子、および問い合わせの作成やチェックに役立ついくつかのツールを紹介します。10～17ページでは、複合命令を含む問い合わせの構造、変数やコメントの使用法、および問い合わせを使用してデータベース・レコード添付をチェックする方法を説明します。18～19ページでは Style By Script(スクリプトによるスタイル設定)、および「if-then-else」による条件付き構造の使用法を説明します。20～23ページでは、ベクタ・オブジェクトの空間的、トポロジ的な属性に基づく問い合わせの例を示します。24～25ページでは、データベース・テーブル内に計算フィールドを作成するためのスクリプトを示します。26～27ページでは、ベクタ・オブジェクトの編集に役立つ問い合わせの例を示します。

1つのフィールドを問い合わせで選択する

ステップ

- Add Vector (ベクタを追加) アイコン・ボタンをクリックし、ドロップダウン・メニューから Add Vector Layer (ベクタ・レイヤーを追加) を選択します。
- QUERYデータ集のCB_SOILQプロジェクト・ファイルからCBSOILS_LITEオブジェクトを選択します。
- Vector Object Display Controls(ベクタ・オブジェクト表示コントロール)ウィンドウで、Lines(線)のタブ付きパネルでSelect(選択)オプションがAll(すべて)に設定されているかチェックします。
- Labels(ラベル)パネルのSelect(選択)オプションをNone(なし)に設定します。
- Polygons(ポリゴン)パネルのStyle(スタイル)オプションがBy Attribute(属性により)に設定されているかチェックします。



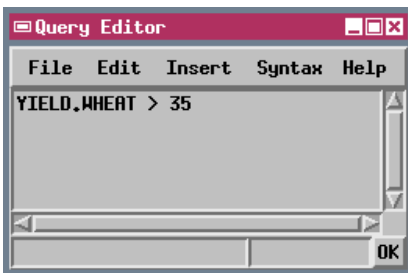
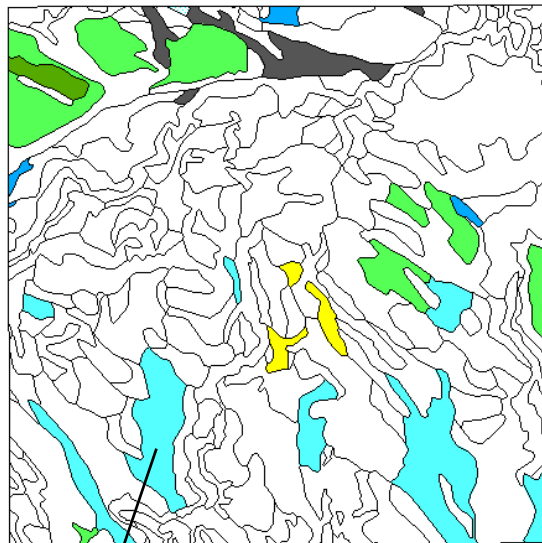
最も単純な形の問い合わせでは、1つのデータベース属性の値に基づいて特定のタイプの空間要素(ベクタ・オブジェクト内のポリゴン、線、または点など)を選択します。この練習問題では、単純な問い合わせを入力して、ベクタ・オブジェクトに含まれる土壤図ポリゴンを選択して表示します。各土壤タイプには、いくつかの作物に関する最大潜在産出高の値が対応付けられています。この問い合わせでは、小麦の潜在作物産出高が1エーカー当たり35ブッシェルより大きいポリゴンを選択します。問い合わせ命令は次のような形になります。

属性 比較演算子 値

問い合わせでは、属性情報が含まれているデータベース・テーブルとフィールドを指定する必要があります。「属性の場所」に関するこの情報は、TABLE.FIELDの形で入力します。この例では、値は単純な数値、比較演算子は「より大」演算子(>)となっています。



- Polygons(ポリゴン)パネルのSelect(選択)オプションをBy Query(問い合わせにより)に設定し、隣のSpecify(指定)ボタンをクリックします。
- Query Editor(問い合わせエディタ)ウィンドウで、次のテキストを(大文字小文字の区別も含め)正確に入力します。
YIELD.WHEAT > 35
- Query Editor(問い合わせエディタ)ウィンドウで[OK]をクリックします。



- Vector Object Display Controls(ベクタ・オブジェクト表示コントロール)ウィンドウで[OK]をクリックし、表示設定をそのまま確定すると、ベクタ・オブジェクトが表示されます。

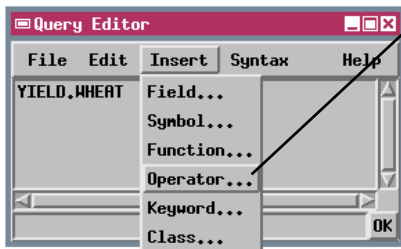
線の選択はAll(すべて)に設定されているため、すべての土壤ポリゴンの輪郭が描画されます。問い合わせによって選択されたポリゴンは、By Attribute(属性により)表示用にあらかじめ設定された、土壤タイプに従うカラーと塗りつぶしパターンにより塗りつぶされます。いくつかの土壤タイプが、小麦の産出高に関する選択基準に適合しています。選択されなかったポリゴンは塗りつぶされないままになっています。

Insert Operator (演算子を挿入) オプションを使用する

前の問い合わせでは、いくつかの土壌タイプ・クラスに属する土壌ポリゴンを選択しました。ここでは、問い合わせで1エーカー当たりの潜在小麦産出高が正確に38ブッシェルであるようなポリゴンだけが選択されるように、選択基準を微調整してみましょう。この選択基準を指定するには、問い合わせ命令の中で「Equal to (等しい)」演算子(==、二重等号)を使用します。Query Editor (問い合わせエディタ) の Insert / Operator (挿入 / 演算子) メニュー・オプションを使用すると Insert Operator (演算子を挿入) ウィンドウが開き、スクロール式のリストから演算子を選択して問い合わせ命令の中の現在のカーソル位置に挿入することができます。

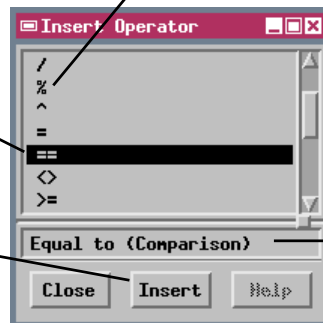
ステップ

- ✓ Layer (レイヤー) アイコンの行の Vector (ベクタ) アイコンをクリックし、Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウを開きます。
- ✓ Polygon Options (ポリゴン・オプション) パネルで Select: [Specify] (選択: [指定]) をクリックします。
- ✓ Query Editor (問い合わせエディタ) で "> 35" をハイライト表示にし、<Delete> (削除) を押します。
- ✓ Insert (挿入) メニューを開いて Operator (演算子) を選択します。
- ✓ Insert Operator (演算子を挿入) ウィンドウを下にスクロールして "==" 演算子を選択し、[Insert (挿入)]、[Close (閉じる)] の順にクリックします。
- ✓ Query Editor (問い合わせエディタ) ウィンドウで、演算子の右側に 38 と入力して [OK] をクリックします。
- ✓ Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで [OK] をクリックして表示設定をそのまま確定すると、ベクタ・オブジェクトが表示されます。



Insert (挿入) メニューから Operator (演算子) を選択して Insert Operator (演算子を挿入) ウィンドウを開くと...

... 使用可能な問い合わせ演算子のリストが表示されます。

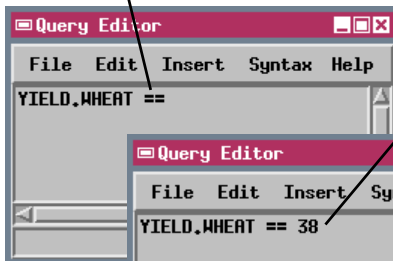


「Equal to (等しい)」演算子を選択します。

[Insert (挿入)] をクリックし...

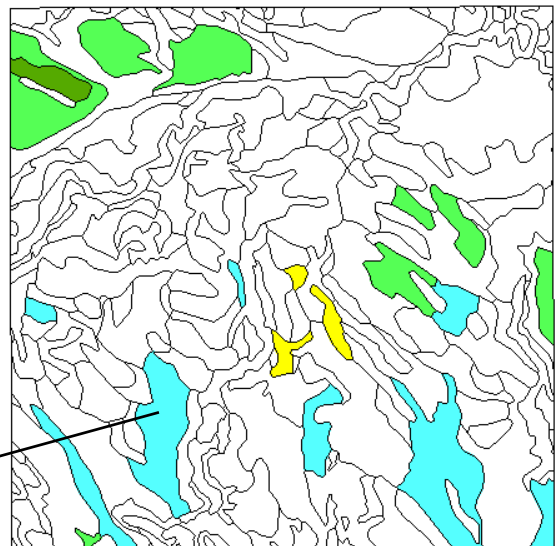
テキスト・フィールドには、選択された演算子の機能説明が表示されます。

... 問い合わせ命令に演算子を入力します。



一致させる値を入力します。

修正された問い合わせ命令では選択基準が厳しくなるため、適合する土壌クラスが少なくなっています。



Insert Field (フィールドを挿入) オプションを使用する

ステップ

- ✓ Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウと Query Editor (問い合わせエディタ) ウィンドウを開きます。
- ✓ 既存の問い合わせの中の YIELD.WHEAT を選択し、<Delete> (削除) を押します。
- ✓ Insert (挿入) メニューを開き、Field (フィールド) を選択します。
- ✓ Insert Field (フィールドを挿入) ウィンドウが開きますので、Table (テーブル) リストの中の YIELD をクリックします。
- ✓ Field (フィールド) リストの中の OATS をクリックし、[Insert (挿入)] をクリックします。
- ✓ Insert Field (フィールドを挿入) ウィンドウで [Close (閉じる)] をクリックします。
- ✓ 問い合わせ命令の右辺の値を 43 に変更して [OK] をクリックします。
- ✓ Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで [OK] をクリックします。

Query Editor (問い合わせエディタ) ウィンドウの Insert / Field (挿入 / フィールド) メニュー・オプションを使用すると、問い合わせの作成や修正にも役立ちます。このオプションは Insert Field (フィールドを挿入) ウィンドウを開きますので、Table (テーブル) や Field (フィールド) を選択し、自動的に属性位置情報を問い合わせ命令に正しい形で挿入することができます。

Insert (挿入) メニューから Field (フィールド) を選択して Insert Field (フィールドを挿入) ウィンドウを開くと ...

...選択された要素タイプに対して使用可能なすべてのデータベース・テーブルがリストされます。

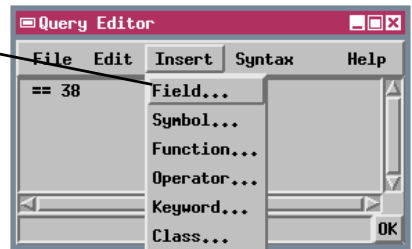
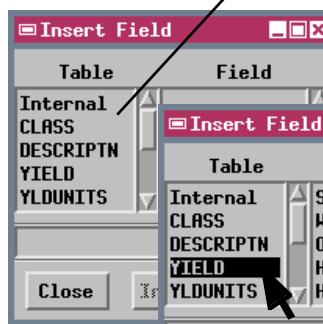
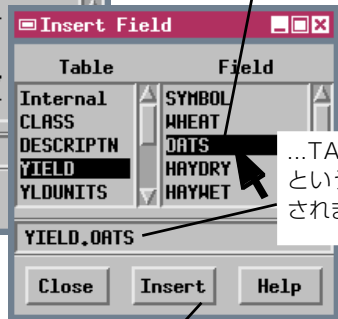


Table (テーブル) を選択しても、対応するフィールドのリストが表示されます。

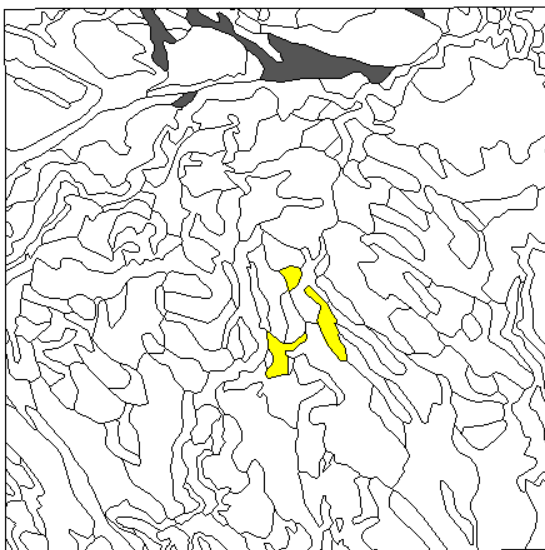


フィールドを選択すると ...



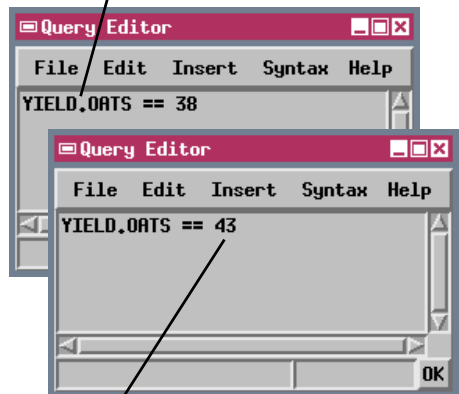
...TABLE.FIELD という項目が作成されます。

この問い合わせでは、(4 ページの) 最初の問い合わせで選択されたのと同じ土壌クラスのうちのいくつかを選択されます。



Insert (挿入) ボタンをクリックし ...

...TABLE.FIELD項目を問い合わせ命令に挿入します。



命令の右辺の値を 43 に変更します。

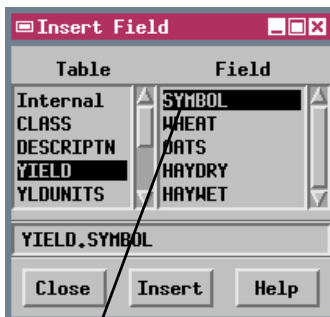
文字列フィールドを問い合わせる

TNTmipsで使用される問い合わせ言語では、大文字と小文字が区別されます。CLASSというテーブルにClassというフィールドが含まれている場合、TABLE.FIELD項目はCLASS.Classとしなければなりません。これを誤ってCLASS.CLASSと入力すると、問い合わせ処理はフィールドを見つけられず、問い合わせの中にエラーがあることを報告してきます。Insert Field (フィールドを挿入) 手順を使用すると、このような間違いを防ぐのに役立ちます。

ここまでの問い合わせで使用したデータベース・フィールドには、数値データが含まれていました。CBSOILS_LITEに対応するYIELDテーブルには、土壌タイプ記号がString (文字列) 形式になったSYMBOLという名前のフィールドも含まれています。「string」という用語は「character string (文字列)」のことであり、このフィールドは数値的には評価されず、テキストや数値でないその他の文字を含むことができます。文字列フィールドに数字 (CLASS1など) が含まれていることもありますが、これらは数値としてではなく文字として解釈されます。問い合わせの中の文字列値は二重引用符で囲む必要があります。また、大文字と小文字も区別されます。

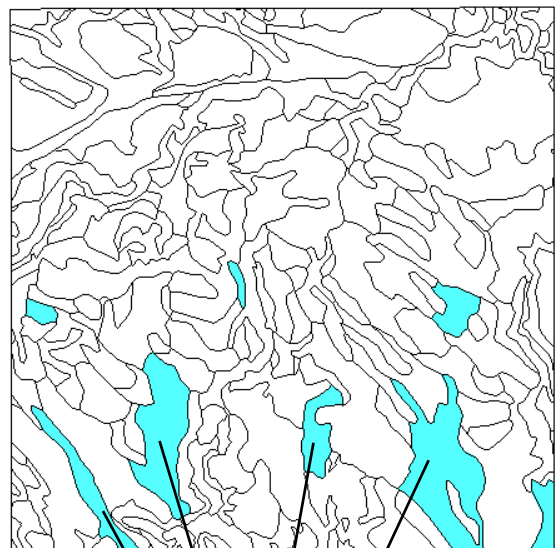
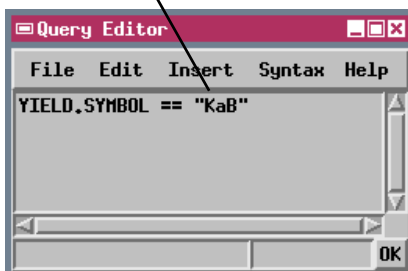
ステップ

- ✓ Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウとQuery Editor (問い合わせエディタ) ウィンドウを開きます。
- ✓ 既存の問い合わせの中のYIELD.OATSを選択して<Delete>(削除)を押します。
- ✓ Insert Field (フィールドを挿入) 手順を使用して問い合わせ命令の左辺にYIELD.SYMBOLを挿入します。
- ✓ 問い合わせ命令の右辺の値を"KaB"に変更し(二重引用符を忘れないでください)、[OK]をクリックします。
- ✓ Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで[OK]をクリックします。



SYMBOL フィールドには文字列値が含まれています。

文字列は二重引用符で囲みます。

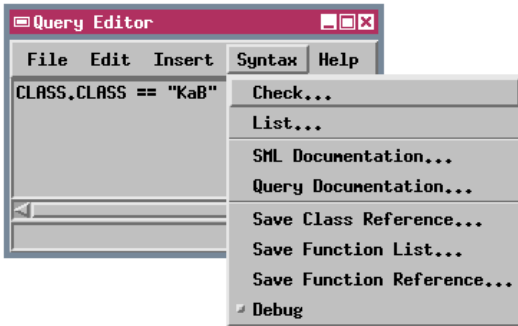


クラスKaBに属する、選択された土壌ポリゴン。

問い合わせのシンタックスをチェックする

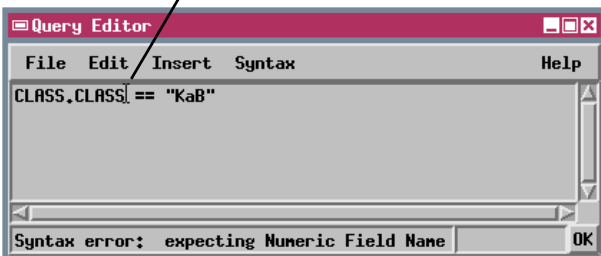
ステップ

- ✓ Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウと Query Editor (問い合わせエディタ) を開きます。
- ✓ 既存の問い合わせ命令の左辺を手動操作で CLASS.CLASS (すべて大文字) に変更します。



- ✓ Syntax (シンタックス) メニューから Check (チェック) を選択します。
- ✓ Message (メッセージ) ウィンドウのエラー・メッセージを確認して [OK] をクリックします。
- ✓ 問い合わせ命令の左辺を CLASS.Class に変更します。
- ✓ Syntax (シンタックス) メニューから Check (チェック) を選択します。
- ✓ 「Syntax OK」というメッセージを確認します。

この例では、カーソルは、無効な TABLE.FIELD 項目の最後にあります。

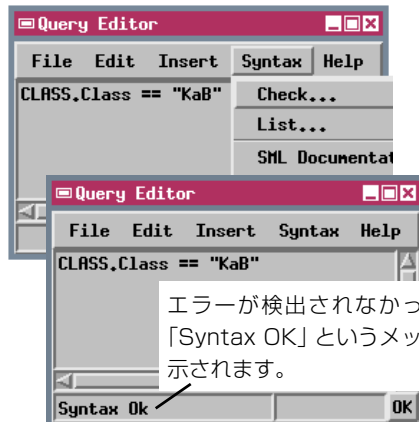


より複雑な問い合わせでは、カーソルの位置より後に、右カッコや、begin / endループの 'end' 命令など、何かが実際に不足しているためにシンタックス・エラーとなることもあります。

文字列における大文字と小文字の区別や引用符に関する規則は、TNTmips 問い合わせ言語のシンタックスの例です。問い合わせシンタックスは、ユーザが [OK] をクリックして問い合わせを実行するときや、File (ファイル) メニューから Close (閉じる) を選択して Query Editor (問い合わせエディタ) を閉じるときに、自動的にチェックされます。(Query Editor (問い合わせエディタ) を閉じようとしたときに、問い合わせの中にシンタックス・エラーがあった場合、ウィンドウは開いたままになりエラー・メッセージが表示されます。)

問い合わせの実行前にシンタックスをチェックするには、Query Editor (問い合わせエディタ) の Syntax (シンタックス) メニューから Check (チェック) を選択します。チェック処理では、スペルの間違い、カッコなどの記号の抜け、存在しないデータベース・フィールドの参照などを検出できます。チェック処理は、問い合わせの先頭からチェックしていきます。シンタックス・エラーが検出されなかった場合は、Query Editor (問

合わせエディタ) の最下部に「Syntax OK」というメッセージ行が表示されます。シンタックス・エラーが検出された場合は、コンピュータが解釈できなかった命令の第1要素の終わりにテキスト・カーソルが表示され、メッセージ行にはエラー・メッセージが表示されます。この例では、CLASS というテーブルの中に CLASS というデータベース・フィールドが存在しないことを問い合わせチェッカが検出するため、CLASS.CLASS 項目の終わりにカーソルが表示されます。シンタックス・エラーをすべて修正したら、もう一度 Syntax (シンタックス) オプションを使用して問い合わせの残りの部分にエラーがないかどうか調べることができます。

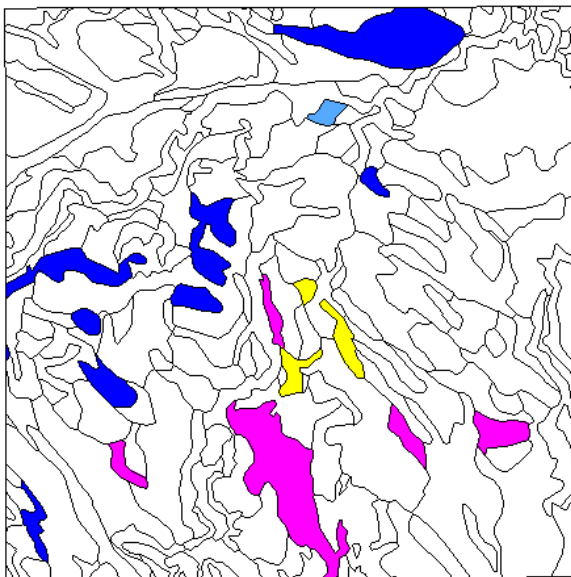


エラーが検出されなかった場合は、「Syntax OK」というメッセージが表示されます。

問い合わせの中で計算を使用する

問い合わせ命令の右辺の値を、データベース・フィールドから与えたり、データベース・フィールドに対する計算を行って与えたりすることもできます。問い合わせでの計算には、標準的な算術演算である加算(+)、減算(-)、乗算(*)、除算(/)を使用できます。必要ならば、Insert Operator (演算子を挿入) ウィンドウから演算子記号を入力できます。Insert / Function (挿入 / 関数) オプションを使用すると、問い合わせ命令の中で使用可能な三角関数やその他の数学的関数にアクセスできます。この練習問題の問い合わせ例は、1 エーカー当たりのオート麦の潜在産出高が小麦の産出高よりも正確に5 ブッシェル大きい土壌ポリゴンを選択します。

ここまでの説明でおそらく、特定のオブジェクトと要素タイプに使用された最後の問い合わせはオブジェクトとともに自動的に保存され、同じ By Query (問い合わせにより) オプションを次の回に選択した場合に開かれることに気がつかれたでしょう。後で使用するために同じオブジェクト用の複数の問い合わせを保存する場合は、Query Editor (問い合わせエディタ) の File (ファイル) メニューの Save (保存)、および Save As (名前を付けて保存) オプションを使用することができます。これらのオプションを使用すると、Query Editor (問い合わせエディタ) ウィンドウに現在表示されている問い合わせを、ファイル拡張子 .QRY の付いたテキスト・ファイル、またはプロジェクト・ファイル内のテキスト・オブジェクトとして保存することができます。保存された問い合わせを再度読み込む場合は File (ファイル) メニューの Open (開く) オプションを使用します。



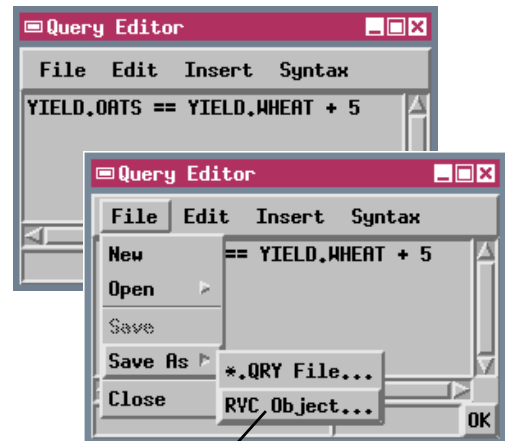
ステップ

- File (ファイル) メニューから New (新規) を選択します。
- Insert (挿入) 手順や手動入力を使用して、次に示す問い合わせ命令を作成します。

```
YIELD.OATS == YIELD.WHEAT + 5
```

- File (ファイル) メニューから Save As (名前を付けて保存) を選択するとメニューが開きますので、*.QRY File (*.QRY ファイル) を選択します。
- 標準の File Selection (ファイル選択) ウィンドウを使用して、問い合わせを保存する新しいファイルに名前を付けます。
- Query Editor (問い合わせエディタ) ウィンドウで [OK] をクリックします。
- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで [OK] をクリックします。

通常、問い合わせは、一意的なデータベース・フィールドを参照するため、特定のオブジェクトに対してしか使用できません。全く同じデータベース・フォーマットを持つ一連のオブジェクトがある場合や、問い合わせが TNTmips によって作成された標準テーブル内のフィールドしか参照しない場合は、任意のオブジェクトに対して同じ問い合わせを使用できます。



プロジェクト・ファイル内のテキスト・オブジェクトとして問い合わせを保存するには RVC オブジェクトを選択します。

複合問い合わせ

ステップ

- ✓ Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウと Query Editor (問い合わせエディタ) ウィンドウを開きます。
- ✓ File (ファイル) メニューから New (新規) を選択します。
- ✓ Insert (挿入) 手順や手動入力を使用して、次に示す問い合わせを作成します。

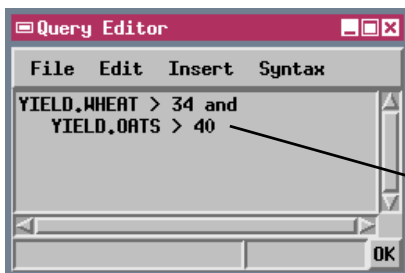
```
YIELD.WHEAT > 34 and YIELD.OATS > 40
```

- ✓ Query Editor (問い合わせエディタ) ウィンドウと Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで [OK] をクリックします。
- ✓ 上記のステップを繰り返します。ただし今回は問い合わせ命令の中の「and」を「or」に変えます。

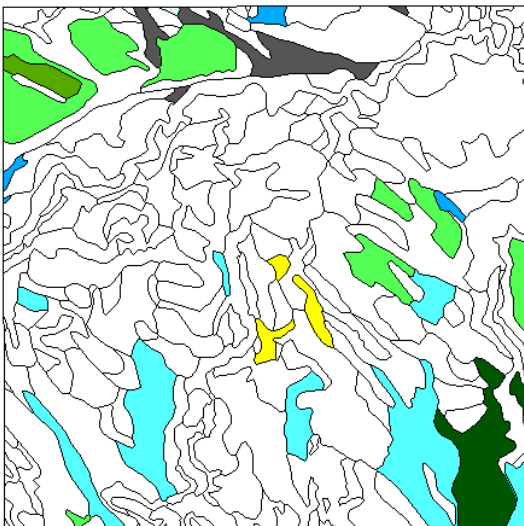
前の練習問題で使用されたそれぞれの問い合わせでは、表示するポリゴンを選択するために1つの選択用比較しか行っていません。多くの場合は、要素を選択する際にいくつかの基準を組み合わせることが必要となります。問い合わせ命令の中の一連の選択用比較式は、集合論の1つまたは複数の「and」、「or」、「not」などの論理演算子によって相互に関係付けられていなければなりません。これらの演算子はすべて小文字で入力されなければいけません。問い合わせ命令で論理演算子を挿入するには Insert Operator (演算子を挿入) 手順を使用します。

2つの比較式が論理演算子「and」でリンクされている場合は、両方の比較結果が真になると、問い合わせ命令全体が真になって要素を選択できます。2つの比較式が論理演算子「or」でリンクされている場合は、独立した個々の比較結果のいずれかが真になれば、問い合わせ命令が真になります。いずれかの基準を満足する要素が選択されます。

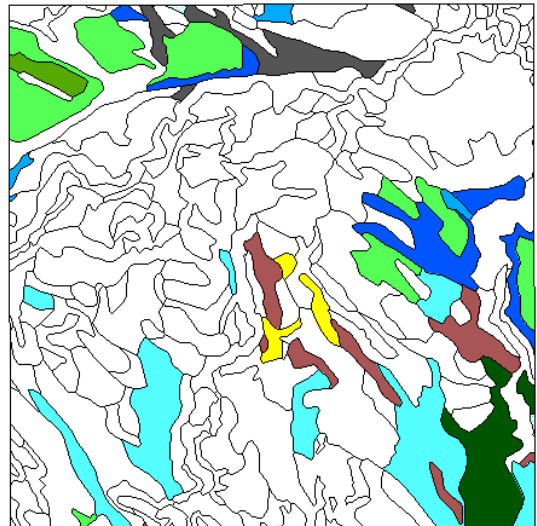
と、問い合わせ命令全体が真になって要素を選択できます。2つの比較式が論理演算子「or」でリンクされている場合は、独立した個々の比較結果のいずれかが真になれば、問い合わせ命令が真になります。いずれかの基準を満足する要素が選択されます。



長い問い合わせ命令を次の行に続けることもできますが、後の部分が1つの命令の一部分であることをわかりやすくするため、後の行は字下げした方が良いでしょう。



小麦の潜在産出高が1エーカー当たり34ブッシェルを超え、かつオート麦の潜在産出高が1エーカー当たり40ブッシェルを超えるポリゴン。



小麦の潜在産出高が1エーカー当たり34ブッシェルを超えるか、またはオート麦の潜在産出高が1エーカー当たり40ブッシェルを超えるポリゴン。

"not equal to" (不等) 演算子を使用する

Crow Butte 区域の大部分の土壌タイプでは、オート麦の方が小麦よりも潜在産出高が高くなっていますが、作物を売る場合の価格は通常、小麦の方がオート麦より高くなります。1ブッシェル当たりの作物価格が、オート麦の場合は3.25ドル、小麦の場合は4.00ドルであると仮定しましょう。この例の問い合わせでは、オート麦の1エーカー当たりの潜在作物価格 (1エーカー当たりの潜在産出ブッシェル数×1ブッシェル当たりの価格) が小麦以上になるような土壌タイプを、問い合わせを使用して調べます。

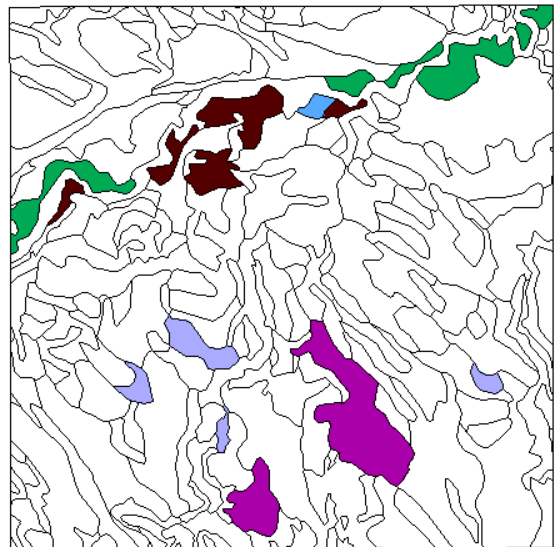
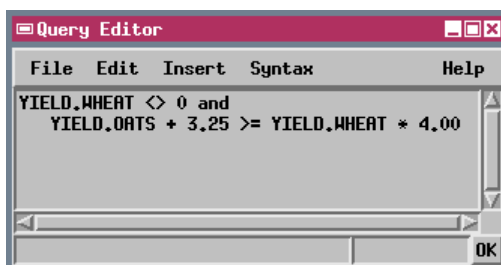
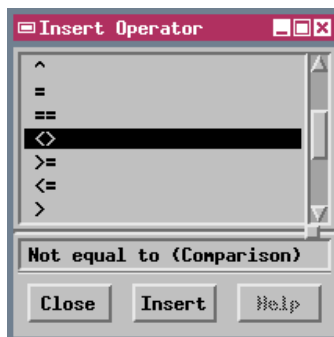
この問い合わせが複雑になるのは、耕作できない土壌タイプの潜在作物産出高が0になり、このような土壌タイプが第2行の選択用比較式を満足してしまうからです。問い合わせの第1行では、産出高が0であるような土壌タイプは除外され、"not equal to" (不等) 演算子 (<>または!=) の使用方法が良くわかります。この命令では、小麦の潜在産出高が0でないポリゴンが選択されます。これらのポリゴンだけが、第2行での価格比較の対象になります。

ステップ

- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウと Query Editor (問い合わせエディタ) ウィンドウを開きます。
- File (ファイル) メニューから New (新規) を選択します。
- Insert (挿入) 手順や手動入力を使用して、次に示す問い合わせを作成します。

```
YIELD.WHEAT <> 0 and
YIELD.OATS * 3.25 >= YIELD.WHEAT *
4.00
```

- Query Editor (問い合わせエディタ) ウィンドウで [OK] をクリックします。
- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで [OK] をクリックします。



潜在作物産出高と1ブッシェル当たりの価格を仮定した場合の、オート麦の1エーカー当たりの産出額が小麦より多くなるポリゴン。

コメントと変数を使用する

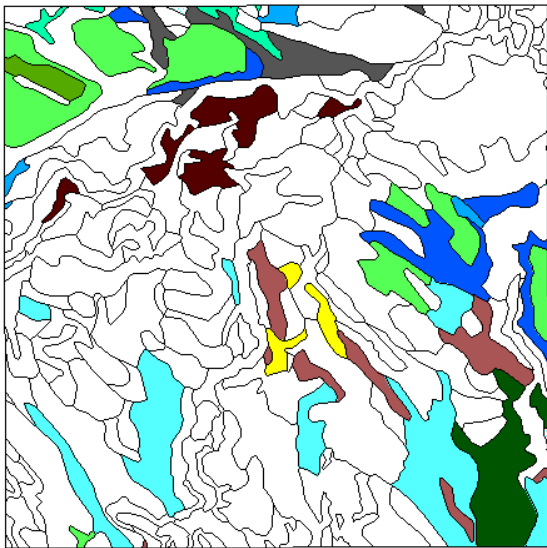
ステップ

- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウと Query Editor (問い合わせエディタ) ウィンドウを開きます。
- File (ファイル) メニューから New (新規) を選択します。
- Insert (挿入) 手順や手動入力を使用して、次に示す問い合わせを作成します。

```
dollars = 129 # define variable for
              # required crop
              # price per acre

# select polygons based on crop price
YIELD.OATS * 3.25 > dollars or
YIELD.WHEAT * 4.00 > dollars
```

- Query Editor (問い合わせエディタ) ウィンドウで [OK] をクリックします。
- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで [OK] をクリックします。

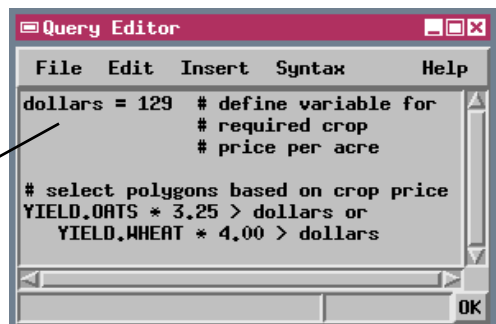


変数名は必ず小文字にし、最初は文字でなければなりません。変数名は、問い合わせコマンド名、データベース・テーブル名、フィールド名と同じではありません。

コメントを付けておくと、問い合わせが、読みやすく、また後で使用しやすくなります。コメントは「#」の記号から始まり、コメントだけの行として記述されたり、命令の終わりに付けられます。問い合わせの中で個々の命令を説明するコメントを使用したり、問い合わせの想定用途や対象とするオブジェクトについて概要説明するためにコメントを使用できます。

TNTmipsの問い合わせ処理では、問い合わせの中で使用する変数に名前を付けて値を設定することもできます。この例の問い合わせは、オート麦と小麦のどちらかに対して要求される1エーカー当たりの潜在作物価格を超える土壌ポリゴンを選択します。問い合わせの第1行は、必要な価格を保存するために「dollars」という数値変数を定義し、その値を129にするという代入命令です。「=」記号は変数に値を代入するために使用されます(「Equal to」(等しい)演算子に「==」が使用されるのはこのためです)。

変数は、同じ値が問い合わせの中で何回も使用される場合に役に立ちます。異なる必要価格でもう一度この問い合わせを実行したい場合にも、変数「dollars」に代入される1つの値を変更するだけで済みます。変数を使用せずにこの問い合わせが記述されている場合は、選択命令の中の2つの実際の数値を変更しなければなりません。



文字列変数を使用する

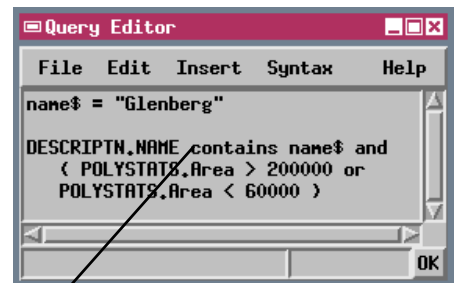
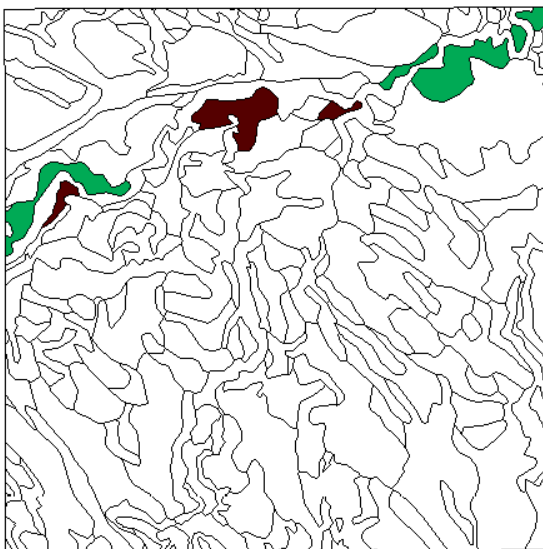
文字列値を含むように変数を定義することもできます。文字列変数用の名前を作成する場合は、最後に\$を付けなければなりません。この練習問題の問い合わせでは、文字列変数 name\$ を定義して値 "Glenberg" を代入します。この問い合わせは、Crow Butte区域の2つの土壤タイプを含むGlenberg土壌シリーズに属する土壤ポリゴンのサブセットを選択します。この問い合わせでは、2つのクラス記号を使用してポリゴンを選択する代わりに、どちらのクラスの場合でも DESCRIPTN テーブルの中の NAME フィールドには "Glenberg" という名前で始まる土壤説明があるということを利用します。この問い合わせは論理演算子「contains (含む)」を使用して、文字列フィールドのすべてまたは一部が指定された文字列と一致する要素を選択します。この場合、一致すべき文字列 ("Glenberg") は name\$ 変数の中に格納されています。この選択用比較式に適合するポリゴン群は、さらに標準 POLYSTATS テーブルの Area (面積) フィールドに保存された面積 (単位は平方メートル) により選別されます。(POLYSTATS テーブルが存在するのは、ベクタ・オブジェクトに対し標準属性が計算されている場合だけです。)

ステップ

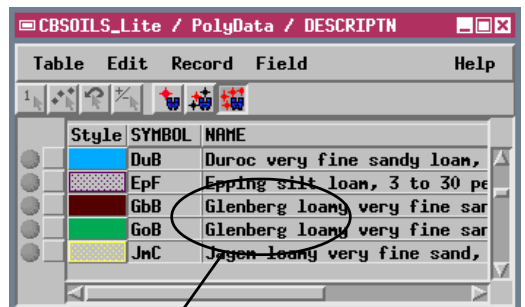
- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウと Query Editor (問い合わせエディタ) ウィンドウを開きます。
- File (ファイル) メニューから New (新規) を選択します。
- Insert (挿入) 手順や手動入力を使用して、次に示す問い合わせを作成します。

```
name$ = "Glenberg"
DESCRIPTN.NAME contains name$ and
( POLYSTATS.Area < 60000 or
POLYSTATS. Area > 200000 )
```

- Query Editor (問い合わせエディタ) ウィンドウで [OK] をクリックします。
- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで [OK] をクリックします。



「contains (含む)」演算子は、name\$ 変数の文字列が DESCRIPTN.NAME 文字列フィールドのいずれかの部分と一致するポリゴンを選択します。



Glenberg シリーズに属する両方の土壤タイプの DESCRIPTN.NAME フィールドに、テキスト文字列「Glenberg」が含まれています。

"not" (否定) 論理演算子を使用する

ステップ

- ✓ Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウと Query Editor (問い合わせエディタ) ウィンドウを開きます。
- ✓ File (ファイル) メニューから New (新規) を選択します。
- ✓ Insert (挿入) 手順や手動入力を使用して、次に示す問い合わせを作成します。

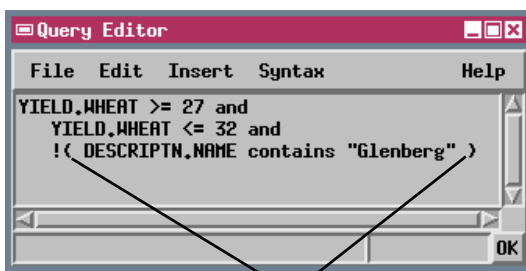
```
YIELD.WHEAT >= 27 and  
YIELD.WHEAT <= 32 and  
!( DESCRIPTN.NAME contains "Glenberg" )
```

- ✓ Query Editor (問い合わせエディタ) ウィンドウで [OK] をクリックします。
- ✓ Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで [OK] をクリックします。

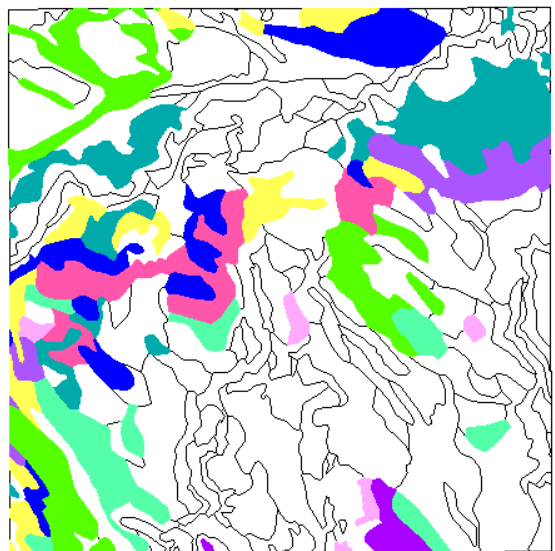
Crow Butte 区域の多くの土壌における小麦の潜在産出高は、2つの Glenberg の土壌の産出高 (1 エーカー当たり 27、32 ブッシェル) と同程度です。この練習問題の問い合わせでは、Glenberg の土壌『**以外で**』小麦の産出高の値がこの範囲に入るすべての土壌タイプを選択します。

この問い合わせの最初の 2 行で、小麦の潜在産出高が指定された範囲に入るポリゴンを選択します。問い合わせの第 3 行の最初の論理演算子「not (否定)」(!) は、この演算子の後に続く変数、演算子、式の結果を反転します。この例の場合、「not」演算子に続いている条件式は、2 つの Glenberg 土壌に属するポリゴンを選択するだけの条件式です。「not」演算子によって、この結果を逆、つまり前の産出高条件に適合するポリゴンの中から Glenberg 土壌ポリゴン以外のすべてのものを選択します。

「not」演算子は、選択したい値を持つものが多数あり、選択したくない値を持つものの方が数が少なく指定しやすい場合に役に立ちます。



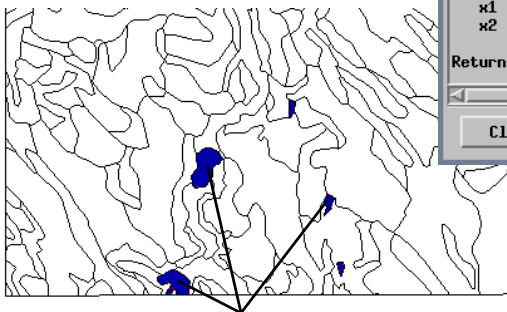
「not」演算子は、演算子に続く次の要素を逆にします (変数やその他の演算子を含む)。「not」演算子を (この例のように) 式全体に適用したい場合は、式をカッコで囲む必要があります。



レコードの添付をチェックする

この練習問題で使用される問い合わせは、前のどの練習問題で使用された問い合わせとも異なります。この問い合わせでは、特定の属性値を問い合わせる代わりにポリゴンに添付された特定のデータベース・テーブル内にあるレコード数を調べます。この問い合わせには、SetNum()集合関数を使用します。SetNum()集合関数は集合中の項目の数を返します。この場合、集合は、指定されたテーブル内の要素に添付されたすべてのレコードを意味するTable[*]という形の式で与えられます。

この問い合わせは、YIELD テーブル内に添付レコードが存在しないすべての土壌クラス・ポリゴンを識別します。これを逆にした問い合わせ [SetNum(YIELD[*]) > 1] は、複数のレコードが添付された要素を識別します。この形の問い合わせを使用すると、まだ属性が割り当てられていないポリゴンを識別したり、何らかの理由で余分なレコードが添付されているポリゴンを識別することができます。この例では、YIELD テーブル内に添付レコードが存在しないポリゴンはすべて、WATER クラスに属します。これらのポリゴンは小さい湖や池を示したものであるため、潜在作物産出高が存在しません。

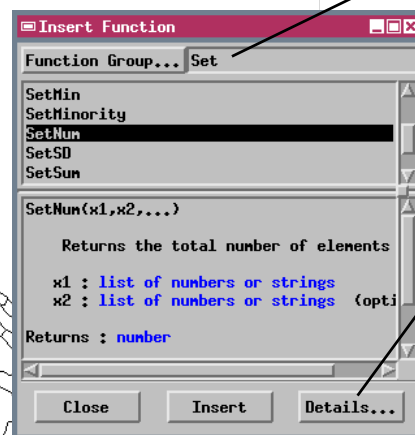


YIELD データベース・テーブル内に添付レコードが存在しないポリゴンはすべて、WATER クラスに属します。Group Controls(グループ・コントロール)や Layout Controls(レイアウト・コントロール)ウィンドウで対応するテーブルのアイコンを右クリックして、ポップアップメニューから Select All Unattached Elements(未添付の要素すべてを選択)を選択しても、レコードがない要素を選択できます。

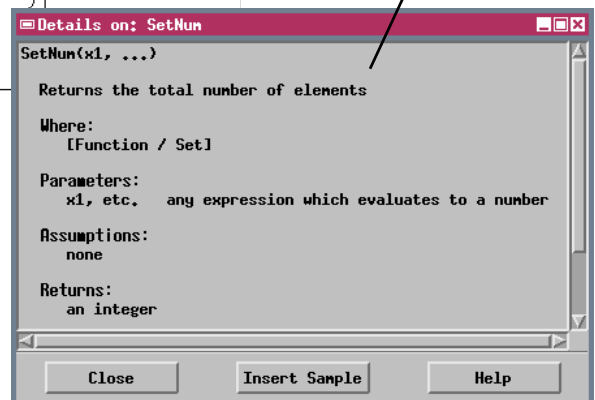
ステップ

- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウと Query Editor (問い合わせエディタ) ウィンドウを開きます。
 - File (ファイル) メニューから New (新規) を選択します。
 - Insert (挿入) 手順や手動入力を使用して、次に示す問い合わせを作成します。
- SetNum (YIELD [,*]) < 1
- Query Editor (問い合わせエディタ) ウィンドウで [OK] をクリックします。
 - Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで [OK] をクリックします。

Insert Function (関数を挿入) ウィンドウの Function Group (関数グループ) オプションを使用して、ウィンドウ内にリストする関数のグループを選択します。



現在選択されている関数やその使用方法に関する詳細を確認するには、Insert Function (関数を挿入) ウィンドウで [Details (詳細)] をクリックします。



複数の添付レコードを使用して選択する

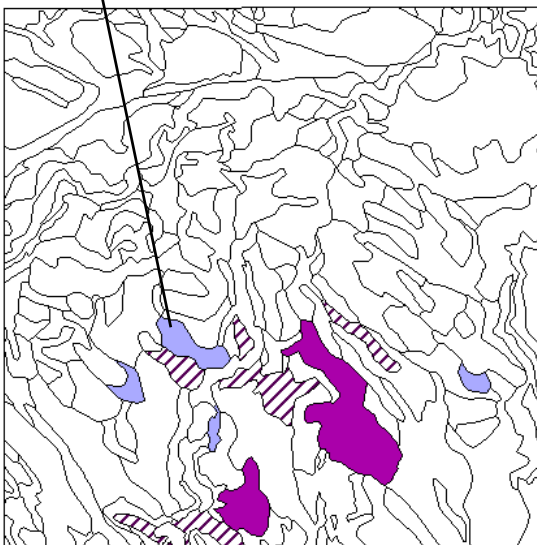
ステップ

- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウと Query Editor (問い合わせエディタ) ウィンドウを開きます。
- File (ファイル) メニューから New (新規) を選択します。
- Insert (挿入) 手順や手動入力を使用して、次に示す問い合わせを作成します。

```
"WB" in LAYER [*] . texture
```

- Query Editor ウィンドウで [OK] をクリックします。
- Vector Object Display Controls ウィンドウで [OK] をクリックします。

代表的な土壤プロファイルの一部が風化した基盤であるような土壤クラス・ポリゴン



CBSOILSQ オブジェクトの土壤クラス・ポリゴン用の LAYER テーブルには、各土壤の代表的な土壤プロファイル内のさまざまな層に関する情報が含まれています。プロファイル内の各層ごとに個別のレコードがあり、各土壤ポリゴンに複数のレコードが添付されています。多数の添付レコードの中から属性に基づいて要素を選択するには、特殊な問い合わせシンタックスが必要です。

たとえばこの練習問題では、土壤プロファイルのいずれか部分に風化した基盤を含んでいる土壤タイプを選択します。この属性は、texture フィールドで文字列「WB」としてコード化されています。通常の選択問い合わせ LAYER.texture=="WB" を使用すると、プロファイルの下部に風化した基盤がもつ土壤があるにもかかわらず、ポリゴンが1つも選択されません。この問い合わせ構文ではテーブル内の各ポリゴンに対する『最初』の添付レコード (この場合は通常、最も上の土壤層の属性を含む layer 1 レコード) しかチェックされません。より深部にある土壤レイヤーに対応するそれ以降のレコードは無視されます。

添付レコードの『すべて』の texture フィールドを問い合わせるには、LAYER[*].texture という式を使用する必要があります。こうすると、現在のポリゴンに添付した各レコードの texture フィールドの内容をリストした集合が返されます。この場合、この集合の中に、希望する属性「WB」に対応する要素があるか調べる必要があります。最も簡単なのは、論理演算子としてキーワード「in」を使用する方法です。演算子の前の変数が、演算子の後続く式で表される集合の要素のいずれかと完全に一致する場合、問い合わせの結果は真になります。この構文は、文字列フィールドにも数値フィールドにも使用できます。

島ポリゴンを見つける

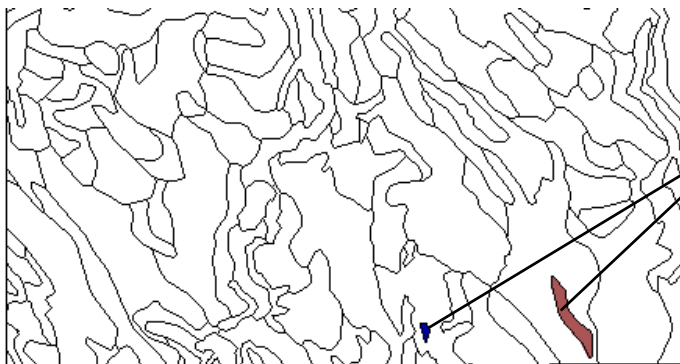
より大きいポリゴンの内側に完全に囲まれたベクタ・ポリゴンのことを**島ポリゴン**と言います。島ポリゴンの属性とその外側のポリゴンの属性が異なる場合も多いため、ベクタ・オブジェクトのトポロジーや属性の割り当てを変更する処理では島ポリゴンの関係が保たれなければなりません。

ポリゴンの**Internal**テーブルには、島ポリゴンに関する情報が含まれたいくつかのフィールドがあります。これらのフィールドに関する問い合わせを行うことで、島ポリゴンや、島を含むポリゴンを選択することができます。**Internal.Inside**フィールドには、囲む側のポリゴンの要素番号が含まれています。島ポリゴンのこのフィールドはすべて、1でない値になります。したがって、最初の問い合わせではすべての島ポリゴンが選択されます。**NumIslands**フィールドは、各ポリゴンに含まれる島の数を示します。この練習問題の第2の問い合わせでは、**NumIslands**の値が0より大きいポリゴン、すなわち島を含むすべてのポリゴンが選択されます。

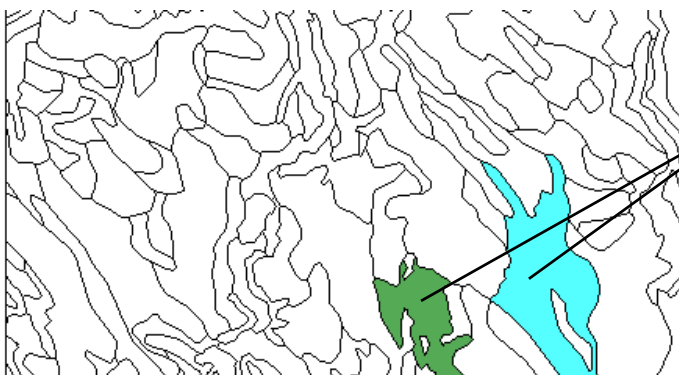
ステップ

- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウとQuery Editor (問い合わせエディタ) ウィンドウを開きます。
 - File (ファイル) メニューから New (新規) を選択します。
 - Insert (挿入) 機能を使用するか手入力により、次の問い合わせを作成します。
- Internal.Inside > 0
- Query Editor ウィンドウで [OK] をクリックします。
 - Vector Object Display Controls ウィンドウで [OK] をクリックします。
 - 次の問い合わせを使用して、上記のステップを繰り返します。

```
Internal.NumIslands > 0
```



最初の問い合わせで選択された島ポリゴン。それぞれの島は、それを囲む外側のポリゴンとは異なる土壌クラスに属します。



CBSOILS_LITE ベクタ・オブジェクト内の2つのポリゴンには島ポリゴンが含まれるため、第2の問い合わせで選択されます。

スクリプトによるスタイル設定

ステップ

- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウを開きます。
- Lines (線) タブ付きパネルで Draw Lines Before Polygons (ポリゴンより先に線を描画) トグル・ボタンをオンにします。

Style by Script (スクリプトによるスタイル設定) オプションを使用すると、選択された要素のサブセットの表示特性をその属性に基づいて指定することができます。スタイル・オプションを紹介するため、この練習問題では、前の選択問い合わせをそのまま使用しますが、ここでは、スタイル・スクリプトを使用して、選択されたすべてのポリゴンに対して新しい表示パラメータを設定します (通常、このためには All Same (すべて同じ) スタイル・オプションを使用します)。



スクリプトによってスタイルを設定する場合、Query Editor (問い合わせエディタ) ウィンドウの Insert (挿入) メニューから、表示特性を設定するその他の変数

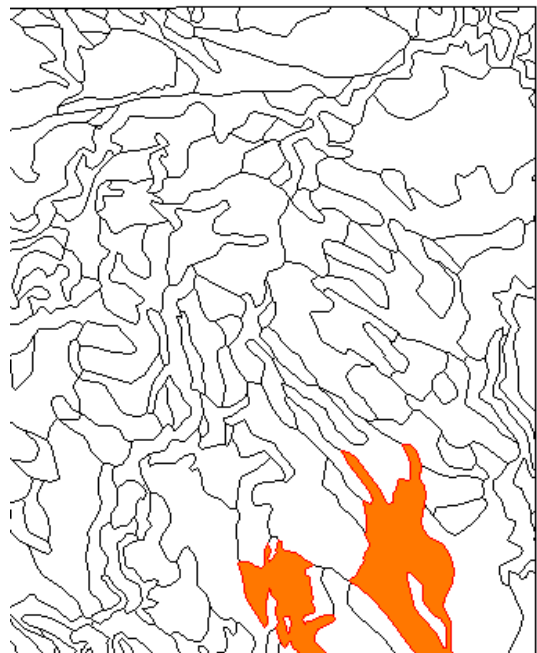
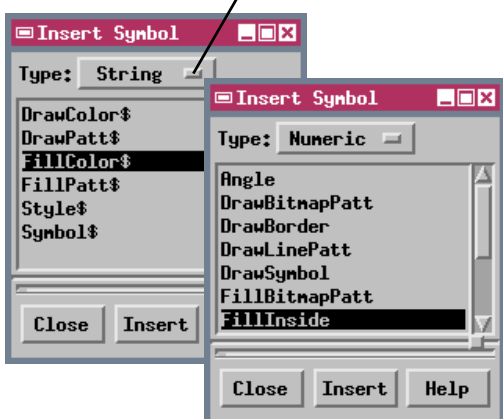
- Polygons (ポリゴン) タブ付きパネルで Style (スタイル) オプションを By Script (スクリプトにより) に設定し、隣の Specify (指定) ボタンをクリックします。
- Insert Symbol (記号を挿入) 手順 (Numeric (数値)、String (文字列) オプション) を使用して、次に示すスタイル・スクリプトを作成します。

にアクセスできます。FillInside と DrawBorder は数値変数であり、選択されたポリゴンを塗りつぶしてその周囲の枠を描画する場合は値を 1 に設定します。FillColor\$ と DrawColor\$ は文字列変数であり、それぞれポリゴンの塗りつぶしとポリゴンの枠に使用するカラーの設定に使用されます。これらの変数に設定できる値 (二重引用符で囲む) は、カラー名 (赤、緑、青、黒、白、黄色、オレンジ、茶、シアン、マゼンタ、グレー)、または一組の RGB 値 (それぞれの範囲は 0 ~ 100%) です。

```
FillInside = 1
FillColor$ = "100 50 0"
DrawBorder = 1
DrawColor$ = "red"
```

- Query Editor (問い合わせエディタ) ウィンドウで [OK] をクリックします。
- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで [OK] をクリックします。

Insert Symbol (記号を挿入) ウィンドウのオプション・ボタンを使用してさまざまな記号リストにアクセスします: 定数または各種変数 (特に数値、文字列、ベクタを含む)。



複合スタイル・スクリプト

この練習問題では、すべての土壌クラスのポリゴンが表示用に選択され、スタイル・スクリプトは、ポリゴンの面積に基づいてポリゴン表示用の2つの異なるパラメータ・セットを定義するのに使用されます。

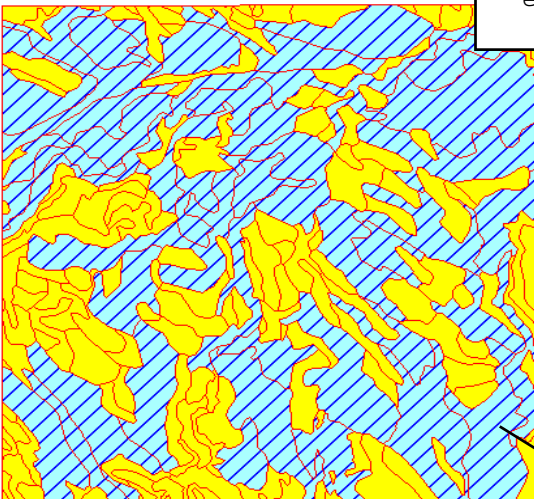
問い合わせやスタイル・スクリプトの中で二者択一的な処理を指定するには、「if-then-else」コマンドを使用してロジックを明確に定義する必要があります。このスクリプトの中の命令の意味を説明すると、「ポリゴンの面積が200000平方メートルより大きいか？ (if)、そうである場合は (then) 黄色でポリゴンを塗りつぶし、そうでない場合は (else) ビットマップ・パターン ('BitmapPatt4') でポリゴンを塗りつぶす」ということとなります。(この例のように)「then」や「else」コマンドの後に関連する命令が複数ある場合は、一連の命令を begin/end コマンドで囲む必要があります。この問い合わせの中の「else」の後のbegin/end コマンドを省略してもシンタックス・エラーにはなりません、その場合は、「then」でない場合の処理としては最初の命令しか使用されず、その他の命令は、(この問い合わせの最初の2行のように) 選択されたすべてのポリゴン全体に対して適用されるものと解釈されます。前に定義されたスタイル・パラメータは無視されます。

ステップ

- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウを開きます。
- Polygons (ポリゴン) というタブの付いたパネルでSelect (選択) オプションをAll (すべて) に設定し、Style (スタイル) オプションはBy Script (スクリプトにより) にしたまま、隣のSpecify (指定) ボタンをクリックします。
- File (ファイル) メニューからNew (新規) を選択します。
- Insert (挿入) 手順や手動入力を使用して、下記の枠内に示す問い合わせを作成します。
- Query Editor (問い合わせエディタ) ウィンドウで [OK] をクリックします。
- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで [OK] をクリックします。

```

DrawBorder = 1
DrawColor$ = "red"
if ( POLYSTATS.Area < 200000 ) then
  begin
    FillInside = 1
    FillColor$ = "yellow"
  end
else
  begin
    FillInside = 1
    FillBitmapPatt = 1
    FillPatt$ = "BitmapPatt2"
  end
end
    
```



スクリプトの中でビットマップ塗りつぶしパターンを使用するには、FillPatt\$変数に設定するパターンが、オブジェクトに対して定義されたパターンの User Set (ユーザ・セット) の中になければなりません。パターンを描画するには、変数FillInsideとFillBitmapPattを両方とも1に設定する必要があります。塗りつぶしパターンやその他のスタイルの作成についての詳細は『TNT入門:スタイルの作成と使用』を参照してください。

面積が200,000平方メートル以上のポリゴンはBitmapPatt2で定義された縞模様のパターンで埋め込まれ、これより小さいポリゴンは黄色で塗りつぶされます。

点を囲むポリゴンを見つける

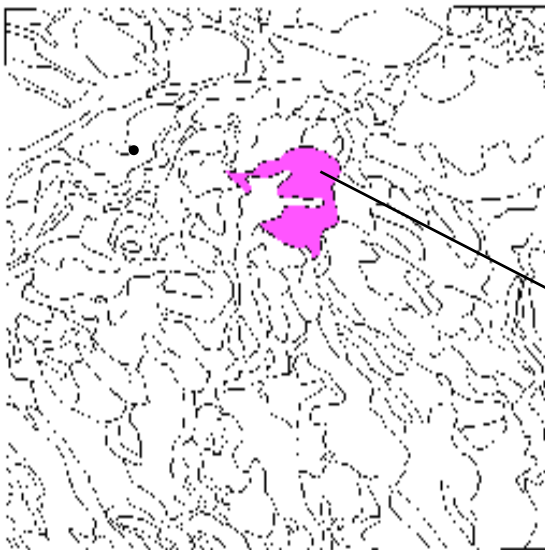
ステップ

- ✓ Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウを開きます。
- ✓ Polygons (ポリゴン) タブ付きパネルで Style (スタイル) オプションを By Attribute (属性により) に設定します。
- ✓ Select (選択) オプションを By Query (問い合わせにより) に設定し、隣の Specify (指定) ボタンをクリックします。
- ✓ Query Editor (問い合わせエディタ) ウィンドウの File (ファイル) メニューから New (新規) を選択します。
- ✓ Insert (挿入) 手順や手動入力を使用して、下記の枠内に示す問い合わせを作成します。
- ✓ Query Editor (問い合わせエディタ) ウィンドウで [OK] をクリックします。
- ✓ Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで [OK] をクリックします。

問い合わせ処理には、要素の選択に使用可能な数多くの空間関数が組み込まれています。(ベクタ関数グループの中の) FindClosestPoly() 関数は、指定されたx,y座標を持つ点を囲むポリゴンの要素番号を返します。この関数に必要なパラメータは、問い合わせるベクタ・オブジェクト、x,y座標値、および座標値の処理に使用されるジオリファレンス・サブオブジェクトのオブジェクトの番号です。ジオリファレンス・オブジェクトの番号は (Georeference 関数グループの中の) 関数 GetLastUsedGeorefObject() によって得られ、変数「georef」に対する代入命令の中で使用されます。(Insert Symbol (記号を挿入) ウィンドウの Vector リストの中の) あらかじめ定義された変数「Vect」は、現在のベクタ・オブジェクトを指定するために使用されます。問い合わせの最初の2つの行には、点の x,y 座標値が含まれています。

FindClosestPoly() 関数は、囲むポリゴンの要素番号を数値変数(この例では c_poly) に格納する代入命令の中で使用されます。問い合わせの中の最後の命令は、オブジェクト内の各ポリゴンの要素番号 (Internal.ElemNum) を、c_poly に格納されている番号と比較し、表示用の一致するポリゴンを見つけます。

```
xvar = -103.33861
yvar = 42.73583
georef = GetLastUsedGeorefObject( Vect )
c_poly = FindClosestPoly( Vect, xvar, yvar, georef )
Internal.ElemNum == c_poly
```



点の位置は座標 xvar (経度) と yvar (緯度) によって指定されています。問い合わせの中で使用される値は、指定されたジオリファレンス・サブオブジェクトと同じ座標系のものでなければなりません。緯度 / 経度座標は度単位で 10 進数により指定しなければなりません。

ポリゴンの隣接性に関する問い合わせ：考え方

選択用の問い合わせでは、ベクタ・オブジェクトに対応付けられたトポロジー情報も使用できます。ベクタ・オブジェクト内の各線には開始ノードと終了ノードがあり、これにより線の左側と右側が決まります。各ポリゴンは特定の線要素から構成され、線に対応するInternal(内部)テーブルには、線の両側にあるポリゴンの要素番号が格納されたフィールドが含まれています。(Insert Function (関数を挿入) ウィンドウの Vector(ベクタ)リストの中の) GetVectorPolyAdjacentPolyList() 関数は、この情報を使用して、現在のポリゴンにどのポリゴンが隣接しているかを判断します。問い合わせの中でこの関数を使用すると、特定のポリゴン・クラスに隣接するポリゴンを選択できます。

例として、土壌クラス「SrD」に属し、かつクラス「Sa」のポリゴンに隣接するポリゴンを選択するCBSOILS_LITEベクタ・オブジェクトに対する問い合わせを調べてみましょう。ポリゴンが隣接しているときみなされるためには、ノードだけでなく境界線も共有していなければなりません。このような問い合わせで使用される一般的な考え方を、次に示します。

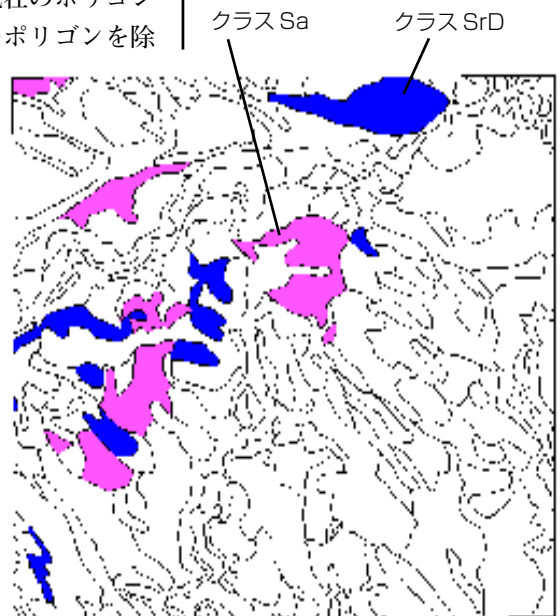
- 1) 選択するクラスを定義します。
- 2) 選択されたクラスにポリゴンが属する場合は、以下のステップ (隣接性のチェック) を実行し、そうでない場合はそのポリゴンを除外します。
- 3) 現在のポリゴンに隣接するポリゴンのリストを得ます。
- 4) 隣接する各ポリゴンのクラス割り当てをチェックします。これらのいずれかが定義された隣接クラスに一致する場合は、現在のポリゴンを表示用を選択します。一致するものがない場合はそのポリゴンを除外します。

次ページでは、この問い合わせのシンタックスについて説明します

ステップ

- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウとQuery Editor (問い合わせエディタ) ウィンドウを開きます。
- File (ファイル) メニューから New (新規) を選択します。
- Insert (挿入) 手順や手動入力を使用して、次ページに示す問い合わせを入力します。
- Query Editor (問い合わせエディタ) ウィンドウで [OK] をクリックします。
- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで [OK] をクリックします。

クラスがSrDおよびSaであるすべてのポリゴンが選択されたベクタ・オブジェクトCBSOILS_LITE (次ページの図と比較してください)。



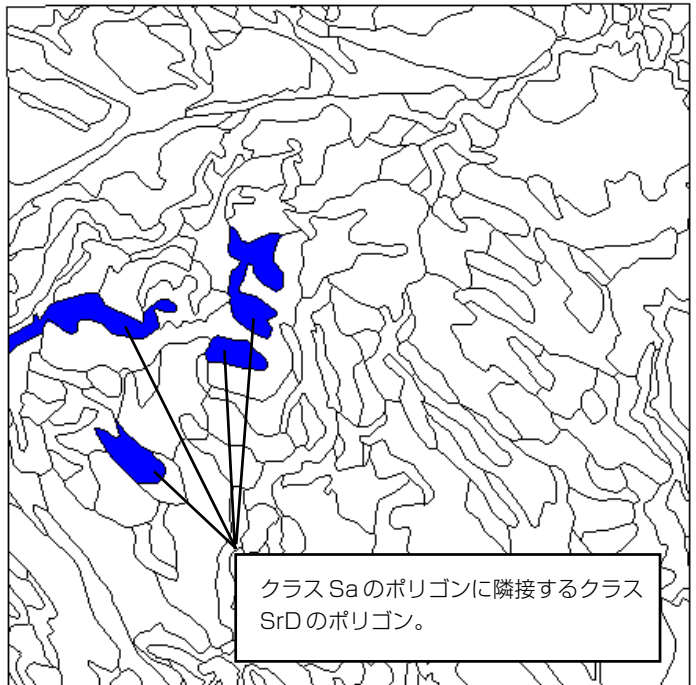
ポリゴンの隣接性に関する問い合わせ：シンタックス

```

1  if (CLASS.Class == "SrD") then
2      begin
3          array polylist [10]
4          numpolys = GetVectorPolyAdjacentPolyList(Vect, polylist)
5          for i = 1 to numpolys begin
6              polynum = polylist[i]
7              if (Vect.poly[polynum].CLASS.Class$ == "Sa")then
8                  return 1
9          end
10         return 0
11     end
12 else return

```

1. 以降のテスト用にクラスがSrDであるポリゴンを条件により選択します。
2. 隣接するポリゴンのクラスをチェックする処理ループを開始します。
3. 現在のポリゴンに隣接するポリゴンの要素番号のリストを格納する「polylist」という一次元配列を定義します。配列サイズを要素10個分に初期化します(配列サイズは次の命令で関数により自動的に変更されます)。
4. 現在のポリゴンに隣接するポリゴンの数に等しい値が代入される「numpolys」という変数を定義します。この値はGetVectorPolyAdjacentPolyList関数により返されます。また、この関数は隣接するポリゴンの要素番号を見つけて「polylist」配列に格納します。あらかじめ定義された変数「Vect」は、現在のベクタ・オブジェクトを指定するために使用されます。
5. 配列内の各ポリゴンのクラスをチェックする処理ループを開始します。このループは、配列内の(配列のインデックスが1である)第1位置から(「numpolys」変数の現在の値で指定される)最終位置までの各要素に対して1回実行されます。各ループの中で、変数「i」には現在の配列インデックスの値が代入され、次の命令で使用されます。
6. (配列内のインデックスにより指定される)現在の隣接するポリゴンの番号を変数「polynum」に代入します。
7. 現在の隣接するポリゴンのクラスを調べ、指定された隣接クラスと比較します。データベースは「Object.database [record#].table.field」の形で指定します(データベース指定の最後の"\$"は、ターゲット・フィールドが文字列フィールドであることを示します)。クラスが一致した場合は、...
8. 「return 1」命令は、上記の条件を満足するポリゴンに対して問い合わせ結果が真であることを明確に示します。したがって、このポリゴンは表示用に選択されます。
9. 配列処理ループの終わりを示します。
10. 隣接するすべてのポリゴンが上記のクラス・テストに適合しなかった場合は、「return 0」命令により問い合わせ結果が偽になったことを示します。
11. ポリゴン隣接性ループの終わりを示します。
12. ポリゴンが命令1の初期クラス選択条件に適合しないため問い合わせ結果が偽になったことを示します。





TIGER データにおける国勢調査の境界

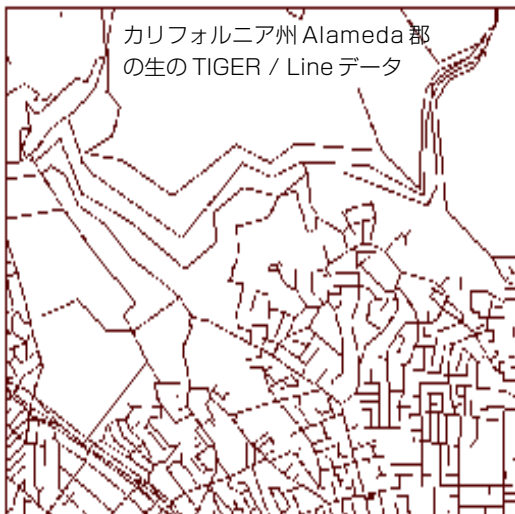
U.S. Census Bureau (米国国勢調査局) の TIGER / Line ファイルからインポートされたベクタ・オブジェクトの構成要素である線セグメントとしては、天然および人工の物理的地物だけでなく、行政上や国勢調査上の境界もあります。国勢調査区域 (およびこれに相当する Block Numbering Areas (ブロック番号区域: BNA) およびその要素である国勢調査ブロックの境界は通常、地図上の他の地物と一致しますが、地図上の基本的な地物のように Census Feature Class Code (国勢調査地物クラス・コード: CFCC) で明確に識別されません。

表示や抽出のために国勢調査ブロックの境界線を選択するには、左側と右側のブロック番号が異なる線を選択するような問い合わせを使用します。より小さい部分に分割されたブロックでは、ブロック番号は同じになりますが、左と右のブロック接尾辞フィールドの異なる文字によって識別されます。問い合わせ例の2番目の命令群では、これらの境界が選択されます。また、隣接する BNA 内のブロックの番号を同じにすることができますので、最後の命令では異なる BNA を分離する線が選択されます。

ステップ


- 前の練習問題で使用したレイヤーを削除します。 
- Add Vector (ベクタを追加) アイコンをクリックし、ドロップダウン・メニューから Add Vector Layer (ベクタ・レイヤーを追加) を選択します。 
- TIGER プロジェクトファイルから ALAMEDA オブジェクトを選択します。
- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウを開き、Lines (線) パネルの Style (スタイル) オプションを All Same (すべて同じ) に、Select (選択) オプションを By Query (問い合わせにより) に設定します。
- Query Editor (問い合わせエディタ) ウィンドウを開き、File (ファイル) メニューから New (新規) を選択します。
- Insert (挿入) 手順や手動入力を使用して、下記の枠内に示す問い合わせを作成します。
- Query Editor (問い合わせエディタ) ウィンドウで [OK] をクリックします。
- Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウで [OK] をクリックします。

```
Basic_Data.Block_Left <> Basic_Data.Block_Right
or ( Basic_Data.Block_Left == Basic_Data.Block_Right
and Basic_Data.BlockSuff_Left <>
Basic_Data.BlockSuff_Right )
or Basic_Data.BNANum_Left <> Basic_Data.BNANum_Right
```



複数のレコードからの計算結果用のフィールド

ステップ

- 前の練習問題の Display (表示) ウィンドウを最小化します。
- TNTmipsのメイン・メニューから Edit / Attribute Databases (編集 / 属性データベース) を選択します。
- CBSOILSQプロジェクト・ファイルの CBSOILS_LITE オブジェクトを捜して選択します。
- Select (選択) ウィンドウの Polygon (ポリゴン) ラジオボタンをオンにします。
- Database Editor (データベース・エディタ) ウィンドウで、Class テーブルに対応するボックスを右クリックし、Edit Definition (定義を編集) を選択します。
- Class テーブルの定義ウィンドウで、Add Field (フィールドを追加) ボタンをクリックします。 
- フィールド・リストで、新しいフィールド用のデフォルト名をハイライト表示にして ClassArea と入力します。
- Field info (フィールド情報) パネルで Field Type (フィールドのタイプ) メニューから Computed (計算結果) を選択して Edit Expression (式を編集) をクリックします。
- Query Editor (問い合わせエディタ) ウィンドウに次の問い合わせを入力します。

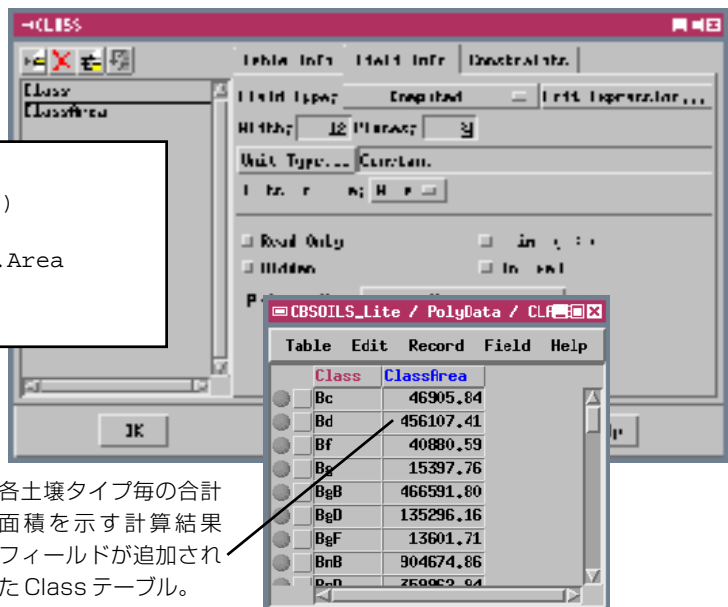
```
sum = 0.0
num = SetNum( POLYSTATS[*] )
for i = 1 to num begin
    sum = sum + POLYSTATS[i].Area
end
return sum
```

- Query Editor ウィンドウで [OK] をクリックします。
- Width (幅) テキストボックスに 12 と入力し、Places (位置) テキストボックスに 2 と入力します。
- 定義ウィンドウで [OK] をクリックします。
- Class テーブルに対応するボックスをダブルクリックして開きます。

スクリプトを使用することで、データベース・テーブル内の計算結果フィールドの値を定義することもできます。多くの場合、これらのスクリプトに要求されるのは、同じレコード内の他のフィールド群の単純な算術結合を作成することだけですが、この練習問題の処理内容はいちより複雑であり、CBSOILS_LITE のポリゴンの Class テーブルの中に、各土壌タイプの合計面積を示す計算結果フィールドを作成します。

ポリゴンの面積は、各ポリゴン毎に POLYSTATS テーブルに保存されていますが、ここでは、各土壌タイプ毎に1つのレコードを持つ Class テーブル内に計算結果のフィールドを作成します。ここに示すスクリプトは、各土壌クラスのポリゴンの面積の和を求めて計算結果フィールドの値としてこの合計値を返すように作られています。

このスクリプトは、POLYSTAT.Area フィールド内で面積の総和を求めるのに用いる数値変数「sum」を定義します。この変数は各クラスに対して、はじめは 0.0 にリセットされていなければなりません。変数「num」は、(現在の土壌クラスに対して) POLYSTATS テーブルの添付レコードの数と等しい数値に割り当てられます。この変数は面積の総和を求めるループの反復回数を設定するために用いられます。



各土壌タイプ毎の合計面積を示す計算結果フィールドが追加された Class テーブル。

- Table / Close (テーブル / 閉じる) を選択して Class テーブルを閉じます。

文字列式のフィールド

文字列フィールドはデータベース・テーブル内にある特殊なタイプの計算結果フィールドです。文字列式フィールドの用途として最も単純なのは、リンクされた別のテーブル内の文字列フィールドの内容を現在のテーブルにコピーするという使用方法です。この場合の式は単に、対応する参照文字列名TABLE.FIELDとなっています。また、文字列式を使用して複数の文字列フィールドの内容を結合して新しい1つのフィールドにすることもできます。たとえば、NAME というテーブルに姓と名前用の個別のフィールドがあるとします。この場合に「+ (追加)」演算子を使用してこれらの文字列を結合することができます。次の式

```
NAME.FIRST + " " + NAME.LAST
```

を使用すると "John Doe" のような形の項目が生成されます。式には、上記のように引用符で囲んだ区切り文字(スペースやカンマ)が含まれていなければなりません。式による結合文字列のフィールドを使用することで、よりわかりやすいデータティップやラベル用のテキストを作成することができます。

この練習問題で使用される式には、複雑な文字列式をより簡単にフォーマットできる sprintf()関数を使用されています。この関数の第1パラメータは(引用符で囲まれた) 制御文字列であり、その後に参照する文字列フィールド名が続きます。制御文字列の中のそれぞれの「%s」という項目は、リストされている参照文字列フィールド名のうちの1つに対応します。制御文字列には、挿入するテキストやスペース、句読点などを含めることもできます。


```
sprintf( "Sec %s Twp %s Rng %s", Sections.Section,
Sections.Township, Section.Range )
```



| Section | Township | Range | SecTwpRng |
|---------|----------|-------|------------------------|
| 34 | 31N | 51W | Sec 34 Twp 31N Rng 51W |
| 35 | 31N | 51W | Sec 35 Twp 31N Rng 51W |
| 36 | 31N | 51W | Sec 36 Twp 31N Rng 51W |
| 13 | 32N | 51W | Sec 13 Twp 32N Rng 51W |
| 14 | 32N | 51W | Sec 14 Twp 32N Rng 51W |
| 15 | 32N | 51W | Sec 15 Twp 32N Rng 51W |
| 16 | 32N | 51W | Sec 16 Twp 32N Rng 51W |

文字列式で作成されフォーマットされたテキスト

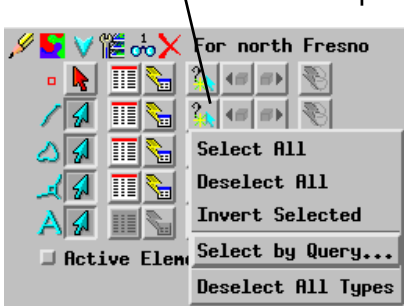
ステップ

- ✓ Database Editor (データベース・エディタ) ウィンドウで File / Open (ファイル / 開く) を選択します。
- ✓ CB_SECT プロジェクト・ファイルの CBSECT オブジェクトを捜して選択します。
- ✓ Select (選択) ウィンドウの Polygon (ポリゴン) ラジオボタンをオンにします。
- ✓ Database Editor (データベース・エディタ) ウィンドウで、Sections テーブルに対応するボックスを右クリックし、Edit Definition (定義を編集) を選択します。
- ✓ Sections テーブルの定義ウィンドウで、リスト内の Range (範囲) フィールドをクリックし、Add Field (フィールドを追加) ボタンをクリックします。 
- ✓ フィールド・リストで、新しいフィールド用のデフォルト名をハイライト表示にして SecTwpRng と入力します。
- ✓ Field Type (フィールドのタイプ) メニューから String Expression (文字列式) を選択して Edit Expression (式を編集) をクリックします。
- ✓ Query Editor (問い合わせエディタ) ウィンドウに次の問い合わせを入力します。

- ✓ Query Editor ウィンドウで [OK] をクリックします。
- ✓ Width (幅) テキストボックスに 25 と入力します。
- ✓ 定義ウィンドウで [OK] をクリックします。
- ✓ Sections テーブルに対応するボックスをダブルクリックして開きます。
- ✓ この練習問題が終わったら、両方の Database Editor ウィンドウで File / Close (ファイル / 閉じる) を選択します。

数値化におけるエラーをチェックする問い合わせ

Select / Deselect (選択/非選択) アイコン・ボタンをクリックし、Select by Query (問い合わせにより選択) を選択し、標準の Query Editor (問い合わせエディタ) ウィンドウを開くと、その要素タイプ用の問い合わせを入力できます。

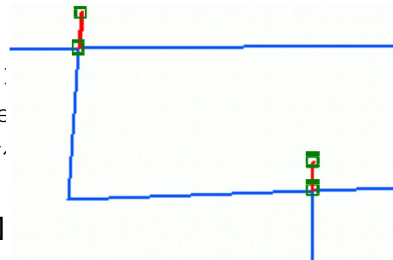


TNTmipsSpatial Data Editor (空間データ・エディタ) を使用してベクタ・オブジェクトを作成したり編集する場合にも、選択用の問い合わせが役に立ちます。複雑なベクタ・オブジェクトには、線のオーバーシュート、閉じていないポリゴン、細長いポリゴンなど、数値化の際のエラーが含まれていることがあります。これらのエラーの多くは倍率を大きくしない限り目には見えませんが、手動操作でチェックするのは難しく、時間がかかります。次の例に示すような問い合わせを使用すると、潜在的なトポロジー上の問題部分を探す作業をスピードアップできます。Spatial Data Editor (空間データ・エディタ) ウィンドウに用意されたアイコンを使用すると、特定の要素タイプを選択するための問い合わせを作成して使用できます。

オーバーシュート

オーバーシュートとは、間違っ交差部を越えて伸びた短い線セグメントのことです。ベクタ・オブジェクトに対して Standard Attribute (標準属性) 処理を実行すると、線の長さに基づいて極端に短い線をすて選択してチェックし、可能ならば削除することもできます。

LINESTATS.Length < [ユーザが指定する長さの値]



閉じていないポリゴン

ポリゴン群のネットワークを1つ含む1つのベクタ・オブジェクト内で、交差すべき2つの線の間にギャップがあると、2つの別々のポリゴンが存在すべきところにポリゴンが1つしかできません。線のエラーのためにポリゴンが閉じていない部分では線の両側が同じポリゴンになりますので、問い合わせによってこの部分を見つけることができます。

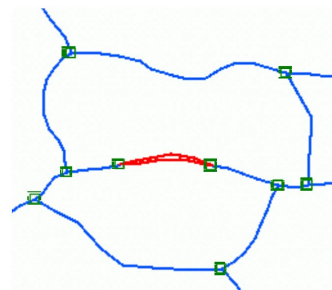
Internal.LeftPoly == Internal.RightPoly



細長いポリゴン

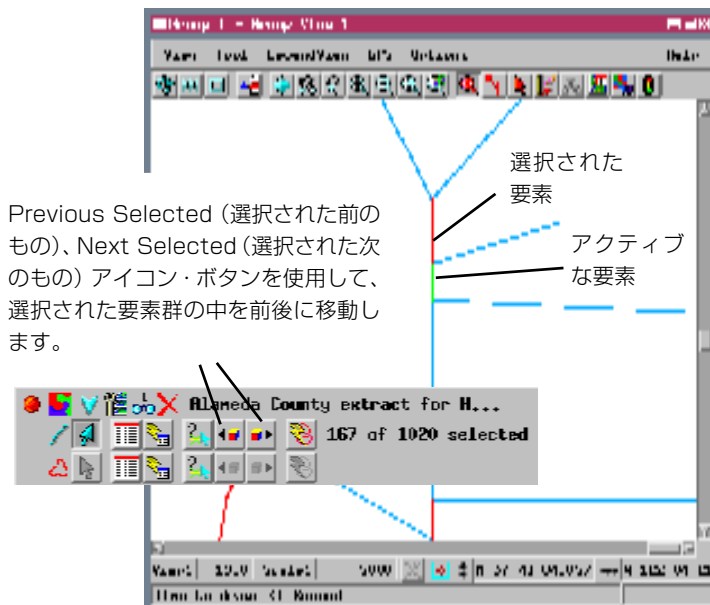
ポリゴンの境界を二重にトレースすると、隣接する2つのポリゴンの境界に沿って極端に細長いポリゴンが形成されることがあります。通常、細長いポリゴンの面積は主ポリゴンに比べてはるかに小さく、極端に細長くなります (Compactness Ratio (矮小率) が大きい)。POLYSTATS テーブルの中の Area (面積) フィールドと CompactRatio (矮小率) フィールドに対し、組み合わせた問い合わせを使用して、細長いポリゴンを選択します。

**POLYSTATS.Area < [ユーザが指定した面積の値] または
POLYSTATS.CompactRatio > 3.00**



問い合わせにより移動する

Spatial Data Editor (空間データ・エディタ) または Spatial Data Display (空間データ表示) の Element Selection (要素選択) ウィンドウから実行される問い合わせで、複数の要素が選択されることが、よくあります。選択されたこれらの要素のうちの1つが「アクティブな」要素として指定されます。アクティブな要素と選択された要素は異なるカラーでハイライト表示されます。編集操作は、アクティブな要素に対しても選択された要素に対しても行えます。Element Selection (要素選択) ウィンドウの Previous Selected (選択された前のもの)、Next Selected (選択された次のもの) アイコン・ボタンを使用すると、選択された要素群の中を前後に移動できます。このとき各要素が1つずつ順番にアクティブになります。現在アクティブな要素が表示されるよう、ビューの位置は (必要に応じ) 自動的に調整されます。この「問い合わせによる移動」機能を使用すると、選択された要素を拡大表示のままで1つずつ容易にたどりながら、各要素をチェック (必要に応じ編集も) することができます。



ステップ

- ✓ View (ビュー) ウィンドウと Group Controls (グループ・コントロール) ウィンドウを復帰させます。
- ✓ Vector Object Display Controls (ベクタ・オブジェクト表示コントロール) ウィンドウを開きます。
- ✓ Lines (線) パネルの Select (選択) オプションを All (すべて) に設定して [OK] をクリックします。
- ✓ Spatial Data Display View (空間データ表示ビュー) ウィンドウで Options (オプション) メニューを開き、Show Scale / Position (縮尺 / 位置を表示) オプションがオンになっていることを確認します。
- ✓ View (ビュー) ウィンドウ下部の Scale (縮尺) テキスト・ボックスに「5000」と入力し、<Enter> を押します。
- ✓ Layer (レイヤー) アイコンの行の Show Details (詳細を表示) アイコン・ボタンをクリックします。
- ✓ 線用の Select (選択) アイコン・ボタンをクリックします。
- ✓ 線用の Select / Deselect (選択/非選択) アイコン・ボタンをクリックし、ドロップダウンメニューから Select by Query (問い合わせにより選択) を選択します。
- ✓ Query Editor (問い合わせエディタ) に次の問い合わせを入力します。

```
LINESTATS.Length < 50
```
- ✓ Select by Query (問い合わせにより選択) ウィンドウで [Apply (適用)] をクリックします。
- ✓ Next Selected (選択された次のもの) アイコン・ボタンをクリックします。

本書では、さまざまな練習問題を通して、TNTmips、TNTedit、TNTviewで使用するデータベース問い合わせの基本的な構造とシンタックスについて紹介してきました。問い合わせ言語は、TNTmipsで使用される Spatial Manipulation Language (空間操作言語: SML) のサブセットとなっており、共通のシンタックスを使用しています。2ページに示した問い合わせに関する資料のほかに、TNTmipsリファレンス・マニュアルに示したSMLに関する資料も参照してください。問い合わせ機能を拡張するためのプログラミングについてさらに詳しいヒントが示されています。

地理空間解析のための先進的ソフトウェア

マイクロイメージ社は、地理空間データの視覚化、解析、出版の高度な処理を行う、専門家向けソフトウェアを提供しています。製品に関する詳細は、マイクロイメージ社にお問い合わせになるか、ウェブ・サイトにアクセスしてください。

- TNTmips** TNTmipsは、GIS、画像解析、CAD、TIN、デスクトップマッピング、地理空間データベース管理機能を統合した専門家のためのシステムです。
- TNTedit** TNTeditはベクタ、画像、CAD、TIN、様々な形式のリレーショナルデータベースプロジェクトデータを作成、ジオリファレンス、編集するための対話的ツールを提供します。
- TNTview** TNTviewには、複雑な地理空間データの視覚化と解釈を行うための強力な表示機能があります。TNTmipsの演算処理機能や加工機能を必要としないユーザに最適です。
- TNTatlas** TNTatlasを使用すると、自分で作成した空間プロジェクトデータをCD-ROMにプレスして、低コストで出版や配布ができます。TNTatlasのCDはどのようなコンピュータにも対応できます。
- TNTserver** TNTserverを使うとTNTatlasのデータをインターネットやイントラネットで公開することができます。ユーザのウェブ・ブラウザ上の地理データ・アトラスやTNTclient Java アプレットを使って操作して下さい。
- TNTlite** TNTliteは、学生や小規模プロジェクトを行う専門家向けの無料バージョンです。マイクロイメージ社のウェブ・サイトからTNTliteをダウンロードできます。また、TNTliteの入ったCDを注文することもできます。

索引

| | | | |
|------------------------------------|---------|--------------------------|--------|
| Insert Field(フィールドを挿入) ウィンドウ | 16 | 比較演算子(続き) | |
| Insert Field(演算子を挿入) ウィンドウ | 5 | contains (含む) | 13 |
| 計算結果用フィールド | 24,25 | 複合問い合わせ | 24,25 |
| コメント | 12 | 変数 | 12,13 |
| 算術計算 | 9 | 文字列フィールド | 7 |
| シンタックス、チェック | 8 | 文字列式フィールド | 25 |
| See 要素選択、問い合わせの保存 | 9 | 文字列変数 | 13 |
| SetNum 関数 | 15 | 要素選択 | |
| スクリプトによるスタイル設定 | 3,18,19 | アンダーシュート | 26 |
| 代入命令 | 12 | オーバーシュート | 26 |
| Table[*]式 | 15 | 島ポリゴン | 17 |
| TIGER data | 23 | 添付されていないレコード | 15 |
| データベース問い合わせ、定義 | 3 | 点を囲むポリゴン | 20 |
| 問い合わせを聞く | 9 | 問い合わせによる | 3 |
| 問い合わせによる移動 | 27 | 複数の添付レコード | 16 |
| 比較演算子 | 4 | 細長いポリゴン | 26 |
| equal to (== 等しい) | 5 | 隣接するポリゴン | 121,22 |
| greater then (> より大) | 4 | 論理演算子 (and.or.not) | 10,14 |
| not equal to (<> 不等) | 11 | | |



MicroImages, Inc.

11th Floor - Sharp Tower
206 South 13th Street
Lincoln, Nebraska 68508-2010 USA

電話 : (402) 477-9554
FAX : (402) 477-9559

email : info@microimages.com
インターネット : www.microimages.com

[翻訳]



株式会社 オープン GIS

〒130-0001 東京都墨田区吾妻橋 1-19-14

1-19-14 Azumabashi, Sumida-ku, Tokyo 130-0001, JAPAN

TEL (03) 3623-2851 FAX (03) 3623-3025