

CONTAINS問合せ演算子(抜粋)

論理演算子

AND OR ACCUM
NOT MINUS MNOT

~~ワイルドカード~~

~~% _~~

スコア関連

* >

近傍

NEAR

等価

EQUIV

セクション

WITHIN
INPATH HASPATH
MDATA SDATA
NDATA

シソーラス

SYN BT NT

エスケープ文字

¥ { }

AND (&)

- 構文

- ***term1* & *term2*** ←「&」の前後に半角スペースはあってもなくてもいい
- ***term1* and *term2*** ←「AND」、「And」、「aNd」等いずれも可

→ *term1* と *term2* の両方が含まれているドキュメントが返る

- スコア

- 指定されたキーワードのうち、最小のスコアが返る
- 例えば、「blue & black & red」という検索条件で、各キーワードのスコアが 10、20、30 であると仮定すると、全体のスコアは 10 になる

```
SELECT id FROM testtab  
WHERE CONTAINS (text, 'オラクル and 日本') > 0;
```

OR (|)

- 構文

- ***term1 | term2*** ←「|」の前後に半角スペースはあってもなくてもいい
- ***term1 or term2*** ←「OR」、「Or」、「oR」等のいずれも可

→ *term1* と *term2* のいずれかが含まれているドキュメントが返る

- スコア

- 指定されたキーワードのうち、最大のスコアが返る
- 例えば、「cats | dogs」という検索条件で、各キーワードのスコアが 30、40 であると仮定すると、全体のスコアは 40 になる

```
SELECT id FROM testtab
WHERE CONTAINS (text, 'オラクル or 日本') > 0;
```

ACCUMulate(,)

※ accumulate = 累積する

- 構文

- ***term1,term2*** ←「,」の前後に半角スペースはあってもなくてもいい
- ***term1 accum term2*** ←「ACCUM」、「Accum」等のいずれも可

→ *term1* と *term2* のいずれかが含まれているドキュメントが返る

- スコア

- 指定されたキーワードの全てが含まれているものに高いスコアが割り当てられる
- スコア以外の部分は、OR演算子と同じ動作

```
SELECT id FROM testtab
WHERE CONTAINS (text, 'オラクル accum 日本') > 0;
```

NOT(~)

- 構文

- ***term1~term2*** ←「~」の前後に半角スペースはあってもなくてもいい

- ***term1 not term2*** ←「NOT」、「Not」、「nOt」等のいずれも可

→ *term1*が含まれるが、*term2*が含まれていないドキュメントが返る

- スコア

- NOT演算子は、他の論理演算子が作成したスコアには影響を与えない

```
SELECT id FROM testtab
WHERE CONTAINS (text, 'オラクル ~日本') > 0;
```

MINUS

- 構文
 - ***term1-term2*** ←「-」の前後に半角スペースはあってもなくてもいい
 - ***term1* minus *term2*** ←「MINUS」、「Minus」、「miNus」等のいずれも可
- *term1*が含まれているドキュメントが返る。ただし、*term1*のスコアから *term2*のスコアを引いてスコアを計算し、正数のスコアを持つドキュメントのみが返る
- スコア
 - *term1*のスコアから*term2*のスコアを引いてスコアを計算する

```
SELECT id FROM testtab
WHERE CONTAINS (text, 'オラクル -日本') > 0;
```

MNOT

※ MNOT=Mild Not

(Oracle Database 11g Release 2(11.2)以降)

- 構文(使用例)
 - ***term1* mnot *term1 term2***
→ *term1*が含まれ、かつフレーズ *term1 term2* が含まれないドキュメントが返る(例えば「mexico MNOT new mexico」のように使う)
 - ***term1* mnot *term2***
→ *term1*が含まれるドキュメントが返る(単に「*term1*」と指定した場合と同じ動作)
- スコア
 - MNOTの左側に書かれた条件に一致し、かつ右側に書かれた条件に一致しない部分のみがスコアの加算対象となる
 - 例えば「*term1 term1 term2*」という文字列を含む文書を「*term1 mnot term1 term2*」で検索した場合、この文書はヒットするが、スコアは最初の*term1*に対してのみ計算される

ワイルドカード(「%」、「_」)

- 2種類のワイルドカード

ワイルドカード文字	説明
%	任意のn文字との一致
_	任意の1文字との一致

- 英数字検索では、ワイルドカード文字を利用する

- たとえば、文字列「Oracle」を含むドキュメントは、「WHERE CONTAINS (列名, '%ac%') > 0」という条件でヒットする

- 日本語検索では、ワイルドカード文字を付けてはいけない

- たとえば、文字列「オラクル」を含むドキュメントは、「WHERE CONTAINS (列名, '%オラクル%') > 0」のように、ワイルドカード文字を付けて検索してもヒットしない。「WHERE CONTAINS (列名, 'オラクル') > 0」のように、ワイルドカード文字を付けない状態で検索をして初めて正常にヒットする

近傍検索(**NEAR**)

- 2つ以上の検索キーワードの出現位置の近さを基準にしたスコアが戻される
- ドキュメント内で互いに近接したキーワードには高いスコアが、互いに離れたキーワードには低いスコアが戻される

近傍検索 (NEAR)

- 構文

```
NEAR((word 1, word 2,..., word n) [, max_span [, order]])
```

- *word 1* ~ *word n*
問合せ語句をカンマ区切りで指定
- max_span
上で指定した語句が何トークン以内に出現するかを、100以下の整数で指定 (デフォルトは100)
- order
上で指定した語句が、指定した順序どおりで出現しているものを探すにはTRUE、そうでない場合はFALSEを指定 (デフォルトはFALSE)

近傍検索(**NEAR**)使用例

- dogとcatが6ワード以内にある文書を検索するには...

```
SELECT id FROM testtab  
WHERE CONTAINS (text, 'NEAR((dog, cat), 6)') > 0;
```

- 3つの検索キーワード Monday、Tuesday、Wednesday
が、20ワード以内に、この順で出現している文書を検索
するには...

```
SELECT id FROM testtab  
WHERE CONTAINS (text,  
'NEAR((Monday, Tuesday, Wednesday), 20, TRUE)') > 0;
```

近傍検索(**NEAR**)使用例

- 他の演算子との組合せ例

```
SELECT id FROM testtab  
WHERE CONTAINS (text,  
  'NEAR((lion, tiger), 10) AND cheetah') > 0;
```

等価検索(EQUIV)

- 検索時に条件を満たすワードの置換を指定する
- 構文

- *term1=term2*
- *term1 equiv term2*

→ *term1* と *term2* が互いに置換語として指定される

- たとえば、

German shepherds=alsatians are big dogs

という検索条件は、「alsatians are big dogs」を含む文書と「German shepherds are big dogs」を含む文書の両方にヒットする

重み付け(*) WEIGHT

- 指定した係数をスコアに掛ける
- たとえば、問合せ「cat, dog*2」は、「catのスコア」と「dogのスコアの2倍」をベースに、全体のスコアが計算される
- スコアの最大値は100
- 構文

- $term * n$ ※ n は、0.1～10の数値

→ $term$ のスコアに n を掛けてスコアが計算される

- Tips: スコアに10より大きい数字を掛けるには？

$(term1 * 10) * 10$ or $term2$

→ これで、 $term1$ を含む文書のスコアは必ず100になる

重み付け(*)使用例

WEIGHT

- 前提
 - スポーツ記事の収集で、ブラジルのサッカーについての記事に関心があると仮定
 - 「soccer or Brazil」で通常の問合せをすると、USサッカーに関する多くの記事が高いランクで戻ってしまう
- このとき、ブラジルのサッカーについての記事のランクを上げるには・・・？

→ 例えば、次の問合せを実行する

soccer or Brazil*3

→ このときのスコアの変化(次ページ)

重み付け(*)使用例

WEIGHT

- 次の表は、重み付け演算子によって、サッカーの情報が含まれている3つのドキュメントA、B、Cのランクが変更されることを示している

スコア・サンプル

	soccer	Brazil	soccer or Brazil	soccer or Brazil*3
A	20	10	20	30
B	10	30	30	90
C	50	20	50	60

しきい値 (>) THRESHOLD

- しきい値の数値を下回るスコアのドキュメントを、検索結果から排除する
- 問合せ語句レベル、式レベルのいずれでも使用可能
- 構文
 - 問合せ語句レベル
 $term > n$
→ 式内で、スコアが n より大きい問合せ語句が含まれているドキュメントが返される
 - 式レベル
 $expression > n$
→ 結果セットで、スコアがしきい値 n を超えるドキュメントが返される

しきい値 (>) 使用例

THRESHOLD

- 問合せ語句レベル

- lionのスコアが30よりも大きく、さらにtigerが含まれているドキュメントを検索するには・・・？

```
SELECT id FROM testtab  
WHERE CONTAINS (text, '(lion > 30) and tiger') > 0;
```

- 式レベル

- relational databasesが含まれているドキュメントを検索し、スコアが75よりも大きいドキュメントのみを検索するには・・・？

```
SELECT id FROM testtab  
WHERE CONTAINS(text, 'relational databases > 75')>0;
```

※ 式レベルでのしきい値指定は、CONTAINS関数の返り値を使って、
「WHERE CONTAINS(col, 'relational databases') > 75」のように指定してはいけない。
必ず「WHERE CONTAINS(col, 'relational databases > 75') > 0」の形で指定する。