

NegativeQueryProblems

- [Pure Negative Queries](#)
 - [Solr Magic](#)

Pure Negative Queries

One common situation encountered in search with Lucene-based programs is the problem of pure negative query clauses. This is a query clause that consists ONLY of negative phrases, using the - notation or the "NOT" boolean operator.

Lucene (the search API that Solr is built on) is not capable of handling purely negative query clauses. The reason for this is fairly simple to understand, but might be surprising. A negative query is a subtraction. If you start with nothing, then try to subtract something, the end result is nothing – because that's what you started with.

In order to get results from a negative query, you must add another clause to the query so that the query starts with something to subtract from. Normally you'll want to start with everything in those situations, but that's not always necessary.

Solr Magic

You might ask why a query like this works:

```
/solr/corename/select?q=-field:value
```

This is because when the entire query is a single negative clause without any complexity such as parentheses, Solr is able to detect the impossible situation and fix it for you. A query of **-field:value** is translated internally into ***:* -field:value** by Solr before it gets sent down to the Lucene layer – a starting point of all documents is implicitly added to the query, and then the subtraction works.

These queries will not work as expected:

```
-field1:value1 OR field2:value2
```

```
field2:value2 OR -field1:value1
```

When there are multiple clauses in the input query, Solr's negative query detection does not work, so the purely negative clauses in these queries cannot be automatically fixed. If these queries are rewritten as the following, adding a starting point of "all documents" to the negative clause, then it works as expected:

```
field2:value2 OR (*:* -field1=value1)
```

Negative query clause problems can be very subtle. This query looks similar to the ones above that don't work, but it DOES work just fine:

```
field1:value1 AND -field2:value2
```

The fact that this works but similar queries using "OR" don't work is NOT a bug. It's just a byproduct of the way that boolean and MUST/SHOULD/MUST NOT query logic works. With this structure, Solr is able to detect and fix the problem with the negative query in the second clause, but with the OR, it can't.