Oracle Database 内での言語データのソート

オラクル・テクニカル・ホワイト・ペーパー
2003 年9 月

ORACLE

Oracle Database 内での言語データのソート

Oracle Corporation 発行「Sorting Your Linguistic Data inside an Oracle Database」の翻訳版です。

Oracle Database 内での言語データのソート

概要

文字列のソートは、大変複雑な操作ですが、ユーザーの目には、その複雑さが分からないことがよくあります。言語が異なれば、ソートのルールも異なります。アルファベットの文字順にソートする言語もあれば、文字の画数でソートする言語もあります。また、単語の発音によってソート順が決まる言語もあります。文字のアクセント記号の扱いも言語によって異なります。

日常的な英語でも、ソートは簡単ではありません。英語の辞書を見ると、単語の順に関しては、大文字と小文字が区別されていません。電話帳で名前を探していると、たとえば、接頭辞の Mac と Mc が同じグループに入っているように、異なる単語が同じ扱いであることに気付きます。

1 つ以上の言語からのデータをソートする必要がある場合、ソートはさらに複雑になります。ある文字が 1 つの言語では文字  の後にソートされ、別の言語では文字 *a* の前にソートされる場合は、どのように処理すればよいのでしょうか?

このホワイト・ペーパーでは、テキスト・リストが特定の言語においてソートされたと見なされる場合に、テキスト・リストの文化的な意味からの要求要素の順序を表わすために、言語ソートまたは照合という用語を使用します。言語のネイティブ・スピーカーは、この順に基づき、他の要素に対して相対的に、リストの要素を見つけることを当然のこととして期待します。たとえば、英語を話すユーザーは、英語の単語が並べられたリストで、B で始まる単語は、A で始まる単語全部の後にあり、C で始まる単語の前にあるべきであると考えます。

ユーザーが慣れている言語の順序でソートされていないデータが提示された場合、情報の検索は難しく、時間のかかる作業になります。

Oracle9*i* Database 以降、多言語でデータを検索し、ソートする必要のある顧客の要求に応えるために、バイナリ・ソート、言語ソートの適用範囲が拡張され、新たな多言語ソートが導入されました。このホワイト・ペーパーでは言語ソートの基本的な概念を概略し、ソート・プロパティが異なる言語の照合順に与える影響を例で示し、Oracle Database 内でデータをソートする方法をどのようにしてカスタマイズするかを説明します。

Oracle Corporation 発行「Sorting Your Linguistic Data inside an Oracle Database」の翻訳版です。

ソートのルールの世界

前述の項では、文化が要求する順序でデータをソートすることに伴う複雑さに触れました。このセクションでは、すべての書体（たとえば、ラテン文字、ギリシア文字、キリル文字など）には固有の順序があり、書体には、サポートする様々な言語に対して、多数の競合する順序を持つものがあるということを詳しく説明します。

西ヨーロッパの言語

ドイツ語では、*ß* は 1 つの文字ですが、配列のためには、2 つの s、つまり *ss* として扱われます。ドイツ語を話す人々は、*ü*、*ä*、*ö* をそれぞれ *ue*、*ae*、*oe* と同等のものとして取り扱うことに慣れています。この 3 つの文字は、母音ペアと同じ順でソートされます。

スペイン語では伝統的に、*ch*、*ll*、*ñ* を独立した文字として扱い、それぞれ *c*、*l*、*n* の後に配置します。たとえば、次のスペイン語の単語は、現在の並び順にソートされます。

cabalmente、caba***ll***a、cantina、ca***ñ***a、clamar、curador、***ch***ácara

最近、この伝統的なスペイン語のソート方法が近代的なスペイン語のソートに置き換えられ、*ch* と *ll* の特別なステータスがなくなりました。

デンマークのアルファベットでは 3 つの追加文字 *æ*、*ø*、*å* は、 の後にソートされます。また、文字の組み合わせ *aa* をアルファベットの最後に、*å* の後に配置する独立した文字として扱い、*ü* と *y* をその文字の変形と見なします。これらの理由により、都市名 Zürich、Aarfit、Årbus は、都市のリストで Zyrardow の後に表示されます。

フランス語ではアクセント記号が付いた母音は、まず、基本的なソート順の観点から、アクセント記号のない母音のように扱われます。アクセント記号を考慮せずにテキストをソートした後、さらに各母音セットの中で、アクセントなし、揚音アクセント、抑音アクセント、曲折アクセント、ウムラウトの順になるようにソートされます。アクセント記号付きの母音は右から左に評価されますが、アクセント記号なしの基本文字は左から右に考慮されるため、ソートはさらに複雑になり、フランス語ソートを使用して、2 つの文字列を照合する場合、*È*dit は *Edít* の前になります。

アクセント記号付きの文字は、言語によって、また同一言語の中でさえ意味が異なり、別の複雑さがあります。数ヶ国語をサポートする Unicode のような多国語のコード体系を扱う場合、これらの言語が、同じ文字に対して相反するアルファベットのルールを含んでいる可能性があるため、混乱が生じます。たとえば、文字 *ä*（ウムラウト付き）は、ドイツ語では、*b* の前にソートされますが、スウェーデン語では、 の後にソートされます。このように、ソートするテキストが多数の言語に属する場合、すべての言語の観点から見て、正しくソートすることは不可能です。

アジアの表意文字

これまで述べたように、言語の間には文化的な期待値の差があります。前述の例は、同じラテン語の書体を使用する西ヨーロッパの言語だけを取り上げたものです。アルファベットの 26 から 40 文字の処理が、かなりの問題であるとすると、アジアの表意文字の処理は、気が遠くなるようなタスクだと言えます。中国語だけでも 4 万以上の文字があります。簡体字（中国とシンガポールで使用される）中国語と繁体字（台湾、香港およびマカオで使用される）中国語の書体の間でも、ソートのルールは異なります。

漢字のソートに使用される最も一般的な照合方法の例を、次に示します。

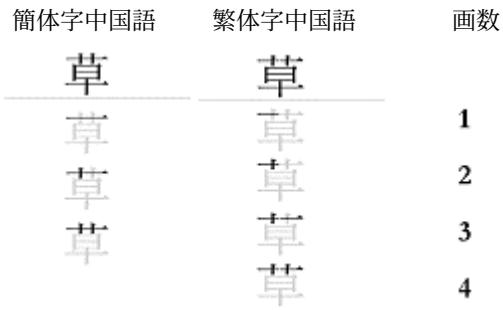
画数 – 漢字をソートする最も一般的な方法。漢字は、部首（基本的な要素）と 0 または複数の構成要素で形成されます。部首と構成要素は両方とも、字画で構成されます。漢字によって、画数は異なります。同じ画数の漢字は、部首によってさらにソートされます。

次の例に、3 つの漢字が、どのように、画数に基づいて配列されるかを示します。

串（画数: 7、部首: 丨）< 吟（画数: 7、部首: 口）< 壑（画数: 17、部首: 土）

ただし、簡体字と繁体字の間で、特定の漢字の画数が異なる場合があります。したがって、2 つの字体を処理するために、2 つの異なったソート・アルゴリズムを使用することが一般的です。

次に、草という漢字を例に、2 つの字体の画数がどのように数えられるかを示します。



部首 – ほとんどの中国語辞書ではこの配列があります。これは画数による配列に似ていますが、部首による配列が画数よりも優先されます。これは漢字をソートする、より伝統的な方法ですが、特定の漢字の正確な部首を直観的に見つけることができない可能性があるため、中国語のネイティブ・スピーカーにとってさえ、重荷になることがあります。部首による配列というのは、部首の画数による配列です。同じ順位の部首を持つ字の場合、字の画数を使用します。

次の例に、3 つの漢字が、部首の画数に基づきどのように配列されるかを示します。

土（部首：土、画数：3）＜地（部首：土、画数：6）＜石（部首：石、画数：5）

ただし、部首には、他の構成部分と結合して、最終的に形を変えるものがあります。部首の順位は、常に原形に基づきます。

次の例に、部首の最初の画数に基づく、4 つの漢字の順位を示します。カッコ内の名前は、漢字の Unicode を示します。

漢字	部首	部首の画数	部首の原形	部首の原形の画数
艾(U8271)		4	艸	6
怕(U6015)		3	心	4
阪(U962A)		3	阜	8
邦(U90A6)		3	邑	7

=> 怕 ＜ 艾 ＜ 邦 ＜ 阪

前述の画数によるソートの場合と同じように、漢字には簡体字と繁体字の間で部首が異なるものがあり、部首のソートがさらに複雑になります。したがって、2 つの字体の相違点を処理するために、2 つの異なったソート・アルゴリズムを使用することが一般的です。

次の例に、同じ漢字のセットの簡体字中国語（SC）と繁体字中国語（TC）の部首の違いを示します。

SC 部首	TC 部首	例
王(U738B)	玉(U7389)	珍珠
(U6237)	(U6236)	房戾
(U9752)	(U9751)	靜靚
(U9EC4)	(U9EC3)	
一(U4E00)	二(U4E8C)	于互
彳(U5F73)	行(U884C)	街衍
月(U6708)	肉(U8089)	肌肚

発音（ぴん音） – ぴん音は簡体字中国語の、発音をベースとするソートです。これに対して、繁体字中国語には、注音字母があります。発音によるソートは、簡単に見えますが、コンテキストによって発音が異なる単語があります。また、同じぴん音を持つ漢字に 5 つの異なる声調があるケースもあります。中国語の話し言葉に 4 つの声調に加えて、特殊な軽い声調があります。声調は 1 から 5 で印され、軽いものが 5 番目で、声調では最後の順位です。同じぴん音と声調を持つ漢字は、さらに画数でソートされます。

次の例に、4 つの漢字が、ぴん音の順序に基づきどのように配列されるかを示します。カッコ内の名前は、漢字のぴん音の値と声調の値を示します。

阿(A1) < 伯(Bo2) < 㗎(Bo5) < 出(Chu1)

ISO/IEC 14651 – INTERNATIONAL STRING ORDERING

フランス語の同形異義語は、ネイティブ・スピーカーには直観的にわかるにもかかわらず、さまざまな市販のソート・プログラムで、処理が首尾一貫していませんでした。1980 年代半ばに、これに気付いたケベックのアラン・ラボンテは、驚き当惑しました。彼は、多数の文字種を配列するための汎用的な方法論を定義できる国際標準の必要性を実感しました。

1992 年には、国際的な文字列の配列標準である ISO 14651 のプロジェクトが立ち上げられました。この国際標準により、全世界のテキスト・データを配列する方法と共通テンプレート表が提供されます。この表が持つ適合化機能は、一定の言語と文化の要件を満たす一方で、他の文字の汎用的なプロパティを維持します。

共通テンプレート表は、異なるローカル環境では調整が必要です。ただし、この国際標準に準拠する場合、「delta」と呼ばれる、テンプレートからはずれるものすべてを宣言して、結果として生じる不一致を記録する必要があります。この標準には、コンテキストとは無関係にテキスト・データを配列する方法が記述されています。

Oracle Database 内のソート

テキストは、従来と同じように、文字のエンコードに使用されるバイナリ・コードに従って、データベース内でソートされます。通常、これは、言語からの正しいソート順にはなりません。一部の例では、ソート順が正しい場合もありますが、それは、特定のコード体系が、適切なアルファベットの規則に従って、すべての文字をバイナリ値の昇順に指定する場合です。残念ながら、ほとんどのコード体系は、そのような規則に従いません。もし、従ったとしても、前述の複雑なソート・シナリオを扱うことはできません。

このような制限をなくすために、オラクル社は言語ソートを提供します。言語ソートは、様々な言語と文化の複雑なソート要件に対応します。あらゆる文字コード体系のテキストが、特定の言語の規則に従って、文字のバイナリ値と無関係にソートされるようになります。

Oracle9*i* 以降の製品でサポートされるデータベースのソートは、次の 3 種類です。

1. バイナリ・ソート
2. 単一言語ソート
3. 多言語ソート

バイナリ・ソート

文字データをソートする、最も一般的な方法は、文字コード体系で定義され、数値で示されるバイナリ・コードでの配列です。これは、バイナリ・ソートの使用によって、実行されます。これは、データベースでの最速のソートです。その理由は、ソートされる値に特別な処理をする必要がないこと、ソートされる列に直接作成されるデータベース索引（標準の索引）が通常、言語索引よりも小さく、後述するように、検索するためのディスク読込みが少ないためです。

ASCII 規格と EBCDIC 規格の両方で、A から Z までの文字が数値の昇順に定義されるため、バイナリ・ソートを使用すると、英語のアルファベットに対して妥当な結果が出ます。ただし、大文字と小文字が別々に分類されるので、完全ではありません。ASCII では、大文字は小文字の前に表示されますが、EBCDIC では、その逆になります。

バイナリ・ソートから生じる結果は、特定のキャラクタ・セット内の文字の配列によって、異なります。英語以外の言語で使用される文字が存在する場合、バイナリ・ソートでは通常、よい結果が得られません。たとえば、昇順の問合せ ORDER BY は、文字 C、E、b、Ë、â を返します。これは、先行する各文字が、後続の文字よりも文字コード体系の数字コードが低いからです。言語のソート・ルールが複雑な場合、バイナリ・ソートは、言語学的に意味のあるデータを提示できません。

バイナリ・ソートには、BINARY という名前が付けられています。Oracle は、次の 5 つの準バイナリ・ソートもサポートします。UNICODE_BINARY、ASCII7、EBCDIC、BIG5、GBK および HKSCS。これらのソートは、それぞれ AL16UTF16、US7ASCII、WE8EBCDIC1047、ZHT16BIG5、ZHS16GBK および ZHT16HKSCS キャラクタ・セットの文字のバイナリ・コードに従ってデータをソートするという点では、バイナリです。ただし、比較の前に、ソートされた文字列を正しいキャラクタ・セットに変換する必要があるため、BINARY ソートと同等ではありません。つまり、標準の BINARY 索引を使用して、これらのソートによって配列されるデータの問合せ要件を満たすことはできません。単一言語ソートと同じように、言語索引を定義する必要があります。

単一言語ソート

特定の言語の表記規則に順守したローカライズされたソート基準を作成するには、文字コード化体系の中でバイナリ・コードから独立して文字をソートできる、別のソート技法を使用する必要があります。この技法は、言語ソートと呼ばれます。言語ソートは、数値（別名: ソート・キー）で文字を置き換えることで機能します。この数値は、指定された言語の各文字の適切な語学上の順序を反映します。単一言語ソートは、Oracle6 で最初に導入されました。

ソート・キーを構成するには、アルファベットの各文字を 2 つの部分、メジャー値とマイナー値に分けます。通常、同じ外観（またはベース文字）を持つ文字は、同一のメジャー値を持ちます。マイナー値は同一のベース文字の発音区別記号とケース・バリアントを識別するために使用されます。

次の表に、文字のサンプルとメジャー値およびマイナー値を示します。

文字	メジャー値	マイナー値
a	10	5
A	10	10
ä	10	15
Ä	10	20
B	20	5

単一言語ソートの場合、文字列は 2 つのステップで比較されます。アルゴリズムは、比較される値にソート・キーを生成し、異なるソート・キーを見つけるまで、または一方がより短くなるまで、ソート・キーを 1 バイトずつ比較します。より短いキーまたはより低い値のバイトを持つキーが小さいと見なされます。

ソート・キーは、ゼロ値とすべてのマイナー値が後に続く、文字列の全文字の大きいソート値全部を連結することにより、生成されます。こうして、メジャー値は、基本的なソート順を決定します。同一のメジャー値を持つ文字列の場合に限って、さらに分類するためにマイナー値が比較されます。

文字列の比較で、スペースやハイフンのような文字を考慮しない場合は、ソート・キーのためにマイナー値のみが使用されます。マイナー値は見落とされないため、無視できる文字だけが異なる文字列は等しいと見なされません。

SQL NLSSORT 関数の結果は、RAW データ型のソート・キーです。

Oracle では、名前の付いた言語ソートを使用して、文字データをソートする方法を指定します。言語ソートの名前は、ほとんどの場合、言語名を使用して定義されます。一部の言語の場合、追加的な拡張された言語ソートが定義されます。従来どおり、これらの言語ソートの名前の前には、X が付きます。

たとえば、Oracle は 2 つのスペイン語の単一言語ソート、近代スペイン語用の `SPANISH` と伝統的なスペイン語用の `XSPANISH` をサポートします。

拡張言語ソートは、言語特有のケースに便宜を図るために設計されています。

1. 二重字、例えばスペイン語の *ll* と *ch* を 1 つの文字としてソートします。
2. 1 つの文字をソートの目的で、二重字に変換します。たとえば、ドイツ語の無声音の s' β 'は、*ss* として処理されます。

サポートする単一言語ソートの完全なリストを次に示します。太字で示した言語ソートは、Oracle Database 10g で追加されたソートです。

ARABIC	EEC_EUROPA3	POLISH
ARABIC_MATCH	ESTONIAN	PUNCTUATION
ARABIC_ABJ_SORT	FINNISH	XPUNCTUATION
ARABIC_ABJ_MATCH	FRENCH	ROMANIAN
AZERBAIJANI	XFRENCH	RUSSIAN
XAZERBAIJANI	GERMAN	SLOVAK
BENGALI	XGERMAN	XSLOVAK
BULGARIAN	GERMAN_DIN	SLOVENIAN
CANADIAN FRENCH	XGERMAN_DIN	XSLOVENIAN
CATALAN	GREEK	SPANISH
XCATALAN	HEBREW	XSPANISH
CROATIAN	HUNGARIAN	SWEDISH
XCROATIAN	XHUNGARIAN	SWISS
CZECH	ICELANDIC	XSWISS
XCZECH	INDONESIAN	THAI_DICTIONARY
CZECH_PUNCTUATION	ITALIAN	THAI_TELEPHONE
XCZECH_PUNCTUATION	JAPANESE	TURKISH
DANISH	LATIN	XTURKISH
XDANISH	LATVIAN	UKRAINIAN
DUTCH	LITHUANIAN	VIETNAMESE
XDUTCH	MALAY	WEST_EUROPEAN
EEC_EURO	NORWEGIAN	XWEST_EUROPEAN

言語ソートは、実際の文字コード化体系と無関係であるため、上記の言語ソートは、データが ASCII または EBCDIC コード化形式のどちらで保存されているかに関係なく、指定できます。

多言語ソート

単一言語ソートは、1 つの言語でデータを比較したり、ソートする場合に役立ちますが、多言語や複数の記述法にまたがったデータをソートできません。Oracle9*i* では、1 回のソートで複数の言語のデータをソートできるように、多言語ソートが導入されました。インターネットが発達しているため、現在、多くの企業がビジネスをグローバル・ビジネスに変えようとしています。顧客が単一のグローバル・データベースに様々な言語のデータを格納するのは普通のことになりつつあります。多言語ソートは、多言語のデータベースを持つ顧客が、任意の言語の情報を検索し、編成する際に役立ちます。

Oracle の多言語ソートは、ISO/IEC 14651 と Unicode 照合アルゴリズム規格をベースとします。複雑な多言語ソートの要件に対応すると同時に、変更に対してより

大きな柔軟性を提供するために、多言語ソートは、前述の 2 つの標準が定義するように、3 レベルの精度で評価されます。

1. 第 1 レベル: 第 1 レベルのソートは、*a* と *b* の違いなど、基本文字を識別します。無視できる文字の場合、第 1 レベルの順位（値）にゼロが割り当てられます。これは、このレベルでのソート比較の間、その文字を無視できることを意味します。無視できる文字には、たとえば、ダッシュ「-」があります。ダッシュを無視する場合、*multi-lingual* という単語を *multilingual* と同じものとして処理できます。
2. 第 2 レベル: このレベルは、特定の基本文字の異なる発音区別記号を識別するために使用されます。たとえば、文字 *Ö* は、発音区別記号があるというだけの理由で、文字 *O* と異なります。*Ö* と *O* は、同じ基本文字 *O* を持っているため、第 1 レベルでは同じですが、第 2 レベルでは異なります。第 2 レベルは、文字列の末尾から先頭に向かって文字を考慮する場合に指定できます。これは、フランス語のソート順に必要です。
3. 第 3 レベル: 第 3 レベルでは、第 1 レベルと第 2 レベルで違いのない文字のケーシング（大文字と小文字）を区別します。

リストが 3 つのレベル全部に基き、どのようにソートされるかを次の例に示します。第 1 レベルでは、resume の順位は resumes の前になります。第 2 レベルでは、発音区別記号のない文字列が、発音区別記号の付いた文字列より前に来ます。第 3 レベルでは、小文字が大文字の前にソートされます。

```
resume
Resume
résumé
Résumé
resumes
```

この 3 レベルのアーキテクチャにより、Oracle Database は複雑なソート・ルールを持つ言語を処理し、多言語のデータを持つデータベースに対して言語ソートのサポートが提供できます。中国語、日本語、韓国語のようなアジアの言語を対象とする言語ソートが、初めて使用可能になりました。実際にオラクル社では、中国語のソートのために、画数、ぴん音、部首に基づく、3 種類のソートを用意しています。

さらに、多言語ソートは、標準的な等価と追加文字のサポートも提供します。

Unicode の 1 つの文字は、結合文字を加えた基本文字の文字列に等しい場合があります。これは、標準的な等価と呼ばれます。たとえば、文字 *ä* は、基本文字 *a* と分音記号¨の組み合わせと同等です。Unicode 規格に準じて、標準的な同等の文字列は、同じものとしてソートする必要があります。この理由で、言語に依存する比較が行われる前に、文字列は正規の形式に整えられます。追加文字は、新たに定義された文字で、Unicode 3.1 規格で導入され、格納に 4 バイトが必要です。追加文字を使用することにより、さらに 100 万の文字を Unicode 規格にエンコードすることが可能です。現在、10 万以下の文字が、最新の Unicode 規格 3.2 で定義されています。Oracle の新しい多言語ソートのアーキテクチャでは、最大 110 万文字を 1 回のソートで定義できます。これには、Unicode 規格に追加できるすべての文字が含まれます。

ISO/IEC 14651: International String Ordering に基づき、オラクル社は、GENERIC_M と呼ばれる共通の多言語照合（ソート）テンプレートを準備しています。GENERIC_M は、ラテン文字、キリル文字、ギリシア文字をベースとする言語を対象とし、ソート定義の始めに、文字のアクセント記号と句読点記号文字を定義します。大文字と小文字はグループ化され、小文字で書かれた各基本文字が大文字の前に配置されます。全角の数字とローマ数字を含む、数字のグループ化も対象になります。GENERIC_M テンプレートでは、約 1000 文字が定義されています。定義されない文字は、Unicode バイナリ順に従って、ソートされます。

Oracle の多言語ソートは、すべて GENERIC_M テンプレートに基づき作成されます。ISO 14651 規格に基づく多言語ソートであることを示すために、ソート名には接尾辞_M が付きます。

Oracle でサポートされる多言語ソートの完全なリストを次に示します。

GENERIC_M	JAPANESE_M	SPANISH_M
	KOREAN_M	TCHINESE_RADICAL_M
CANADIAN_M	SCHINESE_STROKE_M	TCHINESE_STROKE_M
DANISH_M	SCHINESE_PINYIN_M	THAI_M
FRENCH_M	SCHINESE_RADICAL_M	

例えば、Oracle は、XFRENCH と呼ばれるフランス語の単一言語ソートをサポートします。FRENCH_M と呼ばれる多言語のフランス語ソートを指定することによっても、同じ動作を得ることができますが、このソートは GENERIC_M ソート順に基づき、フランス語で必要とされる、発音区別記号の右から左へのソートができます。多言語ソートの利点は、ISO 14651 規格で定義された国際的に承認される順序で、フランス語以外の言語に特有の文字の配列が可能であることです。

データベースに複数言語のデータが含まれる場合は、多言語ソートをお薦めします。単言語データベースに多言語ソートを使用することには、パフォーマンスの面で大きな利点はありません。単言語ソートは、多言語ソートよりもやや高速です。これは、単言語ソートには、1 つの言語のアルファベットのみが含まれているため、定義されている文字が少なく、処理時間が短くなるからです。

言語ソートのパラメータ

SQL 問合せでロケール特有の動作を判断するために、NLS パラメータを使用します。ほとんどの NLS パラメータは、データベース・セッション・レベルで構成できます。データベース・セッションで文化的背景を切り替えることは、多言語をサポートするアプリケーションにとって、きわめて重要です。これにより、異なるロケール要件を持つユーザーが、同一のデータベースに接続できるようになります。

パラメータ NLS_SORT は、ユーザーの SQL セッションの言語ソート・プロパティを管理します。

Paramter type	String
Syntax	NLS SORT = {BINARY | linguistic sort}
Default value	Derived from NLS_LANGUAGE

Parameter scope	Initialization Parameter, Environment Variable and ALTER SESSION
Range of values	BINARY or any valid linguistic sort definition name

NLS_SORT は、NLS_LANGUAGE と呼ばれる他のパラメータによって暗黙的に定義されます。特定のユーザー・セッションで、NLS_LANGUAGE の値が変更されると、NLS_SORT の値が変わる場合があります。

パラメータ NLS_SORT は、ORDER BY 問合せの照合順を指定します。値が BINARY の場合、照合順は、基盤となるコード体系の文字の数字コードに基づきます。これは、データ型によって、データベース・キャラクタ・セットの 2 進数列の順または各国語キャラクタ・セットの順のいずれかになります。

その値が名前の付いた言語ソートである場合は、ソートは定義されたソートの順に基づきます。NLS_LANGUAGE パラメータがサポートする多くの言語は、同じ名前の言語ソートもサポートします。

次の表に、Oracle がサポートする各言語のデフォルトの NLS_SORT 値を表示します。Oracle Database 10g で新たに追加された言語は太字で示します。

NLS_LANGUAGE	NLS_SORT
AMERICAN	BINARY
ARABIC	ARABIC
ASSAMESE	BINARY
AZERBAIJANI	**AZERBAIJANI**
BANGLA	BINARY
BENGALI	BENGALI
BRAZILIAN	WEST_EUROPEAN
PORTUGUESE	BULGARIAN
BULGARIAN	CANADIAN FRENCH
CANADIAN FRENCH	CATALAN
CATALAN	CROATIAN
CROATIAN	CZECH
CZECH	DANISH
DANISH	DUTCH
DUTCH	ARABIC
EGYPTIAN	BINARY
ENGLISH	ESTONIAN
ESTONIAN	FINNISH
FINNISH	FRENCH
FRENCH	GERMAN
GERMAN	GERMAN
GERMAN DIN	GREEK
GREEK	BINARY
GUJARATI	HEBREW
HEBREW	BINARY
HINDI	HUNGARIAN
HUNGARIAN	ICELANDIC
ICELANDIC	INDONESIAN
INDONESIAN	WEST_EUROPEAN
ITALIAN	BINARY

JAPANESE	BINARY
KANNADA	BINARY
KOREAN	SPANISH
LATIN AMERICAN	LATVIAN
SPANISH	LITHUANIAN
LATVIAN	MALAY
LITHUANIAN	BINARY
MALAY	BINARY
MALAYALAM	WEST_EUROPEAN
MARATHI	NORWEGIAN
MEXICAN SPANISH	BINARY
NORWEGIAN	POLISH
ORIYA	WEST_EUROPEAN
POLISH	BINARY
PORTUGUESE	ROMANIAN
PUNJABI	RUSSIAN
ROMANIAN	BINARY
RUSSIAN	SLOVAK
SIMPLIFIED CHINESE	SLOVENIAN
SLOVAK	SPANISH
SLOVENIAN	SWEDISH
SPANISH	BINARY
SWEDISH	BINARY
TAMIL	THAI_DICTIONARY
TELUGU	BINARY
THAI	TURKISH
TRADITIONAL CHINESE	UKRAINIAN
TURKISH	VIETNAMESE
UKRAINIAN	
VIETNAMESE	

一般的に、バイナリ・ソートの場合、他のソートよりもオーバーヘッドが少なくなります。この理由は、キーのバイナリ順に従って作成される標準的な Oracle 索引が、「言語索引の使用」で説明する言語索引よりも小さいためです。

NLS_LANG 環境変数は、NLS_SORT 動作に影響を与えることもあります。NLS_SORT は、NLS_LANG 環境変数の <language>コンポーネントによって定義されるように、特定の NLS_LANGUAGE パラメータに割り当てられるデフォルト値に変更されます。特定のセッションに、正しい言語のシーケンスが使用されるためには、ユーザーが NLS_SORT パラメータを明示的に設定することをお薦めします。

次の例に、バイナリ・ソート、単言語のスウェーデン語の言語ソート、多言語の GENERIC_M 言語ソートの違いを示します。

例 1. バイナリ・ソート

```
ALTER SESSION SET NLS_SORT=BINARY;

SELECT product_name
FROM product
ORDER BY product_name;

PRODUCT NAME
------------
Antenne
Lcd
aerial
```

```
Ähre
ächzen
```

例 2. 単言語のスウェーデン語のソート

```
ALTER SESSION SET NLS_SORT= SWEDISH;

SELECT product_name
FROM product
ORDER BY product_name;

PRODUCT NAME
------------
aerial
Antenne
Lcd
ächzen
Ähre
```

例 3. 多言語の GENERIC_M ソート

```
SELECT product_name
FROM product
ORDER BY NLSSORT (product_name, 'NLS_SORT=GENERIC_M');

PRODUCT NAME
------------
ächzen
aerial
Ähre
Antenne
Lcd
```

比較演算子を使用すると、文字は、指定されるコード体系のバイナリ・コードに従って比較されます。バイナリ・コードが高いほど、文字は大きくなります。文字のバイナリ順が、特定の言語の言語シーケンスと一致しないことがあるため、そのような比較は「言語学的に正確」ではない可能性があります。SQL の NLSSORT は、言語の表記規則を反映するために、そのような比較を可能にする関数です。

例 4. BINARY 比較

```
ALTER SESSION SET NLS_SORT=GERMAN;

SELECT product_name
FROM product
WHERE product_name > 'Antenne'
ORDER BY product_name;

PRODUCT NAME
------------
ächzen
aerial
ÄhLre
Lcd
```

例 5. 言語に依存する比較 GERMAN

SQL の NLSSORT 関数は、比較演算子の両側に追加する必要があります。

```
ALTER SESSION SET NLS_SORT=GERMAN;

SELECT product_name
FROM product
WHERE NLSSORT (product_name) >
      NLSSORT ('Antenne')
ORDER BY product_name;

PRODUCT NAME
------------
Lcd
```

SQL 文で NLSSORT 関数を使用すると、複雑になる場合があります。NLS_SORT セッション・パラメータに従って、比較が言語に依存する必要があることを示すために、新しい NLS パラメータ NLS_COMP が、Oracle8*i* に導入されました。

Parameter type	String
Syntax	NLS_COMP = {BINARY | ANSI}
Default value	BINARY
Parameter scope	Initialization Parameter, Environment Variable and ALTER SESSION
Range of values	BINARY or ANSI

たとえば、***前述の例 5. 言語に依存する比較 GERMAN*** は、次のように書き換えることができます。

```
ALTER SESSION SET NLS_SORT =GERMAN;
ALTER SESSION SET NLS_COMP=ANSI;

SELECT product_name
FROM product
WHERE product_name > 'Antenne'
ORDER BY product_name;

PRODUCT NAME
------------
Lcd
```

注意: NLS_COMP と NLS_SORT は、次の SQL 操作のみに影響を与えます。

WHERE、ORDER BY、START WITH、IN/NOT IN、BETWEEN、CASE WHEN、HAVING。

他の SQL 演算子は、バイナリ・モードのみで比較します。言語に依存する比較を有効にするには、これらの演算子に NLSSORT 関数を追加する必要があります。

言語索引の使用

言語ソートは、言語によって異なり、バイナリ・ソート以上のデータ処理を必要とします。バイナリ・ソートはキャラクタ・セットのエンコーディングの順に行われるため、高速です。データベースに多言語のデータが格納されている場合、使用言語に基づく言語シーケンスで、言語ソートに伴うパフォーマンスの低下を引き起こすことなく、ORDER BY 句を使用して SELECT 文から返される結果セットをアプリケーションに照合させることをお薦めします。これは、言語索引を使用することで実行できます。言語索引は、ファンクション・ベース索引のバリエーションです。

使用する言語のデータをソートするための言語索引のセットアップには、3 つのアプローチがあります。

1. アプリケーションがサポートする必要のある各言語に言語索引を構築します。このアプローチは単純ですが、大きなディスク領域が必要です。各索引について、索引が構築される言語以外の言語の行は、シーケンスの最後で一括して照合されます。次の例で、フランス語とドイツ語のデータをソートするための言語索引を構築します。

   ```
   CREATE INDEX french_index ON product
   (NLSSORT(product_name, 'NLS_SORT=FRENCH'));

   CREATE INDEX german_index ON product
   (NLSSORT(product_name, 'NLS_SORT=GERMAN'));
   ```

   NLS_SORT セッション・パラメータまたは ORDER BY 句で指定される NLSSORT 関数の引数に基づき、索引は SQL オプティマイザによって選択されます。たとえば、セッション変数 NLS_SORT が FRENCH に設定されていると、french_index が選択されます。これが GERMAN に設定されている場合は、german_index が選択されます。

2. GENERIC_M や FRENCH_M などの多言語ソートを使用して、すべての言語に対して単一の言語索引を構築します。この索引は、ISO 14651 規格で定義されている文字のルールに従い文字を照合します。

   ```
   CREATE INDEX generic_index on product
   (NLSSORT(product_name, 'NLS_SORT=GENERIC_M');
   ```

   NLS_SORT セッション・パラメータまたは ORDER BY 句で指定した NLSSORT 関数の引数が、索引の定義で使用されたソート名と同じ場合は、索引は自動的に取得されます。サポートする必要のある言語が、特定の多言語ソートに含まれている場合、このアプローチは便利です。

索引を作成する場合、索引キーの長さは、一定の値を超えることができません。この値は、主として DB_BLOCK_SIZE によって決まります。最大値よりも大きいキーを持つ索引を作成しようとすると、「ORA-1450 maximum key length exceeded」エラーが発生します。

2K ブロックの索引キーの許容最大長は 758、4K ブロックの場合は 1578、8K ブロックの場合は 3218、16K の場合は 6498 です。

3. すべての言語に対して単一の言語索引を構築します。これは、NLSSORT 関数のパラメータとして使用されるように、言語の列を表（次の SQL の例に示す LANG_COL）に含めることによって実行できます。言語の列には、索引が構築される列のデータの NLS SORT 値が含まれます。次の例で、複数言語のための単一の言語索引を構築します。この索引を使用すると、NLS_SORT が同じ値になる列は、相関的に正しく照合されます。異なる値を持つ列は、有意義に比較できません。

```
CREATE INDEX nls_index ON product
(NLSSORT(product_name, 'NLS_SORT=' || LANG_COL));
```

nls 索引は、問合せが `ORDER BY` 句で明示的に `NLSSORT（product_name, 'NLS_SORT=' || LANG_COL）`を指定する場合に限り使用されます。

他のファンクション・ベース索引と同様に、複合言語索引の構築もサポートされます。

例を示します。

```
CREATE INDEX german_index ON product
(NLSSORT(product_name, 'NLS_SORT=GERMAN'),
 NLSSORT(company_name, 'NLS_SORT=GERMAN'));
```

実際に、接頭辞が機能しない複合言語索引が存在する場合、つまり、前述の索引が変更されて、モデル番号を含むようになった場合、ルールベースのオプティマイザは機能しない接頭辞を使用できます。

```
CREATE INDEX german_index ON product
(model_number, NLSSORT(product_name, 'NLS_SORT=GERMAN'),
 NLSSORT(company_name, 'NLS_SORT=GERMAN'));
```

その場合、ルールベースの問合せは、この複合言語索引も利用できます。

言語索引使用の要件

1 つの言語索引の使用、複数の言語索引の使用のいずれの場合も、言語索引を使用するための要件を満たす必要があります。

1. QUERY_REWRITE_ENABLED セッション・パラメータを TRUE に設定します。
2. QUERY_REWRITE_INTEGRITY=TRUSTED を設定、またはそれ以上に設定します。
3. COMPATIBLE フラグが 8.1.6 またはそれ以上に設定されていることを確認します。

   この 3 つの初期化パラメータ設定は、すべてのファンクション・ベース索引に必要です。

4. NLS_SORT を正しく設定します。

   4 番目の要件は、問合せの NLS_SORT パラメータが、索引の CREATE 文で指定された言語の定義を示す必要があることです。それは、暗黙的に指定できます（定数である場合）。または NLSSORT 関数の 2 番目の引数として直接指定できます。

5. オプティマイザのモードを FIRST_ROWS に設定します。

   ファンクション・ベース索引の場合は、ルールベースのオプティマイザが認識しないため、モードが FIRST_ROWS に設定されているコストベースのオプティマイザを使用します。

Oracle 9*i* Database Release 2 以前のデータベースを使用する顧客のために、特別なダミーの WHERE 句を指定しファンクション・ベース索引をトリガーする必要もあります。

```
WHERE NLSSORT(column_name) IS NOT NULL
```

column_name が言語索引を持つ列である場合は、ORDER BY column_name を使用します。これは、ORDER BY 句を使用する場合にのみ必要です。column_name が NOT NULL 列として定義済みの場合、ダミーの WHERE 句は Oracle9*i* Release 2 以降では必要ありません。

次の例に、PRODUCT 表で多言語ソート GENERIC_M に基づき、NLS_GENERIC という言語索引を作成する方法を示します。

```
ALTER SESSION SET QUERY_REWRITE_ENABLED=TRUE;
ALTER SESSION SET QUERY_REWRITE_INTEGRITY=TRUSTED;

CREATE INDEX NLS_GENERIC ON product
(NLSSORT(product_name, 'NLS_SORT=GENERIC_M'));

ALTER SESSION SET NLS_SORT=GENERIC_M;
ALTER SESSION SET OPTIMIZER_MODE=FIRST_ROWS;
```

すべての言語索引の要件が満たされている場合でも、低コストのプランが使用可能であるという理由で、オプティマイザが言語索引の使用を選択しない可能性があります。

ヒント/*+ index(table indexname) */を追加すると、コストベースのオプティマイザが使用され、正当な問合せ計画を可能にする場合は、その索引が使用されます。

```
SELECT * FROM product
WHERE NLSSORT (product_name) IS NOT NULL - Not needed in Oracle9i
Release 2 and onward
ORDER BY product_name;
```

product_name が NOT NULL 列として定義済みの場合、ダミーの WHERE 句は Oracle9*i* Release 2 以降では必要ありません。

大文字/小文字を区別しない検索

SQL の NLS_UPPER および NLS_LOWER 関数は、特定の言語ソートの定義に基づき、言語に依存する文字列のケーシングを実行します。これにより、使用されている言語とは無関係に、大文字/小文字を区別しない検索が可能になります。

```
SELECT product_name
FROM product
WHERE NLS_UPPER(product_name, 'NLS_SORT = XGERMAN') = 'GROSSE';

PRODUCT NAME
------------
groβe
Groβe
GROSSE
```

言語ソートの場合と同様に、大文字/小文字を区別しない検索のパフォーマンスを改善するために、ファンクション・ベース索引を構築できます。例を示します。

```
CREATE INDEX case_insensitive_index ON product
(NLS_UPPER(product_name));
```

この索引は、文字列の比較 NLS_UPPER( )が列の product_name に対して実行されるごとに使用されます。

```
SELECT * FROM PRODUCT
WHERE NLS_UPPER(product_name) = 'GROSSE';
```

GENERIC_BASELETTER ソート

大文字/小文字を区別しない問合せを実行するために、SQL NLS_UPPER 関数と NLS_LOWER 関数を使用する代わりに、Oracle9*i* Release 2 で代替のアプローチが初めて導入されました。これは言語ソート GENERIC_BASELETTER を使用します。GENERIC_BASELETTER は、基本文字の値に基づき、すべての文字をグループ化します。これは、大文字/小文字および発音区別記号の違いを無視することによって実行されます。

GENERIC_BASELETTER 問合せの例を次に示します。

GENERIC_BASELETTER 検索は、言語に依存する検索ではなく、一定の言語に基づきません。このソートは基盤となる文字の基本文字のみを定義します。したがって、大文字/小文字を区別しない、アクセント記号に依存しないソートの動作をシミュレートします。

```
ALTER SESSION SET NLS_COMP=ANSI;
ALTER SESSION SET NLS_SORT=GENERIC_BASELETTER;

SELECT * FROM PRODUCT
WHERE PRODUCT_NAME = 'database';

DATABASE
Database
database
dätäbase
```

Oracle Database 10g の大文字/小文字を区別しない、アクセント記号に依存しない新しい検索機能

Oracle Database 内の演算は、常に大文字/小文字を区別し、文字のアクセント記号（発音区別記号）に依存しますが、場合によっては、大文字/小文字を区別しない、アクセント記号に依存しない比較やソートを行う必要があります。以前のリリースでは、SQL 文が大文字/小文字を区別しないようにするために、関数 LOWER/UPPER を呼び出すか、GENERIC_BASELETTER ソートを使用しました。

LOWER/UPPER 関数の使用は、パフォーマンス低下の原因となります。これらの関数は文字列全体を大文字または小文字に変換し比較します。
GENERIC_BASELETTER ソートは、これらの問題を解決しますが、一般的なソートで、特別な言語またはロケールのためにデザインされていないため、このソート自体には制限があります。大文字/小文字を区別しない、アクセント記号に依存しない新しい検索機能は、あらゆる言語ソートに適用できパフォーマンスを低下させることがありません。さらに顧客が、既存のコードを変更することなく、同じ SQL の動作を使用できるために一挙両得の検索機能です。

大文字/小文字を区別しないソート、またはアクセント記号に依存しないソートの指定

大文字/小文字を区別しないソート、またはアクセント記号に依存しないソートを指定するには、NLS_SORT セッション・パラメータを使用します。

- 大文字/小文字を区別しないソートのためには、Oracle のソート名に_CI を追加します。
- アクセント記号に依存しない、大文字/小文字を区別しないソートのためには、Oracle のソート名に_AI を追加します。

構文:

- `NLS_SORT = <NLS_sort_name>[_AI | _cl]`

構文の例:

- NLS_SORT = JAPANESE_M /*アクセント記号に依存し、大文字/小文字を区別する Japanese_M ソート*/
- NLS_SORT = FRENCH_M_AI /*アクセント記号に依存しない、大文字/小文字を区別しない French_M ソート*/NLS_SORT = XGERMAN_CI /*アクセント記号に依存しない、大文字/小文字を区別しない XGerman ソート*/NLS_SORT = BINARY_CI /*アクセント記号に依存しない、大文字/小文字を区別しないバイナリ・ソート*/
- NLS_SORT = BINARY_AI /***アクセント記号に依存しない、大文字/小文字を区別しないバイナリ・ソート***/

例 1: 大文字/小文字を区別しない、アクセント記号に依存しないバイナリ・ソート

次のデータを持つ列 LETTER を例に説明します。

LETTER

ä
a
A
Z

NLS_SORT の値を変更するには、次の文に類似した文を入力します。

ALTER SESSION SET NLS_SORT=BINARY_CI;

次の表に、NLS_SORT を BINARY、BINARY_CI、BINARY_AI に設定することから生じるソート順を示します。

1) BINARY	2) BINARY_CI	3) BINARY_AI
A	a	ä
Z	A	a
a	Z	A
ä	ä	Z

1. バイナリ・ソートを実行した結果を比較します。大文字が小文字の前に配置されます。発音区別記号が付いた文字は最後に表示されます。
2. ソートが発音区別記号を考慮し、大文字/小文字の区別を無視する（BINARY_CI）場合、発音区別記号が付いた文字は最後に表示されます。
3. 大文字/小文字の区別と発音区別記号の両方が無視される（BINARY_AI）場合、ä は、基本文字が a である他の文字とともにソートされます。基本文字が a である字はすべて z の前に配置されます。

例 2: 大文字/小文字を区別しない、アクセント記号に依存しないドイツ語のソート

同じデータを持ち、NLS_SORT が GERMAN に設定されている列 LETTER を例に説明します。

次の表に、NLS_SORT を GERMAN、GERMAN_CI、GERMAN_AI に設定することから生じるソート順を示します。

GERMAN	GERMAN_CI	GERMAN_AI
a	a	ä
A	A	a
ä	ä	A
Z	Z	Z

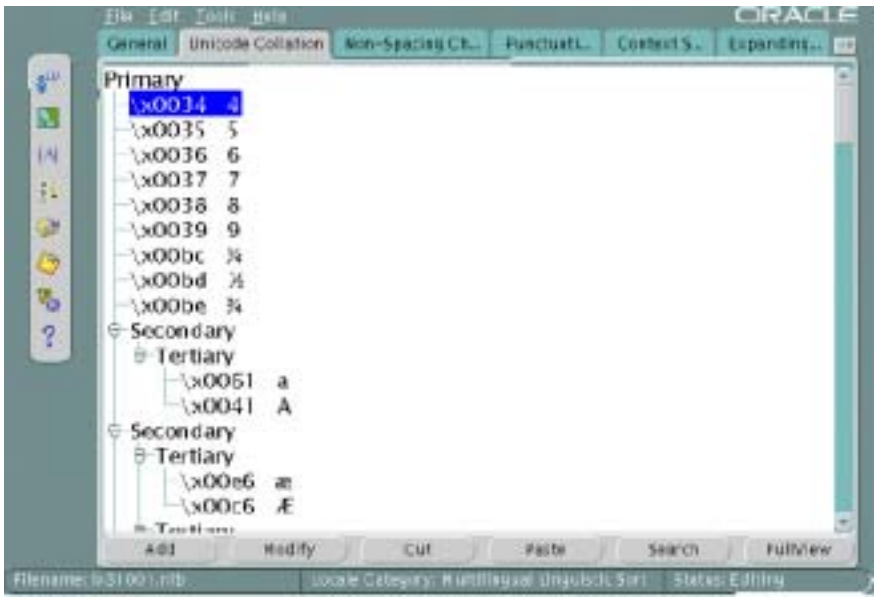
1. ドイツ語のソートでは、小文字が大文字の前に配置され、ä は Z の前に来ます。
2. ソートが大文字/小文字の区別と発音区別記号の両方を無視する（GERMAN_AI）場合、ä は基本文字が a である他の文字といっしょに表示されます。

言語ソートのカスタマイズ

様々な言語ニーズが増大している顧客の要求に応えるために、多くの広範囲にわたる言語ソートが準備されています。ただし、文化的背景が変化し、新しい業界基準（ISO や Unicode 規格など）が次々に出現する結果として、新しいソート要件は常に存在します。場合によっては、種々のプラットフォームで使用される他のソフトウェア製品との互換のために、他のベンダーが採用したアプローチに一致する方法で、言語ソートをカスタマイズする必要があります。

ロケール・データ定義を構成するために Oracle9*i* で導入された GUI ツールとして、Oracle Locale Builder があります。ユーザーは、このツールの使いやすいグラフィカル・インタフェースを使用して、簡単に既存の Oracle ロケール定義の表示やカスタマイズ、あるいは新しい定義が作成できます。言語ソートの定義について、Locale Builder はソート順を図式表現します。これは表示あるいはカスタマイズの場合にも、直観的に使用できます。Oracle Locale Builder は、文字を異なった順序に再配列することをサポートし、文字の縮小および拡大のカスタマイズ、状況依存の文字、追加文字を提供します。

Oracle Locale Builder のスクリーンショット



詳細は、『Oracle Database グローバリゼーション・サポート・ガイド』を参照してください。

まとめ

グローバルにデプロイされたアプリケーションでは、言語に依存した方法でデータをソートすることは、データ処理の重要な部分です。ユーザーの日常言語の順序でソートされていないデータが提示された場合、ユーザーにとって情報の検索は難しく、時間のかかる作業になります。

Oracle は、バイナリ・ソートと言語ソートの適用範囲を拡大するとともに、複数の言語でデータを検索およびソートする必要のある顧客の要求に応えるために、新しい多言語ソートを導入しました。Oracle は今後も言語ソートの適用範囲を拡大し、ISO 14651 – International String Ordering 規格と Unicode Collation Standard（UCA）をサポートすることによって国際標準に準拠します。

Oracle は、広範囲にわたる言語ソートを提供し、60 以上の単言語ソートと 13 の多言語ソートをサポートします。Oracle が提供する様々な言語ソートを超える特殊な要件も、Oracle Locale Builder を使用することによって、顧客は、言語ソートをカスタマイズし、独自の言語ソートを定義するといった柔軟な処理ができます。Oracle Locale Builder は、既存の言語ソートの表示や新しい言語ソートの作成を可能にする、簡単に使用できる新しい GUI ツールです。

Oracle Database 内での言語データのソート
2003 年 9 月
著書:　Simon Law
寄稿者:　Sergiusz Wolicki, Claire Ho, Barry Trute

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

海外からのお問合せ窓口:
電話:　+1.650.506.7000
ファックス:　+1.650.506.7200
www.oracle.com

オラクル社は、インターネット上での活動を強化するソフトウェアを提供します。