

AIDE - recherche et assistance

Requêtes avec caractères de remplacement

Utilisez les caractères de remplacement avec modération, car ce procédé réduit les performances des serveurs. Vous pouvez utiliser ces caractères de remplacement dans des requêtes :

- ? pour remplacer un caractère.
- * pour remplacer zéro, un caractère ou plusieurs caractères.

Remarque : N'utilisez pas de caractères de remplacement dans une recherche d'expression exacte.

Recherche d'expression exacte

Si vous soumettez une expression entre guillemets, le serveur recherche l'expression dans sa forme exacte, sans tenir compte des racines. Par exemple :

“fraîche et charmante”

Le serveur supprime tout mot vide contenu dans la requête (l'exemple de requête ci-dessus contient le mot vide *et*) mais ne tient pas compte des racines. C'est comme si la requête était :

“fraîche charmante”

Lors de la recherche, le serveur produit uniquement des documents qui contiennent une expression correspondant à l'expression de la recherche. La requête « fraîche et charmante » fournit uniquement des documents qui contiennent une expression correspondant à *fraîche charmante* (par exemple, *fraîche charmante*, *fraîche et charmante*, *fraîche ou charmante*, etc.)

Opérateurs de recherche booléens

Opérateur	Explication
Sans opérateur	L'opérateur par défaut est AND si aucun autre n'est précisé. Fait en sorte que tous les documents produits contiennent les deux termes. Par exemple : chien chat Cette requête produit uniquement des documents qui contiennent à la fois les termes chat et chien.
AND	Opérateur binaire. Fait en sorte que tous les documents produits contiennent les deux termes. Il s'agit du comportement par défaut si aucun opérateur n'est explicitement indiqué entre les deux termes. Par exemple : chien AND chat Cette requête produit uniquement des documents qui contiennent à la fois les termes <i>chat</i> et <i>chien</i> .
NOT	Opérateur monadique. Fait en sorte que le terme suivant NOT est exclu de tout document signalé. Par exemple : chat NOT chien Cette requête produit uniquement des documents contenant chat mais pas chien. <u>Remarque</u> : NOT s'applique uniquement au terme qui le suit immédiatement. Pour exclure plusieurs termes, placez ces termes entre parenthèses. Pour exclure une expression, placez l'expression entre guillemets et entre parenthèses. Par exemple : Doc 1 : Je suis allée à Old York Town pour le nouvel an. Doc 2 : Je suis allée à New York City pour le nouvel an. La requête suivante ne produit aucun de ces documents : nouvel NOT (Town OR City) La requête suivante produit les deux documents : nouvel NOT (Town AND City) La requête suivante produit le premier document mais pas le second : nouvel NOT ("New York")
OR	Opérateur binaire. L'un des deux termes, ou les deux, doivent figurer dans le document pour que celui-ci soit signalé. Par exemple : chat OU chien Cette requête produit uniquement les documents qui contiennent <i>chat</i> , <i>chien</i> ou les deux termes.
EOR ou XOR	Opérateur binaire. Opérateur logique exclusif OR. Seul l'un des termes peut figurer dans le document pour que celui-ci soit signalé. Cet opérateur est rarement utilisé. Par exemple : chat XOR chien Cette requête produit uniquement les documents qui contiennent soit le terme <i>chat</i> soit le terme <i>chien</i> . Les documents qui contiennent à la fois <i>chat</i> et <i>chien</i> ne sont pas signalés.
()	Expressions entre parenthèses. Ces expressions sont évaluées de gauche à droite et peuvent être imbriquées. Elles dictent la priorité et le comportement des énoncés d'opérateurs combinés. Par exemple : (chat EOR chien) AND (poisson EOR oiseau) Cette requête produit uniquement des documents contenant l'une de ces combinaisons : <i>chat</i> et poisson <i>chat</i> et <i>oiseau</i> <i>chien</i> et <i>poisson</i> <i>chien</i> et <i>oiseau</i>

Opérateurs de recherche de proximité

Opérateur	Explication
NEARN	<p>Produit uniquement les documents dans lesquels le deuxième terme se trouve à N mots du premier terme; c'est à-dire, N mots ou moins séparent les deux termes. Si vous ne précisez pas N, NEAR a la valeur 5 par défaut. Par exemple :</p> <p style="padding-left: 40px;">rouge NEAR1 vert</p> <p>Cette requête produit uniquement les documents dans lesquels le terme <i>rouge</i> se trouve immédiatement adjacent au terme <i>vert</i>; par exemple, les documents qui contiennent <i>rouge vert</i> ou <i>vert rouge</i>. Les documents qui contiennent <i>rouge orange vert</i> ne sont pas signalés (parce que les termes ne sont pas assez proches).</p>
DNEARN	<p>NEAR dirigé. Produit uniquement les documents dans lesquels le second terme se trouve à N mots du premier, dans l'ordre précisé. Si vous ne précisez pas N, DNEAR a la valeur 5 par défaut. Par exemple :</p> <p style="padding-left: 40px;">rouge DNEAR2 vert</p> <p>Cette requête produit uniquement les documents dans lesquels le terme <i>vert</i> suit le terme <i>rouge</i> et les deux termes ne sont pas séparés par plus de deux mots. Par exemple, les documents qui contiennent <i>rouge orange vert</i> sont produits, alors que ceux qui contiennent <i>vert orange rouge</i> ou <i>rouge orange bleu vert</i> ne sont pas indiqués.</p>
WNEARN	<p>NEAR pondéré (avec opérateur OR). Cet opérateur de proximité produit les documents qui contiennent l'un des deux termes. Il applique une priorité lorsque les termes sont séparés par N mots ou moins (plus les deux termes sont rapprochés, plus la pertinence est grande). Si vous ne précisez pas N, WNEAR prend la valeur 5 par défaut. Par exemple :</p> <p style="padding-left: 40px;">chien WNEAR7 chat</p> <p>Cette requête produit des documents qui contiennent soit <i>chien</i> soit <i>chat</i>. Elle donne la priorité aux documents où <i>chien</i> et <i>chat</i> sont séparés de sept mots ou moins dans le texte. Cette pondération augmente à mesure que les termes se rapprochent. Les documents dans lesquels les termes sont séparés de plus de sept mots où dans lesquels seul l'un des termes figure, sont signalés avec une pertinence normale.</p>
YNEARN	<p>NEAR pondéré (avec opérateur AND). Cet opérateur de proximité produit les documents qui contiennent les deux termes. Il applique une priorité lorsque les termes sont séparés par N mots ou moins (plus les deux termes sont rapprochés, plus la pertinence est grande). Si vous ne précisez pas N, NEAR a la valeur 5 par défaut. Par exemple :</p> <p style="padding-left: 40px;">chien YNEAR7 chat</p> <p>Cette requête produit des documents qui contiennent à la fois <i>chien</i> et <i>chat</i>. Elle donne la priorité aux documents où <i>chien</i> et <i>chat</i> sont séparés par sept mots ou moins dans le texte. Cette pondération augmente à mesure que les termes se rapprochent. Les documents dans lesquels les termes sont séparés par plus de sept mots sont signalés avec une pertinence normale.</p>
BEFORE	<p>Produit uniquement les documents dans lesquels le premier terme précède le second. Par exemple :</p> <p style="padding-left: 40px;">rouge BEFORE vert</p> <p>Cette requête produit uniquement les documents dans lesquels le terme vert se trouve après le terme rouge.</p>
AFTER	<p>Produit uniquement les documents dans lesquels le premier terme suit le second. Par exemple :</p> <p style="padding-left: 40px;">rouge AFTER vert</p> <p>Cette requête produit uniquement les documents dans lesquels le terme rouge figure après le terme vert.</p>
XNEAR	<p>Produit uniquement les documents dans lesquels le second terme se trouve à exactement N mots du premier. Par exemple :</p> <p style="padding-left: 40px;">chats XNEAR2 chiens</p> <p>Cette requête produit uniquement les documents dans lesquels le terme chiens suit le terme chats et se trouve à exactement deux mots du terme chats. Cela signifie que les documents qui contiennent chats et chiens sont signalés, alors que les documents qui contiennent chiens et chats ou chats, chiens ne le sont pas.</p>