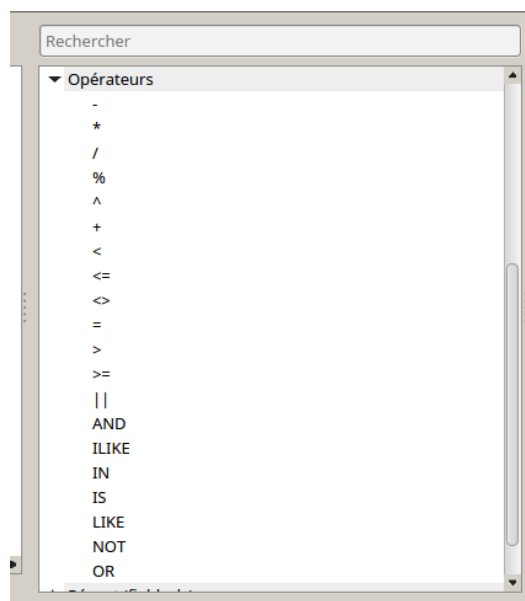


# Les opérateurs

Dans la calculatrice de champs, dans les sélections par expressions et toutes les fois que vous voyez



Les exemples sont donnés sur la base du fichier GEOFLA®

Les exemples sont données sur la base du fichier GEOTEXA																									
CODINSEE	NOM_COM	C.50	STATUT	C.25	X	CHF	Y	CHF	X	CENT	Y	CENT	Z	MO	SUPER	POPUL	C	CO	NOM	DEPT	C.30	CO	NOM	REG	C.35
216	32216	LOURTIES-MONBRUN	Commune simple		500820	6264958	500615	6265413	252	966	139	3	32	76											LANGUEDOC-ROUSSILLON-MIDI-PYRE
033	47033	BOUDY-DE-BEAUREGARD	Commune simple		516424	6384852	515575	6385938	112	1019	413	4	47	75											AQUITAINE-LIMOUSIN-POITOU-CHARE
009	32009	ARMOUS-ET-CAU	Commune simple		472979	6279863	473004	6278937	221	932	95	3	32	76											LANGUEDOC-ROUSSILLON-MIDI-PYRE
225	38225	AUTRANS-MEAUDRE EN VERCORS	Commune simple		898640	6450689	898625	6451597	1234	3371	2973	1	38	84											Auvergne-Rhone-Alpes
890	62890	WILLEMAN	Commune simple		640049	7028672	640115	7029900	79	1023	178	4	62	32											NORD-PAS-DE-CALAIS-PICARDIE
018	08018	ARDEUIL-ET-MONTFAUXELLES	Commune simple		824246	6908952	824391	6908954	125	438	80	4	08	44											ALSACE-CHAMPAGNE-ARDENNE-LOR
113	32113	CRAVENCERES	Commune simple		461332	6300782	460721	6302268	134	919	97	2	32	76											LANGUEDOC-ROUSSILLON-MIDI-PYRE
319	10319	RIGNY-LE-FERON	Commune simple		746925	6790005	747181	6789569	167	1904	362	3	10	44											ALSACE-CHAMPAGNE-ARDENNE-LOR
097	06097	PIERREFEU	Commune simple		1028827	6315717	1027327	6316879	752	2217	296	2	06	93											PROVENCE-ALPES-COTE D'AZUR
070	42070	CORDELLE	Commune simple		782215	6538794	782159	6538837	438	2667	901	2	42	84											Auvergne-Rhone-Alpes
046	31046	BAREN	Commune simple		505288	6199546	506524	6199445	1276	306	10	2	31	76											LANGUEDOC-ROUSSILLON-MIDI-PYRE
524	71524	SIVIGNON	Commune simple		815144	6592954	815098	6593496	362	1261	166	3	71	27											BOURGOGNE-FRANCHE-COMTE
245	53245	SAINT-PIERRE-DES-LANDES	Commune simple		401277	6804722	401041	6808831	150	4223	941	2	53	52											PAYS DE LA LOIRE
388	16388	TROIS-PALIS	Commune simple		470682	6508784	470078	6508793	50	417	900	1	16	75											AQUITAINE-LIMOUSIN-POITOU-CHARE
342	38342	ROISSARD	Commune simple		908416	6423829	907792	6424155	812	1453	279	1	38	84											Auvergne-Rhone-Alpes
181	50181	FEUGERES	Commune simple		384947	6903205	385052	6902960	26	826	351	3	50	28											NORMANDIE
269	78269	GAZERAN	Commune simple		609631	6837774	611101	6836525	162	2574	1272	2	78	11											ILE-DE-FRANCE
126	50126	CHAVOY	Commune simple		381395	6856207	381511	6857446	93	371	131	1	50	28											NORMANDIE
541	25541	SEPTFONTAINES	Commune simple		941921	6658394	941298	6658222	743	1844	341	3	25	27											BOURGOGNE-FRANCHE-COMTE
232	55232	HARVILLE	Commune simple		899151	6892204	898866	6891965	211	562	112	3	55	44											ALSACE-CHAMPAGNE-ARDENNE-LOR
547	31547	SEYSSSES	Commune simple		563481	6268011	560982	6268247	175	2532	7954	1	31	76											LANGUEDOC-ROUSSILLON-MIDI-PYRE
332	33332	PORCHERES	Commune simple		464594	6441156	463878	6442196	43	1336	912	3	33	75											AQUITAINE-LIMOUSIN-POITOU-CHARE
107	14107	BRICQUEVILLE	Commune simple		411805	6917277	412351	6917357	21	695	154	1	14	28											NORMANDIE
364	55364	MOUZAY	Commune simple		860750	6931480	862757	6930828	195	3603	719	3	55	44											ALSACE-CHAMPAGNE-ARDENNE-LOR
311	88311	MONTMOTIER	Commune simple		936930	6768416	936789	6768641	286	428	50	1	88	44											ALSACE-CHAMPAGNE-ARDENNE-LOR
063	18063	CHAVANNES	Commune simple		652436	6639398	654038	6639467	171	2400	184	2	18	24											CENTRE-VAL DE LOIRE
075	47075	LA CROIX-BLANCHE	Commune simple		515943	6357769	515885	6357662	180	1301	889	1	47	75											AQUITAINE-LIMOUSIN-POITOU-CHARE
110	07110	JOYEUSE	Commune simple		798438	6376378	799196	6375892	212	1300	1689	1	07	84											Auvergne-Rhone-Alpes

=, <, >, >=, <=

Ne devraient pas soulever de questions ardues. Sauf peut-être pour les lettres, oui, ces symboles mathématiques peuvent fonctionner sur des chaînes alphanumériques (des lettres, quoi)

- " NOM\_COM " > 'B' ou " NOM\_COM " >= 'B' élimine toutes les communes qui commence par A.
- " NOM\_COM " < 'BE' va sortir toutes les communes qui commencent par A et BA (mais aussi BB, BD, si y'en avait.....).

En fait, imaginer votre liste triée par ordre alphabétique et faites un trait au niveau du B, si vous garder le haut de la liste, vous avez fait > 'B'... et inversement pour le bas < 'B'. Je vous laisse méditer là-dessus et faire des essais aux résultats parfois surprenants.

**!=**

signifie différent de, il fonctionne aussi sous sa forme <>

## LIKE et ILIKE

Rien à voir avec bookFace. C'est 2 mots permettent de faire une recherche sur des « modèles ». Avant les exemples, il est important de préciser que LIKE est sensible à la casse (MAJUSCULE-minuscules) alors ILIKE ne l'est pas.

Sauf recherches particulières, j'ai donc tendance à utiliser que ILIKE

- " NOM\_DEPT " LIKE 'FINISTERE' ⇒tester⇒ 283 lignes
- "NOM\_DEPT " LIKE 'finistere' ⇒tester⇒ 0 lignes
- " NOM\_DEPT " ILIKE 'FINISTERE' ⇒tester⇒ 283 lignes
- " NOM\_DEPT " ILIKE 'finisTERE' ⇒tester⇒ 283 lignes

Vous remarquerez qu'il faut le contenu exact et entier pour qu'ils fonctionnent, aussi LIKE et ILIKE sont quasi indissociables des caractères « jokers » % et \_.

## % et \_

Ces caractères dits JOKER vont permettent de remplacer une partie du contenu recherché.

% remplace une chaîne entière (plusieurs caractères)

- " NOM\_DEPT (I)LIKE 'FINIS%' ⇒tester⇒ va sortir tout ce qui commence par FINIS (FINISTERE, FINIS, FINISBLABLA, FINIS123.....)
- " NOM\_DEPT " (I)LIKE '%ERE' ⇒tester⇒ va sortir tout ce qui finit par ERE (FINISTERE, ISERE, LOZERE, BLABLAERE, 2323ERE.....)
- " NOM\_DEPT " (I)LIKE '%ER%' ⇒tester⇒ va donc sortir tout ce qui contient ER quelquepart (ALLIER, CHER, FINISTERE, GERS, HERAULT, ISERE, LOZERE, TERRITOIRE DE BELFORT...)

Quizz : devinez ce que sort 'A%E' ?

\_ (underscore), qui n'est pas dans les boutons, *c'est mon astuce du chef à moi*, ne remplace que 1 caractère, c'est donc comme la case blanche du scrabble®

- " NOM\_DEPT " (I)LIKE 'A\_B' va sortir ABB, ACB, A2B mais pas ABBB, A123B ou AaaalalalacB

## AND, OR

Deux autres boutons AND (ET) et OR (ou), ces expressions permettent de compiler plusieurs recherches. Un peu d'[Algèbre de Boole](#) (non, restez, cela va être simple) avant de montrer les exemples. Donc pour Boole et ses potes

## AND (ET)

condition 1	AND ou ET ou.	condition 2	resultat
vraie	AND	vraie	= vrai
vraie	AND	fausse	= faux
fausse	AND	vraie	= faux
fausse	AND	fausse	= faux

Le ET fait en sorte que les 2 conditions doivent être remplies pour que le résultat soit positif (vrai ou 1). C'est donc Fromage ET dessert, vous mangez les deux ou rien.

## OR (OU)

condition 1	OR ou OU ou +	condition 2	résultats
vraie	OR	vraie	vraie
vraie	OR	fausse	vraie
fausse	OR	vraie	vraie
fausse	OR	fausse	faux

le OU fait en sorte que l'une des deux conditions doit être vraie pour que le résultat soit positif (vrai ou 1). C'est donc Fromage OU dessert, vous mangez forcément l'un ou l'autre, voire les deux.

Comment cela intervient dans nos filtres, on peut s'en servir pour cumuler 2 requêtes

- " **CODE\_DEPT** " = '29' **AND** " **NOM\_COM** " **ILIKE** 'A%' ⇒ Ici, seules 3 communes remplissent les 2 critères : département 29 et commune commençant par A (Audierne, Argol, Arzano)
- " **CODE\_DEPT** " = '29' **OR** " **NOM\_COM** " **ILIKE** 'A%' ⇒ Je vais avoir toutes les communes du département 29 (la condition 1 est vraie pour ces communes) et toutes les communes commençant par " A " (la condition 2 est vraie pour ces communes) = 2207 résultats

Ça fait mal au crâne au début mais cela vient vite. Et puis rien n'empêche de se tromper et de taper OR à la place AND ou inversement, pour essayer.

## IN, NOT IN

In or not in, that is the question : whether 'tis nobler.....

Ces 2 petits mots permettent d'enchaîner plusieurs valeurs uniques dans un élément, c'est un élément de sous-requête. Cela ne vous parle pas, moi non plus alors exemple concret.

Au lieu de faire " **CODE\_DEPT** " = '22' **OR** " **CODE\_DEPT** " = '29' **OR** " **CODE\_DEPT** " = '56' qui me permet d'afficher les départements 22,29 et 56.

Je peux faire " **CODE\_DEPT** " **IN** ('22','29','56'), ce qui est plus simple lorsque l'on a plein de valeurs. Attention, prenez bien soin à respecter la syntaxe avec les (et, à taper vous-même.

Le NOT IN fait l'inverse de IN, donc " **CODE\_DEPT** " **NOT IN** ('22','29','56') va afficher tous les départements sauf les 22,29,56.

# NOT

Pour vous embrouiller par rapport à avant, le NOT permet de sortir le résultat inverse de celui demandé. Donc quelque soit ma requête, si je mets NOT devant, j'obtiens l'inverse de ma demande.

**NOT " NOM\_DEPT " (I)LIKE '%ER%'** va me sortir toutes les communes qui ne contiennent pas ER dedans,

**NOT " CODE\_DEPT " = '29'** va afficher tous les départements sauf le 29, j'aurais pu faire aussi **" CODE\_DEPT " != '29'** ou **" CODE\_DEPT " NOT IN ('29')**

Quizz : Que me sort **NOT " CODE\_DEPT " NOT IN ('22','29','56')** ?

En fait c'est pratique lorsque vous avez une requête super costaud et que vous voulez inverser son résultat sans tout retaper ou modifier.